



II SIMECA - IFPR

SIMPÓSIO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO

22 a 25 de novembro de 2021



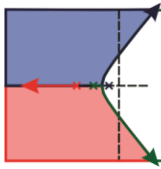
**INSTITUTO
FEDERAL**
Paraná

Campus
Jacarezinho

CADERNO DE RESUMOS

**II - SIMPÓSIO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DO
INSTITUTO FEDERAL DO PARANÁ (II SIMECA - IFPR)**

22 A 25 DE NOVEMBRO DE 2021



II SIMECA - IFPR
SIMPÓSIO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO



INSTITUTO FEDERAL DO PARANÁ – CAMPUS JACAREZINHO

Diretor Geral: Rodolfo Fiorucci

Diretora de Ensino, Pesquisa e Extensão: Andreza Tangerino Mineto

Coordenador de Ensino: David José de Andrade Silva

Coordenador do Curso de Engenharia de Controle e Automação: José André Mota de Queiroz

Comissão Organizadora do II Simpósio de Engenharia de Controle e Automação

Membros Docentes

João Paulo Lima Silva de Almeida (Presidente da Comissão)

Anderson Kenji Hirata

André Luiz Salvat Moscato

Avyner Lorrann de Oliveira Victor

Felipe Augusto de Aguiar Possoli

Gustavo Henrique Bazan

José André Mota de Queiroz

Luis Fabiano Barone Martins

Luiz Eduardo Pivovar

Ricardo Breganon

Rodolfo Rodrigues Barrionuevo Silva

Uiliam Nelson Lendzion Tomaz Alves

Membros Técnicos de Laboratórios

Fernando Sabino Fonteque Ribeiro

Gustavo Vendrame Barbara

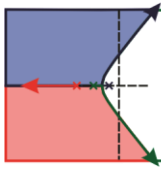
APRESENTAÇÃO

A coordenação do curso de Engenharia de Controle e Automação do Instituto Federal do Paraná (IFPR) - Campus Jacarezinho, juntamente com a Comissão de Organização, apresenta o II Simpósio de Engenharia de Controle e Automação. O evento tem o intuito de reunir estudantes, pesquisadores, profissionais da área e público em geral para participarem do ciclo de palestras, minicurso e apresentações de trabalhos acadêmicos da área.

As palestras proferidas por profissionais externos da Instituição abordam diversos temas relacionados à área de Engenharia de Controle e Automação, suas tecnologias e inovações. O minicurso tem como público-alvo os próprios estudantes do curso e objetiva apresentar os recursos tecnológicos computacionais básicos, necessários para a compreensão de seus diversos componentes curriculares. As apresentações dos trabalhos acadêmicos são primordiais para apresentar temas para debates e contribuem para a formação de profissionais críticos e responsáveis em sua atuação. Portanto, o evento se consolida por meio das diversas atividades mencionadas, enaltecendo a multidisciplinaridade desta formação.

Por fim, este documento apresenta uma síntese de todo o conteúdo apresentado no II SIMECA. Os resumos expandidos dos trabalhos submetidos, aprovados pela Comissão Organizadora e apresentados em dia próprio no evento, estão disponíveis para futuras consultas.

Prof. Dr. João Paulo Lima Silva de Almeida
Presidente da Comissão Organizadora



APRESENTAÇÃO DO INSTITUTO FEDERAL DO PARANÁ - JACAREZINHO

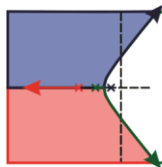


O Campus Jacarezinho iniciou suas atividades em maio de 2010, com os cursos subsequentes de alimentos, informática e eletromecânica. No ano seguinte, iniciou os Cursos Integrados e contava apenas com a infraestrutura do Bloco Didático 02. Atualmente, o Campus conta com um Bloco Administrativo, dois Blocos Didáticos e outro de Laboratórios.

Os seguintes cursos são ofertados:

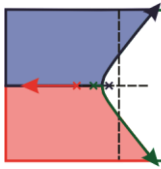
- **Técnicos Integrados ao Ensino Médio:** Alimentos, Eletromecânica, Eletrotécnica, Informática e Mecânica.
- **Técnico Subsequente:** Teatro.
- **Graduações:** Engenharia de Controle e Automação, Licenciatura em Química e Tecnologia em Sistemas para Internet.

Através do Campus Jacarezinho, o IFPR reafirma a sua função social de oferta de vagas públicas, gratuitas e de qualidade.



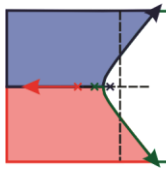
CRONOGRAMA

Palestras		
Data - horário	Título	Palestrante
22/11/2021 19h30min	Desenvolvimento de carreira e panorama de mercado nas áreas de tecnologia e engenharia	Eng. Valdeir Ribeiro Soares Linkedin: https://www.linkedin.com/in/valdeir-ribeiro-soares-27115982/
23/11/2021 19h30min	Uma síntese do Programa Espacial Brasileiro e o Satélite Amazônia 1	Eng. Leonardo Rennó Campos de Paiva Currículo Lattes: http://lattes.cnpq.br/1417587506679440
24/11/2021 19h30min	Sistemas Embarcados: Aplicações e Desenvolvimento de Soluções de Engenharia	Eng. Erick Araujo Nunes Currículo Lattes: http://lattes.cnpq.br/4241870936474335
Minicurso		
Data - horário	Título	Mediadores
23/11/2021 14:00 - 16:00	Métodos computacionais para equações diferenciais aplicadas	Profa. Me. Carolina Lupifierio de Queiroz Currículo Lattes: http://lattes.cnpq.br/9359347437115188
25/11/2021 14:00 - 16:00		
Apresentação de Trabalhos		
Data - horário	Título	Apresentador
25/11/2021 19h30	Proposta de utilização da placa Arduino UNO R3 em um sistema de automatização em estacionamento automotivo	Ana Clara Corradi Jorge
25/11/2021 19h50	Estudo do lugar das raízes em um modelo de robô paralelo	Ricardo Breganon
25/11/2021 20h10	Controle de um sistema viga-hélice usando um compensador de avanço e atraso de fase	Hugo Fernando Yamanaka
25/11/2021 20h30	Aspectos de implementação de controlador PI em ambiente simulado	Filipe Honório Varaschim
25/11/2021 20h50	Projeto de compensador para conversos boost utilizando o fator K	Luís Fabiano Barone Martins
25/11/2021 21h10	Protótipo para automação residencial utilizando Arduino	Ana Carolina Crespo Alves Tavares
25/11/2021 21h30	Modelagem e análise dinâmica de um sistema viga-hélice	Carlos Alexandre de Souza Bispo



SUMÁRIO

Proposta de Utilização da Placa Arduino Uno R3 em um Sistema de Automatização em Estacionamento Automotivo	7
Estudo do Lugar das Raízes em um Modelo de Robô Paralelo	11
Controle de um Sistema Viga-Hélice Usando um Compensador de Avanço e Atraso de Fase	15
Aspectos de Implementação de Controlador PI em Ambiente Simulado	19
Projeto de Compensador para Conversos Boost Utilizando o Fator K	23
Protótipo para Automação Residencial Utilizando Arduino	27
Modelagem e Análise Dinâmica de um Sistema Viga-Hélice	31



PROPOSTA DE UTILIZAÇÃO DA PLACA ARDUINO UNO R3 EM UM SISTEMA DE AUTOMATIZAÇÃO EM ESTACIONAMENTO AUTOMOTIVO

JORGE, Ana Clara Corradi¹; BREGANON, Ricardo ²; ALMEIDA, João P. L. S³; ALVES, Uiliam N. L. T ⁴.

¹ Técnico em Mecânica Integrado ao Ensino Médio, IFPR, Jacarezinho, Paraná, e-mail: accjbarba@gmail.com.

² Doutor em Engenharia Mecânica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, e-mail: ricardo.breganon@ifpr.edu.br.

³ Doutor em Engenharia Elétrica e Informática Industrial, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, e-mail: joao.almeida@ifpr.edu.br.

⁴ Doutor em Engenharia Elétrica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, e-mail: uiliam.alves@ifpr.edu.br.

RESUMO

O aumento da frota automobilística no Brasil trouxe diversos problemas para as cidades, um deles sendo a dificuldade em conseguir achar uma vaga de estacionamento. Nesse contexto, surgiu a ideia de propor uma possível solução para este problema específico. Essa proposta consiste em automatizar um estacionamento, utilizando a placa Arduino UNO R3 para fazer o gerenciamento de vagas. A ideia de utilizar o Arduino como solução se deu pela sua versatilidade e baixo custo de montagem. Basicamente, um sensor LDR detecta se existe ou não um automóvel em determinado espaço, envia a informação para o Arduino que, por sua vez, manda a informação para um Display LCD. Assim, o motorista tem a informação se o estacionamento possui vagas antes de entrar no mesmo, poupando tempo. Os testes desse sistema foram feitos em um simulador da placa Arduino, o Tinkercad, que possibilitou o desenvolvimento do projeto em ambiente virtual.

PALAVRAS-CHAVE: Arduino Uno, Automatização, Gerenciamento de Vagas.

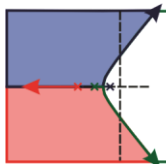
INTRODUÇÃO

A frota automobilística brasileira conta com mais de 110 milhões de automóveis registrados (MINISTÉRIO DA INFRAESTRUTURA, 2021). Essa grande quantidade de veículos trouxe diversos problemas para as cidades sendo um deles a dificuldade em encontrar uma vaga para estacionar o veículo. Em grandes cidades, como é o caso de São Paulo, existem leis municipais que obrigam novos estabelecimentos a destinarem um número mínimo de vagas de estacionamento. Isso gerou outro problema, muitos estabelecimentos contam com vagas disponíveis, mas sem uma boa visualização das mesmas (LAMAS, 2014).

Observou-se então a necessidade de criar uma solução para esse problema que se baseia no gerenciamento e visualização das vagas disponíveis em determinado espaço utilizando a plataforma Arduino.

O Arduino consiste em uma tecnologia aberta para desenvolvimentos de experimentos que envolvam microcontroladores. De uma forma geral, os elementos de *hardware* consistem em uma placa que se usa para construir projetos e o *software* onde se escrevem os comandos a serem executados pelo processador presente na placa utilizando o Arduino IDE. Além disso, possui um baixo custo e é fácil de programar (SOUZA, 2013). Por possibilitar a execução de experimentos com prototipagem rápida e possuir um baixo custo, o Arduino UNO R3 foi escolhido como a placa controladora do sistema proposto.

Para ser possível realizar as simulações com o Arduino foi utilizado o software Tinkercad, que é um simulador com diversos sensores e componentes, além da placa Arduino UNO R3 (TINKERCAD, 2017). Tendo isso em vista, desenvolveu-se essa proposta pensando em uma solução de baixo custo e que possibilite uma visualização das vagas disponíveis em determinado estacionamento.



O sistema de gerenciamento de vagas automotivas foi simulado e os resultados foram satisfatórios. Também foi realizada a modelagem 3D de uma possível maquete para o estacionamento. A confecção da maquete é um trabalho futuro que possibilitará a simulação do sistema em ambiente físico.

SISTEMA AUTOMATIZADO PARA GERENCIAMENTO DE VAGAS

Os componentes utilizados na proposta do sistema de gerenciamento de vagas foram quatro sensores LDRs, que fazem a detecção da intensidade de luz de cada vaga, um Display LCD, que tem a função de mostrar a quantidade de vagas disponíveis, e uma placa Arduino UNO R3, que interliga os componentes e faz o gerenciamento das informações recebidas pelos sensores. Esses componentes são conectados às portas do Arduino, no caso os LDRs estão dispostos nas portas analógicas A0, A1, A2 e A3, já o Display LCD é comandado pelas portas digitais 2, 3, 4, 5, 11 e 12.

O software do sistema funciona da seguinte forma: inicialmente, se faz o mapeamento de todos os pinos utilizados no Arduino definindo quais portas servirão de entrada ou saída de informação, no caso os sensores são colocados como entrada e o Display LCD como saída. Depois que é feita essa definição, é feito o “loop” que é a parte da programação na qual as funções são executadas repetidamente, ou seja, toda informação necessária para a análise dos dados é tratada nessa função.

A comunicação entre os componentes do sistema é feita da seguinte forma: as portas analógicas (A0, A1, A2 e A3) informam a intensidade da luz que cada LDR está captando. Esses dados são enviados para o Arduino. Na placa, os dados são processados. No caso, o valor utilizado para a intensidade de luz foi 300 Lux, então, se a luminosidade estiver acima desse valor significa que a vaga está vazia, já se estiver abaixo desse valor a vaga está ocupada, essa informação é dada em números binários, ou seja, 0 para ocupada e 1 para disponível. Ainda no Arduino é feito a soma do total de vagas disponíveis. Essa informação é enviada para o Display LCD que tem a função de mostrar visualmente a informação que o Arduino processou. Esse processo está esquematizado na Figura 1.

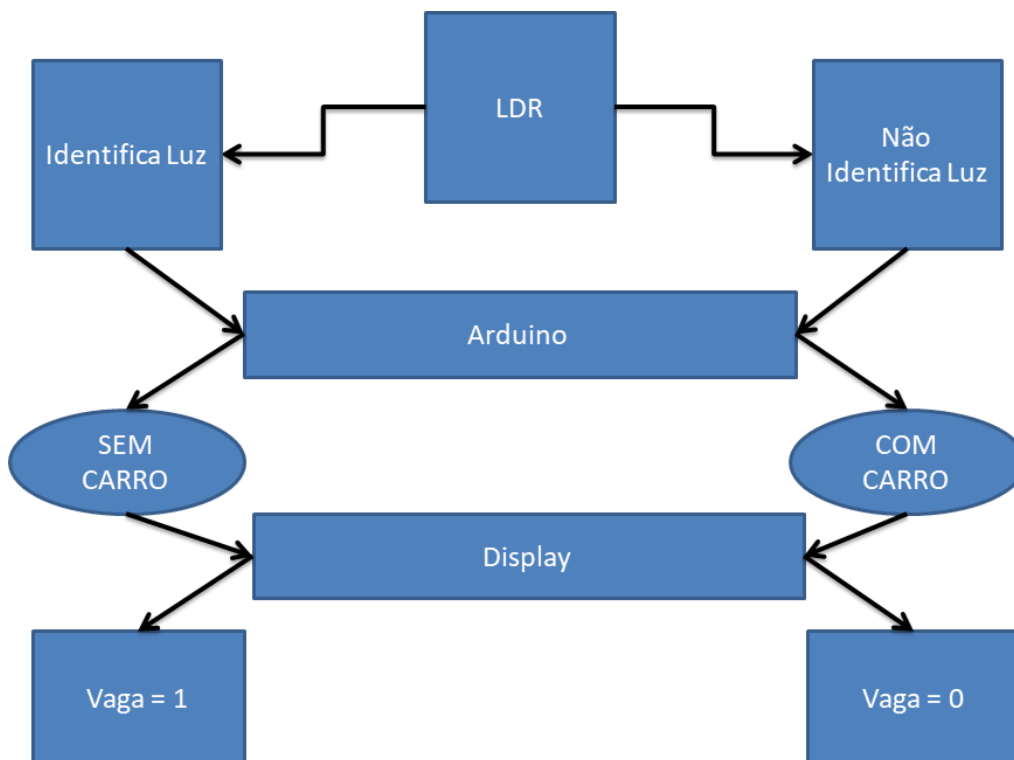
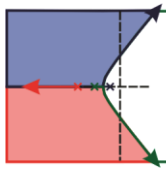


Figura 1 - Lógica do funcionamento do sistema de automação de estacionamento.



Em uma situação prática, o motorista irá passar em frente ao estacionamento e verá se possui ou não vagas disponíveis. Se possuir, ele estacionará e imediatamente o valor mostrado no Display LCD será alterado. Na Figura 2 é mostrada a simulação no Tinkercad de uma possível situação para o gerenciador de vagas, nela existem duas vagas disponíveis, ou seja, nessas duas vagas a incidência de luz é maior que o limite definido na programação. As vagas disponíveis estão representadas na cor verde e as ocupadas na cor vermelha.

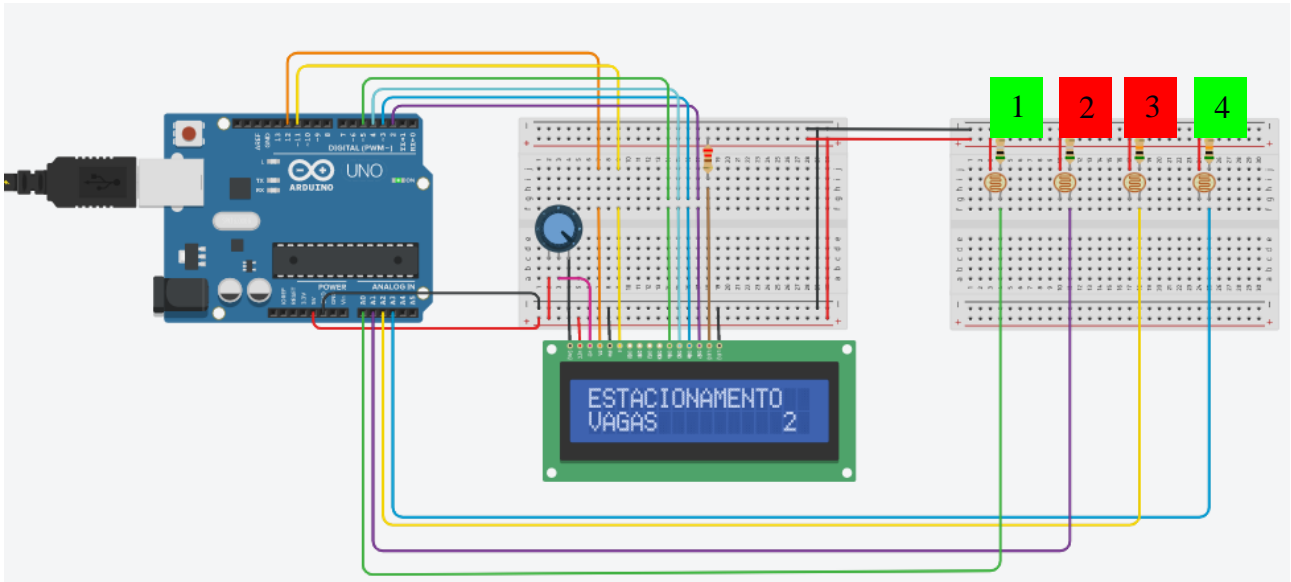


Figura 2 - Simulação no software Tinkercad do sistema de estacionamento proposto.

O próximo passo do desenvolvimento do sistema de gerenciamento de vagas seria fabricar uma maquete para realizar os testes, então se fez o projeto 3D da maquete. Na Figura 3 é mostrada a modelagem 3D de uma possível maquete com a disposição de cada componente no estacionamento. Ao centro de cada vaga são colocados os sensores LDRs e na parte exterior do estacionamento fica disponibilizado o Display LCD.

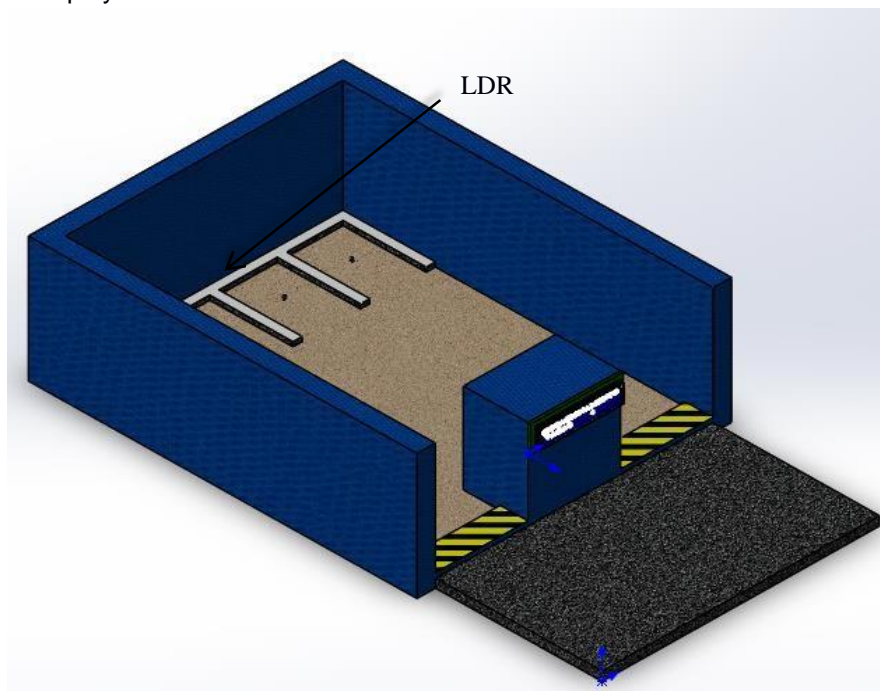


Figura 3 - Modelagem 3D de uma maquete para o teste do sistema de gerenciamento de vagas de estacionamento.



CONSIDERAÇÕES FINAIS

O sistema de gerenciamento de vagas automotivas foi devidamente simulado e os resultados obtidos foram satisfatórios. A possível implantação desse sistema traria grande economia de tempo para os motoristas em cidades com problemas de estacionamentos e o uso do Arduino permite um projeto de baixo custo. Além disso, a implementação deste sistema automatizado poderia ser incorporado ao gerenciamento de redes de estacionamentos de grande porte, tais como os que possuem diversas vias de entrada/saída (shoppings). Neste caso, também seria possível o sistema indicar qual a via de acesso a ser tomada pelo motorista, sem a interferência de um agente auxiliar.

Durante testes iniciais com o sensor LDR foi possível perceber que existe uma grande variação de luminosidade em um estacionamento fazendo com que a detecção da presença de automóveis via sensores LDRs não seja confiável em cenários com variação de luminosidade, por isso para trabalhos futuros sugere-se a utilização do sensor Reed Switch e a fabricação de um protótipo. No caso da utilização do sensor magnético Reed Switch deve-se incluir um ímã na parte inferior de cada protótipo de automóvel. Isso deve ser feito, pois os contatos presentes no sensor magnético fecham quando entram em um campo magnético, possibilitando assim a detecção do veículo.

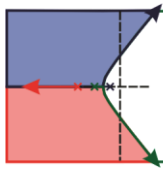
REFERÊNCIAS

LAMAS, J. **Estacionamentos, os novos vilões da mobilidade urbana**. Exame, 2014 Disponível em: <<https://exame.com/brasil/estacionamentos-os-novos-viloes-da-mobilidade-urbana/>>. Acesso em: 13 out. 2021.

MINISTÉRIO DA INFRAESTRUTURA. **Frota de Veículos – 2021**. Portal Gov.br, 2021. Disponível em: <<https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-denatran/frota-de-veiculos-2021>>. Acesso em: 11 out. 2021.

SOUZA, F. Arduino UNO - Conheça o hardware da placa Arduino em detalhes. Embarcados, 2013. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em: 13 jul. 2021.

TINKERCAD. **Apostila de introdução ao Tinkercad**. DocPlayer, 2017. Disponível em: <<https://docplayer.com.br/65200375-Apostila-de-introducao-ao-tinkercad-antigo-circuits-io.html>>. Acesso em: 18 ago. 2021.



ESTUDO DO LUGAR DAS RAÍZES EM UM MODELO DE ROBÔ PARALELO

BREGANON, Ricardo¹; ALVES, Uiliam Nelson Lenzion Tomaz²; MARTINS, Luis F. Barone³; ALMEIDA, João Paulo L. Silva de⁴

¹ Doutor em Engenharia Mecânica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, ricardo.breganon@ifpr.edu.br

² Doutor em Engenharia Elétrica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, uiliam.alves@ifpr.edu.br

³ Doutor em Engenharia Elétrica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, luis.martins@ifpr.edu.br

⁴ Doutor em Engenharia Elétrica e Informática Industrial, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, joao.almeida@ifpr.edu.br

RESUMO

Um robô paralelo é um dispositivo de seis graus de liberdade, em que uma plataforma móvel é ligada a uma base fixa por meio de seis atuadores. A alteração da posição e orientação da plataforma móvel é realizada por meio da extensão de cada atuador e o comprimento de cada atuador para uma dada configuração é determinada pela cinemática inversa do equipamento. O modelo matemático utilizado neste trabalho para o projeto do controlador foi obtido por meio da função de transferência de um atuador eletromecânico, relacionando a tensão aplicada e o comprimento do atuador, que foi replicada para todos os outros que compõem o robô analisado. O projeto do controlador foi realizado por meio do estudo do lugar das raízes e então um ganho K foi encontrado para a realimentação. Resultados de simulação são apresentados para verificar o desempenho do sistema e validar a metodologia proposta.

PALAVRAS-CHAVE: Robô paralelo. Lugar das raízes. Atuador eletromecânico.

INTRODUÇÃO

Os robôs paralelos são estruturas que podem ser usadas para controlar ou gerar movimentos em 6 graus de liberdade, com aplicações em diferentes processos de manufatura, bases de simuladores de voo e de automóveis e tarefas de precisão (DASGUPTA & MRUTHYUNJAYA, 2000; BEN-HORIN, SHOHAM & DJERASSI, 1998).

O mecanismo deste tipo de robô consiste em uma base inferior fixa e uma base superior móvel que são conectadas por 6 elos. Estes elos estão formados por duas juntas universais posicionadas nos extremos de cada atuador (AT). Os atuadores podem se movimentar de forma independente mudando seu comprimento, porém, para a base móvel alcançar uma posição desejada, precisa-se da combinação dos comprimentos de todos os atuadores, obtendo-se desta forma a transformação dos seis deslocamentos lineares em três movimentos de translação (X , Y , Z) e três movimentos de rotação (φ , θ , ψ) da base superior móvel. A Figura 1 apresenta o diagrama esquemático de um robô paralelo, também conhecido como plataforma de Stewart (STEWART, 1965).

Para o controle de posição e orientação do robô paralelo, a cinemática inversa é um aspecto a ser considerado, onde, a partir da posição e atitude da base superior móvel com relação à base inferior fixa obtém-se os comprimentos dos atuadores. A saída da cinemática inversa é o comprimento que cada atuador deve possuir para que a base móvel alcance a posição desejada, gerando assim a referência a ser aplicada em cada um dos atuadores (BREGANON, 2018; ACUÑA, 2009).

Este trabalho apresenta o projeto de um controlador pelo método do lugar das raízes aplicados a um modelo de robô paralelo. O modelo matemático do robô paralelo juntamente com a cinemática inversa utilizados para o projeto do controlador podem ser consultados em Breganon (2014).

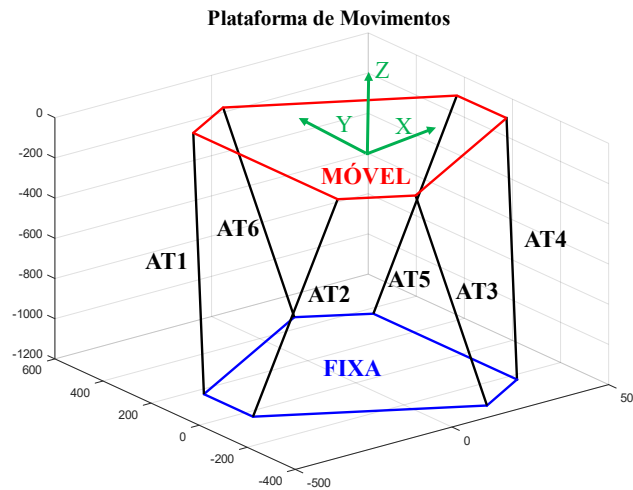
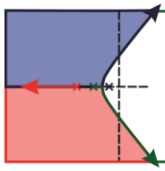


Figura 1 - Esquemático de um robô paralelo.

PROJETO DO CONTROLADOR USANDO O LUGAR DAS RAÍZES

Para o projeto do controlador, foi utilizado o modelo matemático de um robô paralelo. Ele consiste na dinâmica de um atuador eletromecânico, composto por um motor elétrico 12V DC com transmissão por engrenagens para o fuso de esfera com curso máximo de 610 mm, que é replicada para todos os atuadores. Esta dinâmica é mostrada na Equação (1), sendo a entrada a tensão aplicada ($V(s)$) e a saída, o comprimento do atuador ($L(s)$).

$$\frac{L(s)}{V(s)} = \frac{7137}{s^3 + 68,45s^2 + 1171s} \tag{1}$$

O método do lugar das raízes permite analisar graficamente como os polos de um sistema de malha fechada variam conforme a mudança de um parâmetro do sistema. A abordagem mais comum analisa como os polos de malha fechada variam conforme o valor de um ganho de realimentação K assume valores de 0 a ∞ (NISE, 2011). A Figura 2 mostra o uso do controlador com ganho K no controle dos atuadores do robô paralelo.

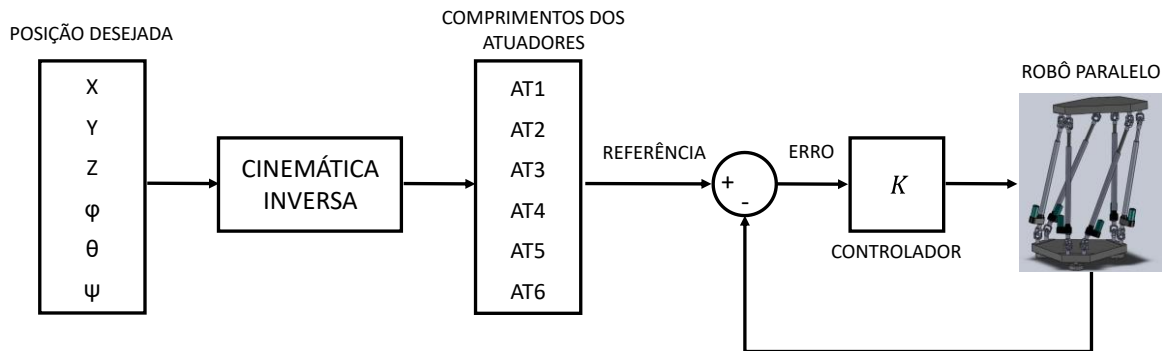
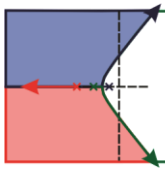


Figura 2 - Diagrama de blocos do robô paralelo.

Para a dinâmica do atuador (Equação (1)), é possível obter o gráfico do lugar das raízes apresentado na Figura 3. Através da ferramenta *rtool* do Matlab®, foi escolhido um tempo de acomodação de $t = 1s$ e obteve-se um ganho $K = 0,32927$, utilizado para o controle de cada atuador que posiciona e



orienta o robô paralelo. Assim, o sistema de malha fechada tem a função de transferência dada na Equação (2), sendo $L(s)$ o comprimento do atuador e $R(s)$ a referência. Os polos em malha fechada destacados com asterisco na Figura 3 são $[-45,5629 \quad -13,1045 \quad -9,7826]$.

$$\frac{L(s)}{R(s)} = \frac{5841}{s^3 + 68,45s^2 + 1171s + 5841} \quad (2)$$

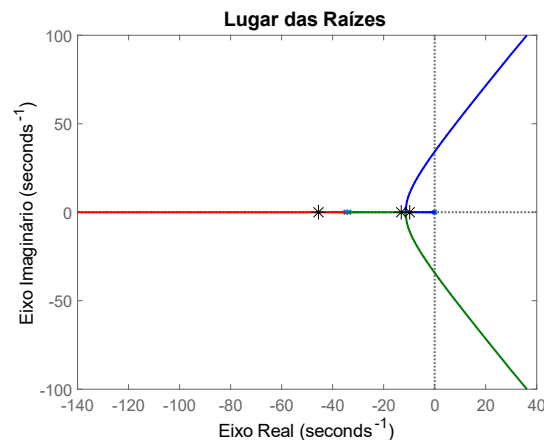


Figura 3 - Lugar das Raízes do sistema.

RESULTADOS E DISCUSSÕES

A fim de validar a proposta deste trabalho, aplicou-se ao modelo matemático do robô paralelo o controlador projetado pelo método do lugar das raízes. Resultados de simulação foram analisados para uma mudança de configuração da plataforma superior de 15° em φ .

A Figura 4 apresenta a posição final dos 6 atuadores para o ângulo requerido, a Figura 5 mostra que o erro tendeu a zero para todos os atuadores e por fim, a Figura 6 apresenta a tensão aplicada para cada atuador chegar no comprimento requerido, necessário para posicionar o robô paralelo na configuração desejada. É importante ressaltar que, para uma tensão nula (ou próxima de zero), o atuador fica imóvel.

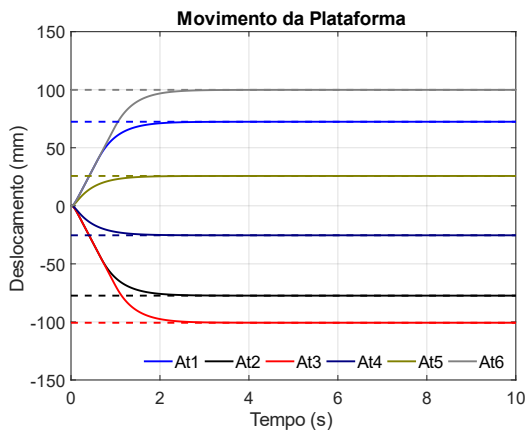


Figura 4 - Comprimento dos atuadores para configuração de 15° em φ .

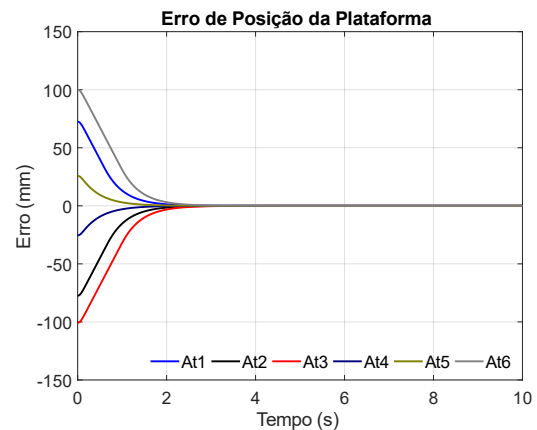


Figura 5 - Erro entre a referência e o comprimento dos atuadores para configuração de 15° em φ .

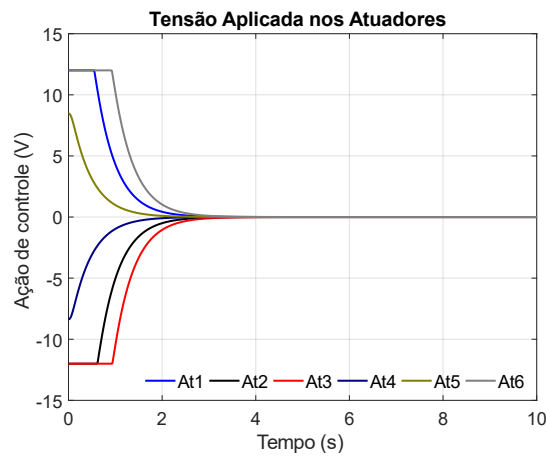
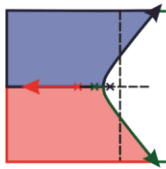


Figura 6 - Ação de controle para configuração de 15° em φ .

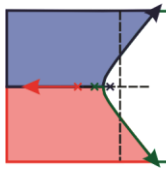
CONSIDERAÇÕES FINAIS

Neste trabalho realizou-se o projeto do controlador pelo método do lugar das raízes, utilizando a ferramenta *rltool* do Matlab® e aplicou-se no modelo matemático de um robô paralelo.

Conclui-se que a metodologia utilizada foi adequada para o robô paralelo, pois, através de resultado de simulação, observou-se que o sistema foi capaz de chegar na posição desejada, fazendo o erro tender a zero.

REFERÊNCIAS

- ACUÑA, H. G. **Projeto mecatrônico de uma plataforma de Stewart para simulação dos movimentos nos navios**. Dissertação (Mestrado em Engenharia Mecânica), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 112 p., 2009.
- BEN-HORIN, R., SHOHAM, M., & DJERASSI S. **Kinematics, dynamics, and construction of a planary actuated parallel robot**. Robotics and Computer Integrated Manufacturing. V.14, 163-172 p., 1998.
- BREGANON, R. **Análise, desenvolvimento e controle de uma plataforma de movimentos com 6 graus de liberdade**. Tese (Doutorado em Engenharia Mecânica), Escola de Engenharia de São Carlos da Universidade de São Paulo, 143 p., 2014.
- BREGANON, R., MONTEZUMA, M. A. F., SOUZA, M. M., LEMES, R. C., & BELO, E. M. **Optimal H Infinity Controller Applied to a Stewart Platform**. International Journal of Advanced Engineering Research and Science. V.5 (7), 51-59 p., 2018.
- DASGUPTA, B., & MRUTHYUNJAYA, T. S. **The Stewart Platform Manipulator: a review**. Mechanism and Machine Theory. 35. 15-40 p., Pergamon, 2000.
- NISE, N. S. **Control Systems Engineering**. John Wiley & Sons, 2011.
- STEWART, D. **A Platform with Six Degrees of Freedom**. Proceedings of Institution of Mechanical Engineers. V.180, n.15, 371-386 p., 1965.



CONTROLE DE UM SISTEMA VIGA-HÉLICE USANDO UM COMPENSADOR DE AVANÇO E ATRASO DE FASE

YAMANAKA, Hugo F.¹; BISPO, Carlos A. S.²; ALMEIDA, João, P. L. S.³; BREGANON, Ricardo⁴; ALVES, Uiliam, N. L. T.⁵

¹ Discente do curso de Engenharia de Controle e Automação, IFPR, Jacarezinho, Paraná, e-mail: hugofermandoyamanaka@gmail.com.

² Discente do curso de Engenharia de Controle e Automação, IFPR, Jacarezinho, Paraná, e-mail: carlosalexandrehk@gmail.com.

³ Doutor em Engenharia Elétrica e Informática Industrial, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, e-mail: joao.almeida@ifpr.edu.br.

⁴ Doutor em Engenharia Mecânica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, e-mail: ricardo.breganon@ifpr.edu.br.

⁵ Doutor em Engenharia Elétrica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, e-mail: uiliam.alves@ifpr.edu.br.

RESUMO

No estudo de sistemas de controle, um equipamento que pode ser utilizado é o sistema viga-hélice, o qual consiste em uma haste, em que uma das extremidades rotaciona em torno de um eixo fixo enquanto há um conjunto de motor e hélice na extremidade livre. Este trabalho apresenta um modelo matemático em espaço de estados de um sistema viga-hélice e uma estratégia de controle utilizando um compensador de avanço e atraso de fase. O compensador foi desenvolvido a partir da equação diferencial, linearizada através da série de Taylor. Primeiro, foi projetado um compensador de avanço de fase para melhorar a resposta transitória do sistema. Posteriormente, projetou-se o compensador de atraso de fase para atender ao requisito de erro em regime permanente. Utilizou-se o *software* Simulink/Matlab® para representar a planta não linear com o compensador, a fim de simular o sistema e verificar a eficácia do controlador.

PALAVRAS-CHAVE: Viga-hélice, compensador de avanço e atraso de fase, resposta transitória, erro em regime permanente.

INTRODUÇÃO

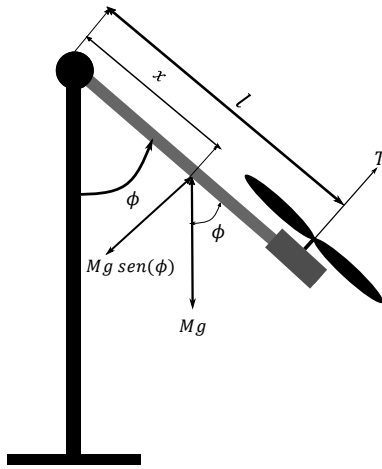
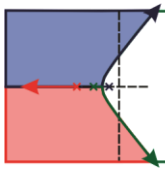
Os projetos de controladores de sistemas mecânicos envolvendo a dinâmica do viga-hélice são considerados problemas clássicos de controle (JOB; JOSE, 2015). Os sistemas viga-hélice, em geral, são naturalmente não lineares e são aproximados, em alguns casos, por modelos lineares em torno do respectivo ponto de equilíbrio (ENIKOV; CAMPA, 2012).

Uma abordagem comum é gerar uma aproximação linear da planta e, após isso, usar o modelo linearizado para projetar um controlador. O *software* Simulink/Matlab® pode então ser empregado para simular o desempenho do controlador quando aplicado ao modelo não linear completo (CHEN, 1998).

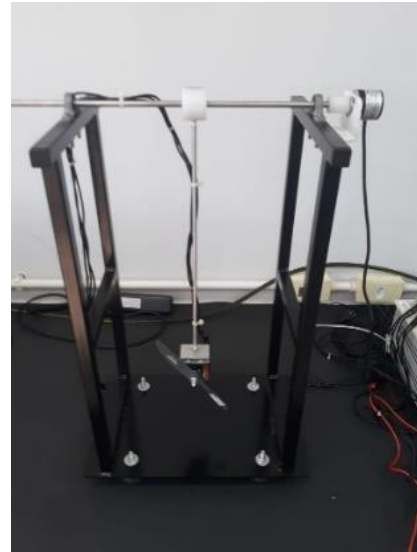
Com base nas características do sistema em análise, foi desenvolvido um controle por compensação de avanço e atraso de fase, através da função de transferência do sistema, no *software* Matlab®, para reduzir o tempo de acomodação e melhorar o erro em regime permanente. Tal controle foi implementado no *software* Simulink/Matlab® junto com a representação do sistema não linear em espaço de estados, a fim de simular o sistema e verificar a eficácia do controlador.

MODELO MATEMÁTICO DO SISTEMA VIGA-HÉLICE

Para explicar o princípio de funcionamento do viga-hélice é possível utilizar como analogia um sistema de pêndulo simples, o qual possui uma massa acoplada a um fio inextensível e fixo em uma extremidade a uma superfície (JOB; JOSE, 2015). O sistema desenvolvido consiste em uma haste, com rotação em torno de um eixo fixo. Na outra extremidade da haste existe um atuador, que é o conjunto motor e hélice que causa a variação do ângulo ϕ , produzindo uma força que atua de forma contrária à gravidade, a depender da posição do motor. Na Figura 1 (a) é apresentado um esquemático de funcionamento do sistema viga-hélice e na Figura 1 (b), o protótipo desenvolvido no laboratório de automação do IFPR/Jacarezinho. O objetivo do sistema é controlar o ângulo de inclinação da haste em relação à vertical.



(a)



(b)

Figura 1 - (a) Diagrama de corpo livre do sistema viga-hélice e (b) Protótipo viga-hélice presente no laboratório de automação do IFPR/Jacareizinho.

A partir do diagrama de corpo livre do sistema, Figura 1 (a), foi feito o somatório e torques envolvidos, em relação ao ponto de rotação, resultando em

$$J \frac{d^2 \phi(t)}{dt^2} + c \frac{d\phi(t)}{dt} + mlgd_c \text{sen}(\phi(t)) = T(t). \quad (1)$$

Ao escolher x_1 e x_2 como variáveis de estado do sistema, considerando o deslocamento angular $x_1(t) = \phi(t)$ e a velocidade angular $x_2(t) = d\phi(t)/dt$, deriva-se em ambos os lados, $dx_1(t)/dt = d\phi(t)/dt$ e $dx_2(t)/dt = d^2\phi(t)/dt^2$, substituindo em (1), e rearranjando os termos, obtêm-se

$$\dot{x}_1(t) = x_2(t), \quad (2)$$

$$\dot{x}_2(t) = -\frac{mld_c \text{sen}(x_1(t))}{J} - \frac{cx_2(t)}{J} + \frac{T(t)}{J}. \quad (3)$$

Linearizando (1) em torno de um ponto de equilíbrio usando a série de Taylor (NISE, 2018), feitas as seguintes substituições: $\Delta_\phi = \phi - \phi_0$, ($d\Delta_\phi/dt = d\phi/dt$) e $\Delta_T = T(t) - mlgd_c \text{sen}(\phi_0)$, tem-se

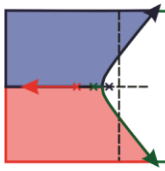
$$J \frac{d^2 \Delta_\phi}{dt^2} + c \frac{d\Delta_\phi}{dt} + mlgd_c \cos(\phi_0) \Delta_\phi = \Delta_T. \quad (4)$$

Aplicada a transformada de Laplace (NISE, 2018) em (4), são organizados os termos de entrada do sistema sobre a saída, obtendo-se a função de transferência

$$\frac{\Delta_\phi(s)}{\Delta_T(s)} = \frac{\frac{1}{J}}{s^2 + \frac{cs}{J} + \frac{mld_c \cos(\phi_0)}{J}}. \quad (5)$$

Para projetar o controlador de avanço e atraso de fase no software Matlab® é necessário utilizar a função de transferência do sistema em malha aberta (NISE, 2018). Como parâmetros do sistema viga-hélice, foram adotados os seguintes valores: aceleração da gravidade $g = 9,81 \text{ m/s}^2$, massa do conjunto hélice motor $m_1 = 0,295 \text{ kg}$, massa da haste $m_2 = 0,020 \text{ kg}$, comprimento da haste $l = 0,306 \text{ m}$, coeficiente de amortecimento viscoso $c = 0,0086 \text{ Nms/rad}$, momento de inércia do sistema $J = 0,0282 \text{ kg.m}^2$, posição do centro de massa do sistema $d_c = 0,2963 \text{ m}$ e ângulo $\phi_0 = 90^\circ$. Ao substituir os parâmetros em (5), obteve-se:

$$G(s) = \frac{35,4}{s(s + 0,3045)}. \quad (6)$$



COMPENSADOR DE AVANÇO E ATRASO DE FASE

Primeiro, projetou-se um compensador de avanço de fase para melhorar a resposta transitória do sistema. Em seguida, avaliou-se a melhoria no erro em regime permanente que ainda era requerida. Finalmente, foi projetado o compensador de atraso de fase para atender ao requisito de erro em regime permanente (NISE, 2018). Ao se utilizar a função de transferência em malha aberta do sistema (6), com o fator de amortecimento, $\xi = 0,591$, operando com 10% de ultrapassagem e ângulo, $\phi_0 = 90^\circ$, gerou-se o lugar geométrico das raízes através do Matlab®, que resultou em um ganho $K = 0,00192$ e o polo desejado $-0,152 + j0,21$. A partir da parte real do polo, calculou-se o tempo de acomodação sem compensação como $T_s = 4/0,152 = 26,32$ s, ou seja, uma resposta muito lenta (NISE, 2018). Em seguida, reduziu-se o tempo de acomodação por um fator de 15, ou seja, $T_s = 4/(15 \times 0,152) = 1,75$ s. A nova parte real do polo desejado $-\xi\omega_n = -4/T_s = -2,28$. A nova parte imaginária do polo desejada obtida através do plano s , Figura 2 (a), resultou-se, $\theta_1 = 126,16^\circ$, portanto $\omega_d = tg(180^\circ - 126,16^\circ) \times 2,28 = j3,12$.

O zero do compensador de avanço de fase foi escolhido arbitrariamente, $z_c = 0,3$. Com base na análise nos polos da função de transferência (6), determinou-se a contribuição angular no ponto de projeto desse zero junto com os polos do sistema, Figura 2 (b), $\theta_1 = 126,16^\circ$, $\theta_2 = 123^\circ$ e $\theta_3 = 122,34^\circ$. Realizou-se a soma dos ângulos desse zero e dos polos e zeros do sistema sem compensação, considerando a diferença entre 180° , para encontrar a contribuição angular do polo do compensador (NISE, 2018). Deste modo, foi encontrado $\theta_4 = 54,5^\circ$, que é requerida a fim de determinar a posição do polo do compensador de avanço de fase. A Figura 2 (c), é utilizada para calcular a posição do polo do compensador, $tg 54,5^\circ = 3,12/(p_c - 2,28)$. A partir disso foi obtido o valor do polo do compensador $p_c = 4,51$. Gerou-se novamente o lugar geométrico das raízes através do Matlab®, com a função de transferência (6) junto com o compensador de avanço de fase, $G_{avanço}(s) = (s + 0,3)/(s + 4,51)$, que resultou em um novo ganho de $K = 0,415$.

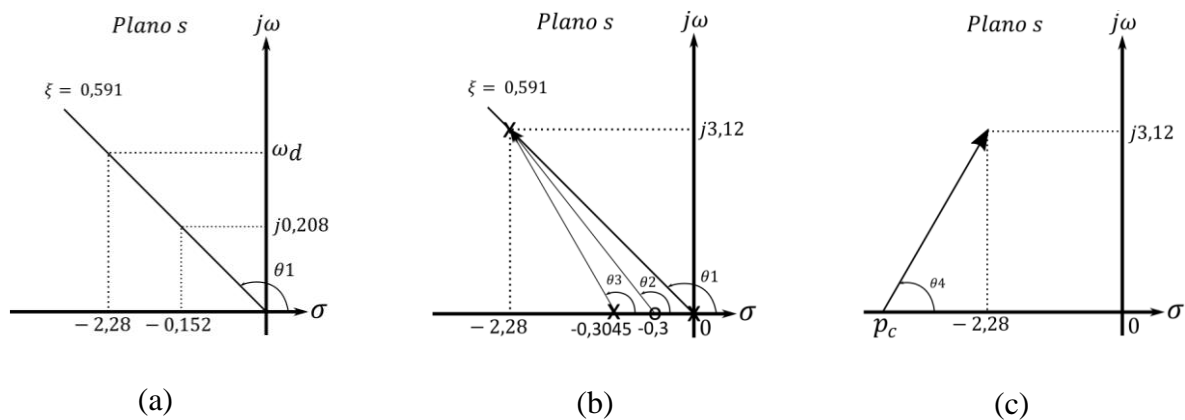


Figura 2 - (a) Diagrama para obter a parte imaginária da posição desejada, (b) Diagrama dos polos e zero do sistema e (c) Diagrama utilizado para calcular a posição do polo do compensador.

O compensador de atraso de fase foi projetado a partir de uma entrada degrau, $e(\infty) = 1/(1 + \lim_{s \rightarrow 0} G(s))$, encontrou-se $e(\infty) = 0,086$, melhorando esse valor em 10 vezes, ou seja, $e(\infty) = 10/0,086 \cong 116$. Escolheu-se arbitrariamente o polo perto da origem para não ocorrer muita influência sob o controlador, $p_c = 0,01$ (NISE, 2018). Calculando o zero através da razão requerida entre o zero do compensador e o polo do compensador, $z_c/p_c = K_{pn}/K_{po} = 116$, obteve-se $z_c = 116 \times p_c = 1,16$.

Com o compensador de avanço e atraso de fase projetados e com as equações de estado, foram implementados em diagrama de blocos no software Simulink/Matlab®, levando em consideração que $T(t) = \Delta_T + mlgd_c \text{sen}(\phi_0)$ e $\Delta_\phi = \phi - \phi_0$, o diagrama apresentado na Figura 3. A resposta do sistema a uma entrada degrau de 0,15 rad, considerando o modelo linearizado e o modelo não linear do sistema viga-hélice, é apresentado na Figura 4. Escolheu-se o pequeno degrau em virtude de uma resposta linear, que pode ser estabelecida se a faixa de variação dos valores ou se o valor de entrada em torno do ponto de operação for pequena (NISE, 2018).

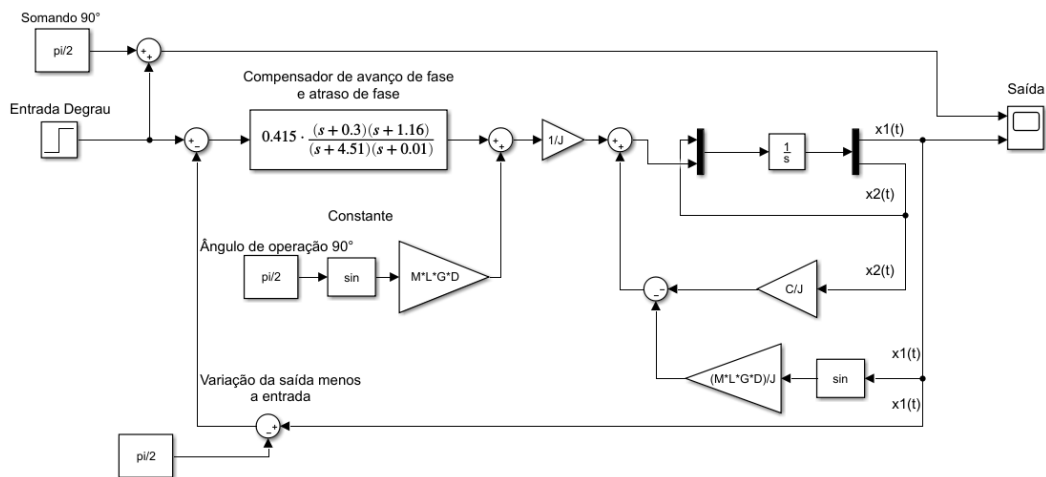


Figura 3 - Diagrama de blocos do compensador de avanço e atraso de fase com o sistema não linear.

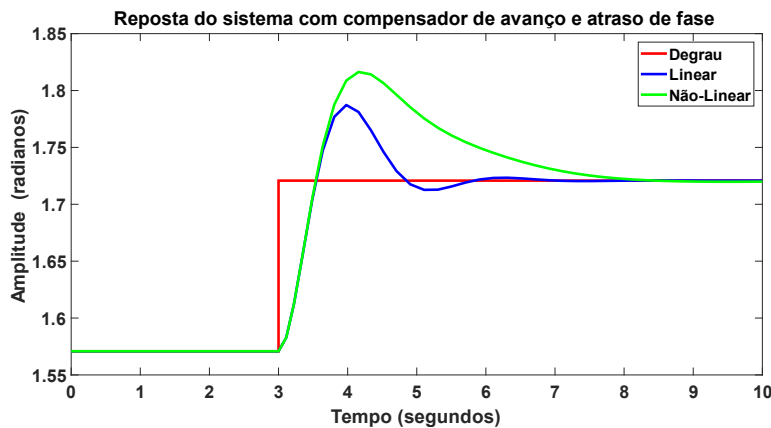


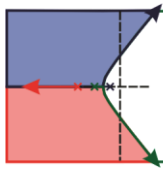
Figura 4 - Resposta a uma entrada degrau de valor 0,15 rad do sistema linear e não linear, ambos em malha fechada com o compensador projetado.

CONSIDERAÇÕES FINAIS

Este trabalho apresentou a modelagem matemática do sistema viga-hélice, tanto em espaço de estados quanto em função de transferência para projetar o compensador de avanço e atraso de fase. Resultados de simulação para a dinâmica não linear e linear mostraram que, embora diferentes, o compensador desempenhou satisfatoriamente os requisitos de projeto.

REFERÊNCIAS

- CHEN, C.-T. **Linear system theory and design**. Oxford: Oxford University Press, 1998.
- 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*, Piscataway, v. 55, n. 4, p. 538-545, 2012.
- JOB, M. M.; JOSE, P. S. H. **Modeling and control of mechatronic aeropendulum**. In: *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2015, Coimbatore. Anais... Piscataway: IEEE, 2015. p. 1-5.
- NISE, N. S. **Engenharia de Sistemas de Controle**. Rio de Janeiro. 7 Edição; LTC, 2018.
- OGATA, K. **Engenharia de Controle Moderno**. São Paulo. 5 Edição; Pearson, 2010.



ASPECTOS DE IMPLEMENTAÇÃO DE CONTROLADOR PI EM AMBIENTE SIMULADO

VARASCHIM, F. H.¹; ALMEIDA, J. P. L. S. de.²

¹ Discente do curso de Engenharia de Controle e Automação, IFPR, Jacarezinho, Paraná, e-mail: filipe_varaschim@hotmail.com

² Doutor em Engenharia Elétrica e Informática Industrial, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, e-mail: joao.almeida@ifpr.edu.br

RESUMO

Sistemas de controle e de automação estão cada vez mais presentes nas demandas industriais e, por este motivo, os conceitos que envolvem tais temas são importantes elementos de estudo e pesquisa em cursos de Engenharia de Controle e Automação, tanto quando considerados em experimentos em plantas reais, em escalas laboratoriais, quanto em simuladores. Neste contexto, este trabalho tem os objetivos de apresentar os principais aspectos de implementação de um controlador Proporcional-Integral no ambiente de simulação *Tinkercad*, por meio do *hardware* de controle Arduino, e mostrar as principais observações que podem ser efetuadas por um estudante/pesquisador. Os componentes utilizados são padrões do simulador. Para fins de validação, o controlador foi utilizado para o controle de um processo, representado neste trabalho por um sistema de primeira ordem. Os resultados mostram que a simulação é uma opção válida para o estudo de implementação de controladores PI.

PALAVRAS-CHAVE: Sistemas de Controle, Controle PI, Simulação, *Tinkercad*.

INTRODUÇÃO

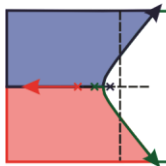
Com a industrialização cada vez mais exigente e avançada, produzindo mercadorias com maior qualidade, agilidade e eficácia, se faz necessário estabelecer maneiras eficientes de controlar os processos produtivos e, neste aspecto, a escolha de um controlador adequado pode ser crucial para que os requisitos operacionais sejam atingidos.

Um dos controladores mais conhecidos e usados em ambientes industriais é o do tipo Proporcional-Integral-Derivativo (PID), incluindo suas variações, P, PI e PD. A característica básica destes controladores, que os tornam muito populares, é o fato de quando adequadamente ajustados (manualmente ou automaticamente), eles geralmente conduzem o sistema a um desempenho satisfatório, em termos de redução de sobressinais, de erros em regime e tempo de acomodação (OGATA, 2003).

Neste trabalho, a implementação de um controlador do tipo PI será apresentada sob a perspectiva do uso de simuladores para tal finalidade. Neste contexto, considerou-se a necessidade de controlar a saída de um sistema de primeira ordem, que pode ser a representação de diversos tipos de processos encontrados em demandas industriais, tais como os que envolvem o controle de nível de fluido, de temperatura, de vazão e de pressão, dadas as devidas simplificações matemáticas acerca de suas características dinâmicas. A implementação do controlador e do sistema de primeira ordem foi realizada por meio do simulador *on-line Tinkercad* (*disponível em: <https://www.tinkercad.com/>*). Uma das principais contribuições deste trabalho é a de mostrar aspectos de implementação, ainda que em simulação, do controlador PI em um *hardware* de baixo custo e de fácil acesso por pesquisadores, que se trata da plataforma Arduino, além de afirmar que os ambientes de simulação podem ser uma opção válida para os estudos de sistemas de controle.

METODOLOGIA

De uma forma geral, a simulação realizada objetiva implementar um controlador PI para o controle da saída de um sistema de primeira ordem de representação genérica, como apresentado pela Equação (1), em que: K é o ganho do processo; τ é a constante de tempo; $Y(s)$ é a saída; e $U(s)$ é a entrada. Assume-se, como variável manipulada deste sistema, a velocidade de rotação de um motor DC com *encoder*, que pode representar, indiretamente, a velocidade de uma bomba centrífuga para o deslocamento



de fluido, o atuador de um sistema de refrigeração, etc. A saída do sistema (variável controlada) é uma medida que dependerá do processo a ser representado pela Equação (1) e, portanto, pode ser a medida de nível em um reservatório, a temperatura de um determinado objeto, a vazão de fluido em uma tubulação, entre outras possibilidades.

$$\frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \quad (1)$$

Inicialmente, para que o sistema da Equação (1) pudesse ser implementado no *hardware* digital de simulação, aplicou-se uma discretização na função de transferência considerando: $K = 1$, $\tau = 1$ e $dt = 0,1s$, com o método retentor de primeira ordem. O resultado da discretização é mostrado na Equação (2).

$$\frac{Y(z)}{U(z)} = \frac{0,04837z + 0,04679}{z - 0,9048} \quad (2)$$

Inicialmente, foi montado no simulador *Tinkercad* um circuito baseado em (PAVANI, 2021), mostrado na Figura 1, em que podem ser observados os seguintes componentes: uma placa Arduino UNO, uma matriz de contatos (*protoboard*), um motor DC com *encoder*, um circuito para o condicionamento do sinal do *encoder* (transistor TIP120, resistor de 10 k Ω e resistor de 5 Ω), uma fonte de tensão variável e conectores gerais.

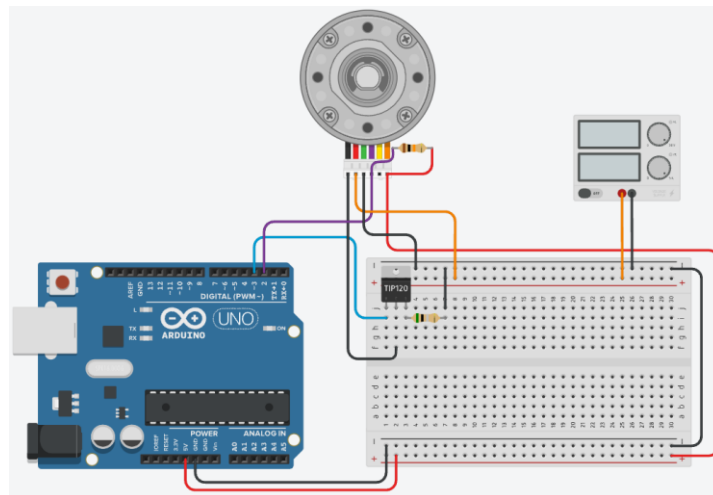


Figura 7 – Circuito eletrônico no simulador Tinkercad (adaptado de Pavani (2021)).

Em seguida, foi estruturado o algoritmo do controlador PI, o qual é mostrado nas Figuras 2 e 3. O algoritmo foi estabelecido de tal forma que o usuário estabeleça um valor para *setpoint* e para os ganhos K_p e K_i do controlador. Além disso, pode-se observar na linha 53 (Figura 3), que o sinal PWM é aplicado ao motor e, na linha 37 (Figura 3), que é realizado o cálculo da velocidade do motor, variável que é aplicada como entrada do sistema de primeira ordem discretizado, como mostrado na linha 39 (Figura 3). É importante ressaltar que, apesar da representação da planta a ser controlada (Equação (2)) ser incluída no algoritmo executado pelo próprio *hardware* Arduino, o que não é a forma mais adequada de simulação, demais parâmetros de implementação podem ser observados, tais como a utilização da linguagem difundida para plataformas Arduino, a geração de sinais PWM compatíveis com o atuador (motor DC), a utilização da instrução para a leitura do *encoder* (medida de velocidade do motor DC), entre outras.

```

1 // Mapeamento Hardware
2 #define PWM_OUT 3
3 #define ENCODER_A 2
4
5 int output; // define a saída de pwm
6 int output_last; // sinal de controle anterior
7 int speed; // velocidade do motor
8 int speed_last; // velocidade do motor anterior
9 int Y = 0; // sinal de saída da FT
10 int Y_last = 0; // sinal de saída anterior da FT
11 int error = 0; // erro atual
12 int lastError = 0; // declara o último erro
13 int setpoint = 300; // declara o setpoint
14 int I_error = 0; // ganho integral
15 float KP = 1; // constante proporcional
16 float KI = 0.1; // constante integrativa
17
18 void setup() {
19   Serial.begin(9600); // inicia a comunicação serial
20
21   pinMode(ENCODER_A, INPUT); // define o encoder para enviar dados
22   pinMode(PWM_OUT, OUTPUT);
23   // define a porta 3 para envio de sinal PWM para controle de velocidade
24
25   output = 10; // define um valor inicial para pwm_valor
26   output_last = output;
27   analogWrite(PWM_OUT, output); // controla o valor pwm através da saída porta 3
28
29   // realiza a leitura do encoder e transforma o valor dos pulsos em velocidade
30   speed = 19.1*((60*1000*10) / pulseIn(ENCODER_A, HIGH));
31   speed_last = speed;
32 }
33

```

Figura 2 – Algoritmo do controlador PI parte 1.

```

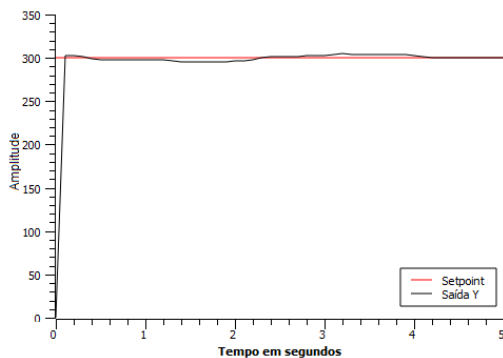
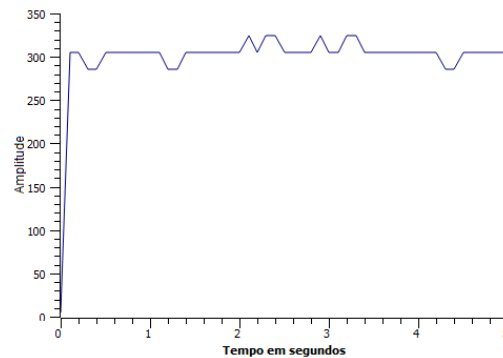
34 void loop() {
35
36   // realiza a leitura do encoder e transforma o valor dos pulsos em velocidade
37   speed = 19.1*((60*1000*10) / pulseIn(ENCODER_A, HIGH));
38
39   Y = 0.04837*speed + 0.04679*speed_last + 0.9048*Y_last;
40
41   error = setpoint - Y; // calcula o valor do erro
42
43   I_error += error; // somatório do erro
44
45   // Implementação do Algoritmo PI
46
47   output = KP*error + KI*I_error;
48
49   // limita a saída do controle para os valores válidos de pwm (10-255)
50   output = constrain(output, 10, 255);
51
52   // envia o sinal para o "motor"
53   analogWrite(PWM_OUT, output);
54
55   // Imprime Speed, Y e Setpoint
56   Serial.print(speed);
57   Serial.print("\n");
58   Serial.print(Y);
59   Serial.print("\n");
60   Serial.println(setpoint);
61
62   speed_last = speed;
63   Y_last = Y;
64
65   delay(100);
66 }

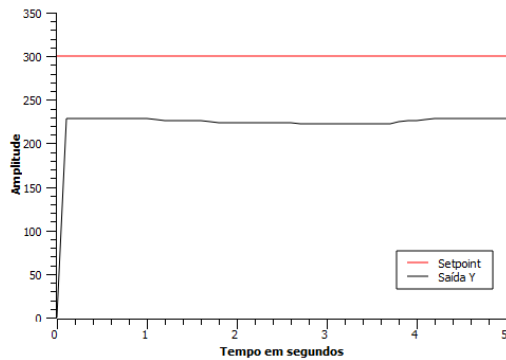
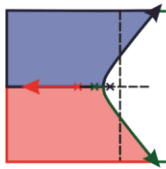
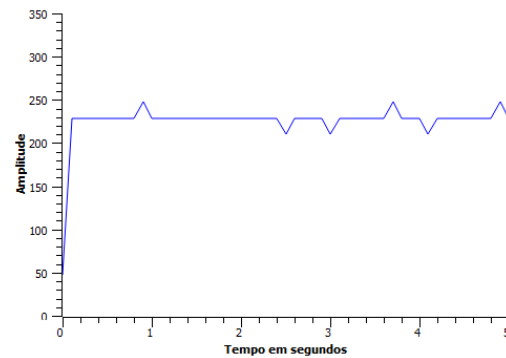
```

Figura 3 – Algoritmo do controlador PI parte 2.

RESULTADOS E DISCUSSÕES

Dada a utilização do algoritmo apresentado nas Figuras 2 e 3, os resultados de duas simulações são apresentados nas Figuras 4 – 7. Especificamente, nas Figuras 4 e 5 são apresentadas, respectivamente, a resposta do sistema considerando um controlador PI ($K_p = 1$ e $K_i = 0,1$) e a evolução da ação de controle ao longo do tempo. A fim de mostrar uma segunda possibilidade de simulação, para o mesmo algoritmo, o termo K_i foi anulado do controlador, com o objetivo de estabelecer um controlador P, cujos resultados de resposta do sistema e ação de controle são apresentados nas Figuras 6 e 7, respectivamente. No segundo experimento, nota-se que apenas a ação proporcional não foi suficiente para levar a saída do sistema ao valor de *setpoint*, o que representa uma situação de erro em regime.

Figura 4 – Resposta do sistema ($K_p = 1$ e $K_i = 0,1$).Figura 5 – Ação de controle ($K_p = 1$ e $K_i = 0,1$).

Figura 6 – Resposta do sistema ($K_p = 1$).Figura 7 – Ação de controle ($K_p = 1$).

CONSIDERAÇÕES FINAIS

O presente resumo expandido apresentou os principais aspectos de implementação, via simulador *Tinkercad*, de um controlador PI em uma malha de controle cujo processo a ser controlado foi representado por um sistema de primeira ordem. Nas simulações, foi considerado um conjunto de *hardware* de baixo custo e de fácil acesso por pesquisadores, que se trata da plataforma Arduino e de componentes comuns compatíveis.

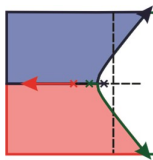
Para fins de validação foi apresentada a resposta do sistema de primeira ordem, sob atuação do controlador PI, para duas condições de ganhos de K_p e K_I . Os resultados mostraram uma resposta já esperada considerando este conjunto de aplicações, em relação ao erro em regime permanente, tempo de subida, entre outros conceitos abordado em teorias de controle.

Vale ressaltar que, como o foco deste trabalho foi de mostrar aspectos de implementação do controlador PI na plataforma Arduino, ou seja, uma etapa que vem após a aplicação de metodologias de projeto (sintonia e ganhos) deste tipo de controlador, as análises do ponto de vista de controle não foram consideradas, tais como a aplicação de métodos de sintonia, a experimentação com outros elementos dinâmicos no sistema (perturbação, ruído, características não-lineares, etc). Além disso, ainda que a representação da planta a ser controlada seja incluída no algoritmo executado pelo *hardware* de controle, o que não é um método convencional de simulação, pôde-se observar os principais aspectos de implementação do código em um *hardware* comum e de tecnologia aberta. Portanto, estas abordagens ausentes podem ser consideradas em trabalhos futuros.

REFERÊNCIAS

OGATA, Katsuhiko; **ENGENHARIA DE CONTROLE MODERNO**. Tradução Paulo Alvaro Maya; Revisão técnica Fabricio Leonardi, São Paulo, Prentice Hall, 2003.

PAVANI, V. **Controle PID Utilizando Arduino**. UFRJ Nautilus. Disponível em: <https://pt.ufrjnautilus.com/post/controle-pid-utilizando-arduino>. Acesso em: 20/10/2021.



PROJETO DE COMPENSADOR PARA CONVERSOR *Boost* UTILIZANDO O FATOR K

MARTINS, L. F. B.¹, BREGANON, Ricardo², ALMEIDA, J. P. L. S.³, ALVES, U. N. L. T.⁴

¹Doutor em Engenharia Elétrica, IFPR, Jacarezinho, e-mail: luis.martins@ifpr.edu.br

²Doutor em Engenharia Mecânica, IFPR, Jacarezinho, e-mail: ricardo.breganon@ifpr.edu.br

³Doutor em Engenharia Elétrica, IFPR, Jacarezinho, e-mail: joao.almeida@ifpr.edu.br

⁴Doutor em Engenharia Elétrica, IFPR, Jacarezinho, e-mail: uiliam.alves@ifpr.edu.br

RESUMO

Os conversores CC-CC têm desempenhado um papel fundamental para diferentes campos de aplicações. Grande parte destas aplicações exige uma fonte constante de tensão, independentemente de quaisquer variações de entrada ou de carga. O comportamento em regime permanente dessas fontes de alimentação são definidos pela topologia do conversor. Entretanto, o controlador tem um importante papel no desempenho dinâmico destes conversores. Algumas destas topologias como a *Boost* e a *Buck-Boost*, possuem um zero à direita do plano complexo, ou seja, há um problema de fase não mínima, o que restringe o uso de controladores do tipo PID. Por esta razão, é utilizado um controlador avanço-avanço, conhecido como controlador Tipo III, para melhorar o desempenho do sistema. Este trabalho investiga o desempenho de um controlador Tipo III para um conversor *Boost* em modo de controle de tensão.

PALAVRAS-CHAVE: conversor *Boost*. sistema de fase não mínima. controlador tipo III.

INTRODUÇÃO

Atualmente, existe uma infinidade de dispositivos eletrônicos que operam com tensão constante fornecida por um conversor CC-CC, como por exemplo, células fotovoltaicas, aplicações para carregamento de bateria, entre outras (MOHAN; UNDELAND; ROBBINS, 2003).

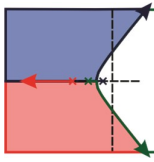
Geralmente, um conversor CC-CC pode ser classificado em três tipos: (i) Conversor *Buck* ou abaixador em que a tensão de saída é menor que a tensão de entrada, (ii) Conversor *Boost* ou elevador em que a tensão de saída é maior que a tensão de entrada; e (iii) Conversor *Buck-Boost*, um conversor que combina as características das topologias *Buck* e *Boost*, no qual a tensão de saída pode ser maior ou menor que a tensão de entrada (ROGERS, 1999).

O desempenho dos conversores CC-CC pode ser analisado de duas formas: (a) em regime transitório e (b) em regime permanente. O desempenho em regime permanente é definido principalmente pela topologia do conversor. Por outro lado, a estabilidade do conversor é mantida pela configuração do controlador. Os compensadores clássicos, entre eles o PI e o PID, foram desenvolvidos ao longo dos anos para garantir o desempenho desejado do conversor sob certas condições. Entretanto, alguns conversores, como o *Boost*, o *Buck-Boost* e o *Fly-Back* são sistemas de fase não mínima, ou seja, têm um zero do lado direito do plano complexo, o que torna difícil para controladores do tipo PID alcançar um desempenho satisfatório sob carga, variações de linha e incertezas paramétricas (OGATA, 2010; DORF; BISHOP, 2017). Por esta razão, os controladores Tipos I, II e III propostos em Venable (1983) são os mais adequados.

Compensadores Tipo II podem ser utilizados para obter uma resposta rápida e estável com pequeno erro de regime em conversores *Buck*. Contudo, este compensador pode inserir um avanço de fase máximo $\Phi_C = +90^\circ$, que pode ser insuficiente para algumas topologias de *Choppers*, entre elas a *Boost*. Para estes casos, controladores com capacidade de avanço na faixa $90^\circ < \Phi_C \leq 180^\circ$ como o de Tipo III podem ser necessários para manter uma margem de fase satisfatória.

Neste trabalho, o desempenho de um compensador Tipo III será avaliado em duas representações de um conversor *Boost*: uma linear, gerada à partir de um modelo médio à pequenas perturbações (MIDDLEBROOK; CUK, 1976) e outra; não linear, baseada no modelo chaveado do circuito elétrico do conversor, gerada pelo software Simulink®.

Além da introdução, este trabalho está dividido em mais quatro partes: na Seção 2 é apresentado o modelo médio do conversor *Boost*; na Seção 3 é exibida uma forma generalizada do compensador de Tipo III; os resultados do trabalho são discutidos na Seção 4 e, finalmente; na Seção 5 são abordadas as conclusões.



FUNÇÃO DE TRANSFERÊNCIA DO CONVERSOR *BOOST* PARA CONTROLE DE TENSÃO

Para controlar a tensão de saída do conversor, representado pela Figura 1a, em torno de um ponto de operação, é necessário definir uma função de transferência (FT) que relaciona a variável de entrada, neste caso a razão cíclica $d(t)$, com a tensão de saída v_o , que coincide com a tensão no capacitor v_C . Esta função de transferência linearizada $G_P(s)$, obtida a partir do modelo médio à pequeno sinais (MIDDLEBROOK; CUK, 1976; MIDDLEBROOK, 1988) do conversor *Boost* é apresentada na eq. (1).

$$G_P(s) = \frac{\Delta v_C(s)}{\Delta d(s)} = \frac{V_i}{(1-D)^2} \frac{\frac{1}{RC}s - \frac{1}{LC}(1-D)^2}{s^2 + \frac{1}{RC}s + \frac{1}{LC}(1-D)^2} \quad (1)$$

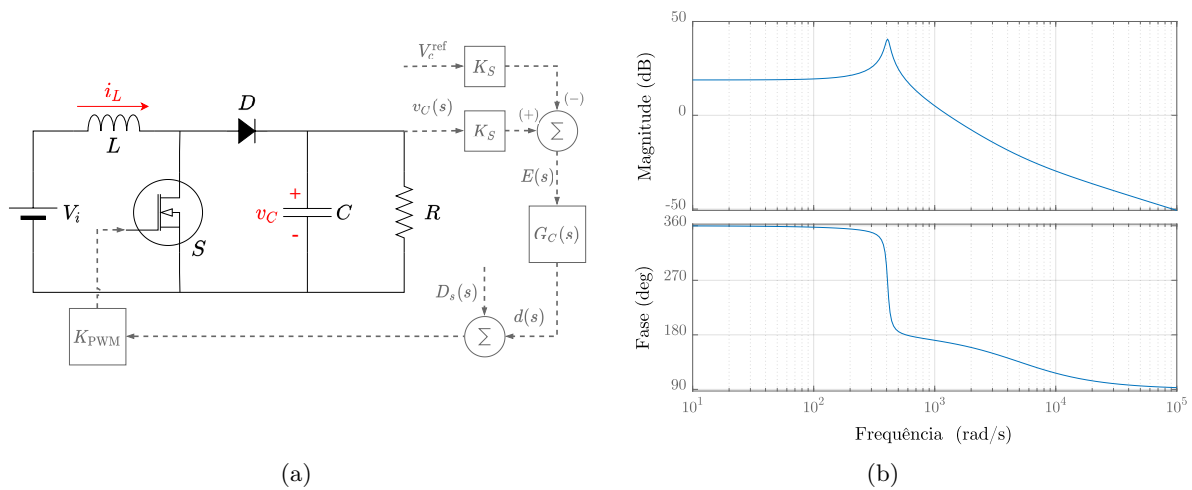


Figura 1 – a) Diagrama esquemático em malha fechada para o conversor *Boost*. b) Diagramas de Bode de magnitude e fase do conversor *Boost* com os parâmetros da Tab. (1).

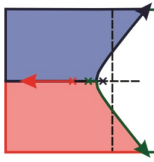
Em relação à eq. (1) e à Tab. (1), R , L , C e V_i são, nesta ordem, o resistor, indutor, capacitor e a tensão de entrada do conversor; V_c^{ref} é a tensão de referência; D e K_{PWM} são a razão cíclica e o ganho do modulador PWM; K_S é o ganho do sensor de tensão; G_C é a função de transferência do compensador e; D_s é uma perturbação na razão cíclica.

DEFINIÇÃO E PROJETO DO CONTROLADOR TIPO III

A escolha do controlador e o devido ajuste de seus parâmetros desempenha um papel fundamental para manter o bom desempenho dinâmico e a regulação da saída de um conversor CC-CC. O controlador tipo III é um controlador do tipo avanço-avanço com um pólo na origem, ou seja, fornece aumento de fase de 0° a 180° com erro nulo em regime. Mesmo que o conversor *Boost* seja de fase não mínima, com o ajuste adequado dos parâmetros do controlador, é possível obter um bom desempenho em malha fechada com o mínimo de sobressinal e erro nulo em regime permanente. A função de transferência $H_C(s)$ de um controlador Tipo III em sua forma generalizada e, na sua forma mais comum, com $\omega_{z1} = \omega_{z2} = \omega_z$ e $\omega_{p1} = \omega_{p2} = \omega_p$ são dadas pela eq. (2).

$$H_C(s) = \frac{\left(1 + \frac{s}{\omega_{z1}}\right) \left(1 + \frac{s}{\omega_{z2}}\right)}{\frac{s}{\omega_{p0}} \left(1 + \frac{s}{\omega_{p1}}\right) \left(1 + \frac{s}{\omega_{p2}}\right)} = \frac{\left(1 + \frac{s}{\omega_z}\right)^2}{\frac{s}{\omega_{p0}} \left(1 + \frac{s}{\omega_p}\right)^2} \quad (2)$$

O Fator k é um conceito matemático originalmente idealizado para auxiliar no projeto de amplificadores tipos I, II e III. Ele permite que o projetista calcule de forma simplificada seus componentes à partir da frequência



de cruzamento ω_c e da margem de fase P_m do sistema em laço aberto $G_L(s) = K_{PWM} \cdot G_P(s) \cdot K_S$. O ganho K_C e o avanço de fase φ_C necessários em ω_c são determinados pelas eqs. (3) e (4), onde $|H_C(j\omega_c)|$ e $\varphi_{H_C}(j\omega_c)$ são a magnitude e o ângulo, respectivamente, da FT do compensador, dada por (2), fazendo $s = j\omega_c$.

$$K_C = [K_{PWM} |H_C(j\omega_c)|]^{-1} \quad (3)$$

$$\varphi_C = P_m - \varphi_{H_C}(j\omega_c) - 90^\circ \quad (4)$$

O ângulo de fase de $H_C(s)$ na eq. (2) atinge seu máximo em ω_c , que coincide com a média geométrica das frequências dos pares pólo-zero $\sqrt{\omega_z \cdot \omega_p}$ da FT do compensador. Assim, o fator k pode ser compreendido como uma separação geométrica entre os pólos e zeros para produzir o avanço de fase necessário para atender as especificações do projeto. De forma mais clara, para amplificadores tipo III, k é definido como sendo a razão entre as frequências angulares dos pares pólo-zero de $H_C(s)$, ou seja, $k = \omega_p/\omega_z$. O fator k também pode ser relacionado ao avanço de fase φ_C inserido no sistema. Para o controlador tipo III, k pode ser escrito de acordo com a eq. (5) (VENABLE, 1983). Além disso, as frequências dos pólos e zeros de $H_C(s)$, ω_{p0} , ω_p e ω_z , podem ser escritas em função do fator k como mostrado no conjunto de eqs. (6) – (8).

$$k = \tan^2 [0,25 (\varphi_C + \pi)] \quad (5)$$

$$\omega_p = \sqrt{k}\omega_c \quad (6)$$

$$\omega_z = \left(\sqrt{k}/k\right)\omega_c \quad (7)$$

$$\omega_{p0} = K_C(\omega_c/k) \quad (8)$$

RESULTADOS E DISCUSSÕES

São apresentados na Tab. 1 os parâmetros de projeto do conversor *Boost* proposto neste trabalho. Sua FT de laço aberto $G_L(s) = K_S \cdot G_P(s)$, obtida à partir do modelo médio à pequenas perturbações, apresenta os diagramas de bode de magnitude e de fase mostrados na Fig. 1b e, a partir destes, são obtidas as margens de ganho $G_m = -18,8$ dB e de fase $P_m = -12,9^\circ$ e, as frequências de cruzamento de ganho $\omega_{cg} = 577$ rad/s e de fase $\omega_{cp} = 1,293$ krad/s. A análise destes resultados indica que o sistema em malha fechada é instável, justificando o uso de um compensador.

Tabela 1 – Parâmetros de projeto do conversor *Boost*.

V_i (V)	v_o (V)	D	R (Ω)	L (mH)	C (μF)	K_{PWM}	K_s	Frequência do PWM f_{sh} (kHz)
100	250	0,60	189,4	6,1	158,4	1,000	0,014	20

O ajuste dos pólos e zeros do controlador tipo III se inicia com a definição de ω_c que, de modo geral, deve ser inferior a 10% da frequência de chaveamento f_{sh} . Além desta recomendação, existem outras duas. A primeira é limitar a frequência de cruzamento ao máximo de 30% da frequência do zero do conversor, ou seja, $\omega_c < 0,3 \frac{R}{L} (1 - D)^2$. A segunda, é fazer com que ω_c seja maior que a frequência de ressonância ω_o do conversor, caso contrário, não haverá ganho suficiente para eliminar os efeitos da ressonância e, mesmo com uma boa margem de fase, o sistema poderá ficar instável. Desta forma, $\omega_c > 3(1 - D)(LC)^{-\frac{1}{2}}$ será uma boa escolha (BASSO, 2012). Partindo destas condições, foi escolhido $\omega_c = 1,36$ krad/s, ou seja, a média do intervalo $]1224, 1500[$ rad/s calculado seguindo as recomendações anteriores.

Aplicando ω_c nas eqs. (3) e (4), e $P_m = 60^\circ$ na eq. (4) são determinados $K_C = 1,117$ e $\varphi_C = -163,7^\circ$. A partir daí, são utilizadas as eqs. (5) – (8) para calcular o fator $k = 197,0$ e as frequências angulares $\omega_z = 97,0$ rad/s, $\omega_{p0} = 7,73$ rad/s e $\omega_p = 19,1$ krad/s. É mostrado na Fig. 2a os diagramas de Bode para a FT de laço aberto com a inclusão de H_C , $G_{LC}(s) = G_L(s) \cdot G_C(s)$, e na Fig. 2b a resposta no tempo considerando perturbações em torno do ponto de operação ($V_c^{ref} = 250$ V) para os modelos não linear e linearizado. No detalhe da Fig. 2b é exibido o efeito do chaveamento em $v_C(t)$ para o modelo não linear.

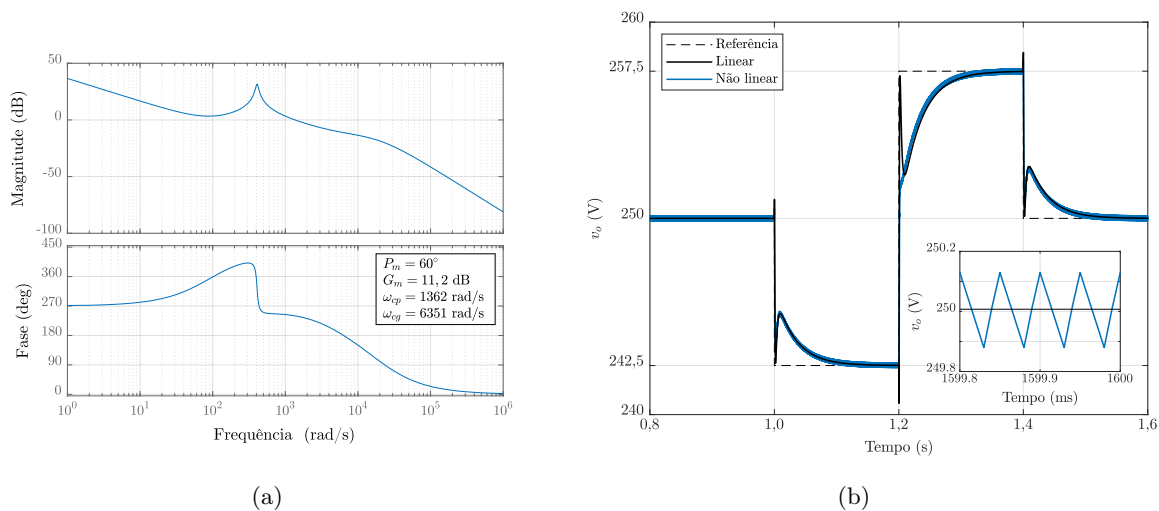


Figura 2 – a) Diagramas de Bode de magnitude e fase de $G_{LC}(s)$. b) Resposta da tensão no capacitor $v_C(t)$ à variações em degrau na tensão de referência V_C^{ref} .

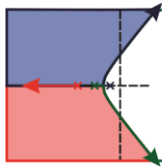
A análise dos gráficos de Bode da Fig. 2a mostra que $G_{LC}(s)$ tem $P_m = 60^\circ$ em $\omega_{cp} = 1,392$ krad/s e $G_m = 11,2$ dB em $\omega_{cp} = 6,351$ krad/s. A partir dos valores das margens de fase e de ganho, o sistema em MF é considerado estável. Para validar esta conclusão, são examinadas as resposta no tempo da saída dos modelos linearizado e chaveado (não linear) do conversor utilizando o compensador projetado, Fig. 2b, que indicam um bom desempenho de $v_C(t)$ frente à pequenas variações em degrau da tensão de referência nos dois casos.

CONSIDERAÇÕES FINAIS

Neste artigo, foi apresentada uma metodologia para o projeto de controlador Tipo III em malha fechada para um conversor CC-CC *Boost* linearizado a partir do modelo médio à pequenos sinais. Mesmo o sistema sendo de fase não mínima, o comportamento da tensão de saída em malha fechada com o controlador foi satisfatório para ambos os modelos estudados. Porém, é razoável supor que é possível obter melhores resultados com o ajuste fino dos parâmetros do controlador e essa investigação pode ser tema de trabalhos futuros.

REFERÊNCIAS

- BASSO, C. *Designing Control Loops for Linear and Switching Power Supplies : a Tutorial Guide*. Norwood, GA: Artech House, 2012. ISBN 9781608075577. 3
- DORF, R. C.; BISHOP, R. H. *Modern Control Systems*. 13th. ed. USA: Prentice-Hall, Inc., 2017. ISBN 0134407628. 1
- MIDDLEBROOK, R. Small-signal modeling of pulse-width modulated switched-mode power converters. *Proceedings of the IEEE*, v. 76, n. 4, p. 343–354, 1988. 2
- MIDDLEBROOK, R. D.; CUK, S. A general unified approach to modelling switching-converter power stages. In: *1976 IEEE Power Electronics Specialists Conference*. [S.l.: s.n.], 1976. p. 18–34. 1, 2
- MOHAN, N.; UNDELAND, T. M.; ROBBINS, W. P. *Power electronics: converters, applications, and design*. Hoboken, NJ: John Wiley & Sons, 2003. ISBN 9780471226932. 1
- OGATA, K. *Modern control engineering*. 5th. ed. Boston, MA: Prentice-Hall, Inc., 2010. 894 p. ISBN 9780136156734. 1
- ROGERS, E. *Understanding Buck-Boost Power Stages in Switch Mode Power Supplies*. [S.l.], 1999. 1
- VENABLE, H. D. The k factor: A new mathematical tool for stability analysis and synthesis. In: *Proceedings of Powercon: International Solid-State Power Electronics Conference*. [S.l.: s.n.], 1983. 1, 3



PROTÓTIPO PARA AUTOMAÇÃO RESIDENCIAL UTILIZANDO ARDUINO

TAVARES, Ana Carolina Crespo Alves¹; AMARAL, Daniel Nascimento²

¹ Graduanda em Engenharia Elétrica, IFF - Macaé, Rio de Janeiro. E-mail: ana.tavares@gsuite.iff.edu.br

² Graduando em Engenharia Elétrica, IFF - Macaé, Rio de Janeiro. E-mail: n.amaral@gsuite.iff.edu.br

RESUMO

Com os avanços da indústria tecnológica, vários dispositivos eletrônicos que possibilitam o controle de sistemas residenciais, seja por computador pessoal ou por *smartphone*, estão hoje disponíveis no mercado. A expansão da Internet das Coisas (IoT) é um dos fatores que tem contribuído significativamente para os avanços nos recursos tecnológicos para usos residenciais. Entre os diversos sistemas disponíveis, neste artigo, é apresentado um protótipo para automação de uma residência, desenvolvido com auxílio da placa Arduino, para comando e monitoramento de dispositivos domésticos através de controle remoto, computador, *smartphone* ou *tablet*, possibilitando ao usuário conhecer e aplicar a forma que lhe proporciona maior conforto e praticidade.

PALAVRAS-CHAVE: Arduino, Automação Residencial, Microcontroladores, Comunicação.

INTRODUÇÃO

O conceito de automação surgiu na década de 1970, quando os primeiros sistemas automatizados de controle foram criados para aplicações industriais nos Estados Unidos. A cada ano, desde o surgimento da automação residencial, novas soluções são lançadas no mercado, sempre acompanhando as tendências de consumo. Um sistema automatizado não apenas gera comodidade e conforto, como segurança e sustentabilidade.

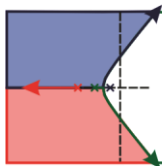
A aplicação de tecnologias IoT (Internet das Coisas), amplamente disseminadas nos últimos anos, adiciona a possibilidade de conectar dispositivos, intensificando ainda mais o mercado da automação residencial. A literatura aponta potencial utilização desta tecnologia em equipamentos e aparelhos presentes em residências, visto que esta é capaz de proporcionar facilidade à comunicação entre homem e máquina (BONINO et al., 2012).

A presença da domótica em residências brasileiras ainda é pouco expressiva. Segundo pesquisa publicada pela Associação Brasileira de Automação Residencial e Predial (AURESIDE), apenas 0,5% das residências brasileiras possuem algum tipo de automação (NETO et al., 2018). A lenta expansão da automação residencial no Brasil está relacionada aos custos, que costumam ser elevados.

Este trabalho apresenta um protótipo de baixo custo para automação residencial construído com auxílio da placa Arduino, plataforma *web* e aplicativo *App Inventor 2*. A domótica proposta neste protótipo possibilita o controle e monitoramento de sistemas elétricos presentes em uma residência composta por dois quartos, sala, cozinha, banheiro, área de serviço e garagem com portão eletrônico. O acesso à entrada principal da casa é feito com fechadura eletrônica. A iluminação dos cômodos, portão eletrônico e ar-condicionado podem ser acionados por controle remoto, página *web* ou aplicativo de *smartphone*. Através de um *click*, é possível acessar os sistemas, verificar o estado e alterar a condição de operação dos dispositivos.

METODOLOGIA

A verificação do funcionamento do protótipo é inicialmente feita por meio de uma ferramenta *on-line*, desenvolvida pela Autodesk (<https://www.tinkercad.com>), que permite a simulação dos circuitos eletrônicos que compõem o sistema. Nesta etapa, são previstos os componentes a serem empregados, suas ligações elétricas e o código fonte que será executado pelo microprocessador do Arduino. A programação é feita em C++ adaptada para Arduino.



O acionamento dos sistemas presentes no protótipo pode ser feito a partir de um aplicativo móvel, disponível em Android ou IOS, desenvolvido na plataforma *App Inventor 2* da MIT (Instituto de Tecnologia de Massachussetts). Esta ferramenta possibilita a criação de aplicativos com base em programação em blocos, eliminando a necessidade de conhecimentos avançados em desenvolvimento de aplicativos (COSTA e PIEDADE, 2021).

A comunicação entre a placa Arduino e o aplicativo móvel é viabilizada pela placa *Ethernet Shield W5100*, dispositivo que, ao ser acoplado ao Arduino, permite conexão com a rede local por meio dos protocolos de comunicação TCP e UDP. É elaborada uma página *web*, construída em HTML, por onde também podem ser feitos os acionamentos e monitoramento dos sistemas.

DISCUSSÕES

Primeiramente, o protótipo é simulado por meio do simulador virtual *TinkerCad*. A montagem dos circuitos é feita em duas etapas: a primeira abrange o acionamento do sistema iluminação, ar-condicionado e motor do portão eletrônico; a segunda compreende o funcionamento da fechadura eletrônica, conforme exibem respectivamente as Figuras 1 e 2. A separação em duas etapas deve-se a limitação da plataforma de simulação possuir apenas o Arduino Uno, que dispõe de treze portas digitais e cinco portas analógicas, não atendendo a necessidade de projeto.

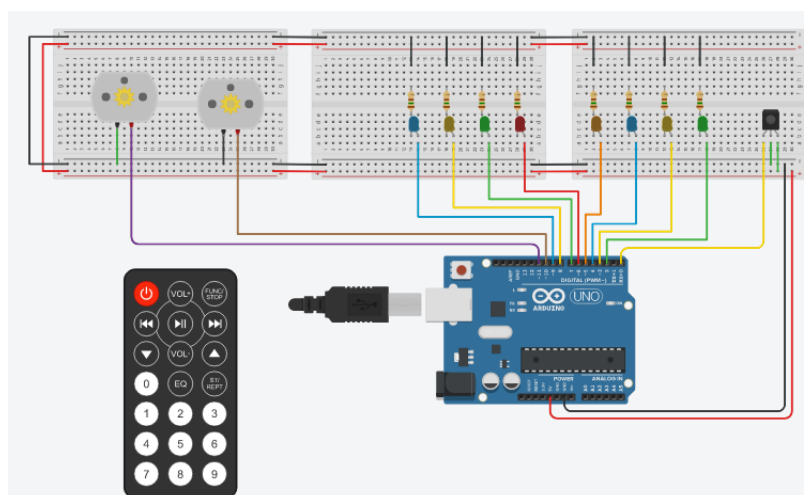


Figura 1. Circuito de acionamento das lâmpadas dos cômodos, portão eletrônico e ar-condicionado.

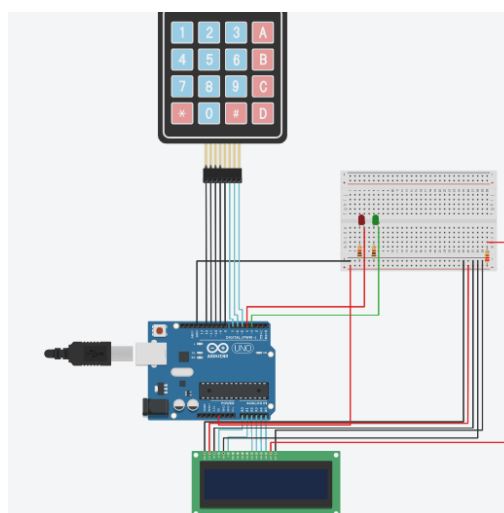


Figura 2. Circuito de acionamento da fechadura eletrônica.

Para simulação do acionamento do sistema de iluminação, portão eletrônico e ar-condicionado, Figura 1, considera-se, para fins de visualização, a implementação de dois motores que representam o funcionamento do portão eletrônico, diodos emissores de luz (LED) indicativos de iluminação dos ambientes e funcionamento do ar-condicionado. O código de programação implementado permite o acionamento dos dispositivos por controle remoto, conforme indica a Tabela 1. Para simulação da fechadura eletrônica, Figura 2, aplica-se um sistema com senha de seis dígitos para liberação da porta, utiliza-se um display de cristal líquido, LEDs para indicar fechamento e abertura da porta e um teclado matricial.

Tabela 1. Descrição das teclas do controle remoto.

TECLA DO CONTROLE REMOTO	DISPOSITIVO	LED/PORTA ARDUÍNO SIMULAÇÃO
1	Iluminação Quarto 1	LED 1 – Porta D2
2	Iluminação Quarto 2	LED 2 – Porta D3
3	Iluminação Sala	LED 3 – Porta D4
4	Iluminação Cozinha	LED 4 – Porta D5
5	Iluminação Varanda	LED 5 – Porta D6
6	Iluminação Garagem	LED 6 – Porta D7
7	Ar-Condicionado Quarto 1	LED 7 – Porta D8
8	Ar-Condicionado Quarto 2	LED 8 – Porta D9
Esquerda	Abre Portão Eletrônico	LED 9 – Porta D10
Direita	Fecha Portão Eletrônico	LED 10 – Porta D11

Após simulação em ambiente virtual, o protótipo foi montado, conforme exibe Figura 3. O sistema de iluminação, ar-condicionado e portão eletrônico é montado na placa Arduino Mega, enquanto o sistema de porta com fechadura eletrônica em um Arduino Uno. A utilização do Arduino Mega deve-se ao fato da disponibilidade de portas de saída ser maior que no Arduino Uno. Para comunicação dos sistemas via rede de *internet*, foi necessária utilização da placa de *Ethernet Shield W5100*, que requer cinco portas para sua operação, pinos 4, 10, 11, 12 e 13. Somado a isso, também foi verificado que a memória interna do Arduino Mega, por ser maior que a do Uno, possibilita uma página HTML para acionamento e monitoramento das saídas com mais recursos.



Figura 3. Montagem do protótipo. (a) Circuito de acionamento e controle da iluminação, ar-condicionado e portão com Arduino Mega. (b) Circuito de acionamento e controle da fechadura eletrônica com Arduino Uno.

O código de programação executado no Arduino Mega é desenvolvido para implementar um algoritmo que seja capaz de ligar ou desligar as lâmpadas, portão e ar-condicionado. Destaca-se que, nesse sketch, foram inseridas bibliotecas e comandos específicos para comunicação da placa *Ethernet Shield* e controle remoto. A partir do código fonte, é gerada uma página *web* usando HTML, exibida na

Figura 4(b), por onde também é possível realizar os acionamentos e verificar estados dos dispositivos. O código implementado no Arduino Uno é responsável pelo funcionamento da fechadura eletrônica, verificando se a senha digitada é válida.

Em paralelo, foi feito o desenvolvimento da aplicação para dispositivos móveis a partir da plataforma *App Inventor 2*. Neste aplicativo, são disponibilizados botões para acionamentos das lâmpadas, portão eletrônico e ar-condicionado em uma interface gráfica, conforme ilustra a Figura 4(a). A programação é feita em blocos, incluindo a lógica para comunicação do aplicativo à página *web* hospedada no *Ethernet Shield*.

Por fim, após a montagem física dos componentes, foi realizado o *upload* dos *sketches* nas placas de Arduino, sendo possível verificar a operação da fechadura eletrônica com senha, no Arduino Uno, e dos acionamentos de iluminação, ar-condicionado e portão eletrônico, no Arduino Mega, via controle remoto, página HTML ou aplicativo móvel.

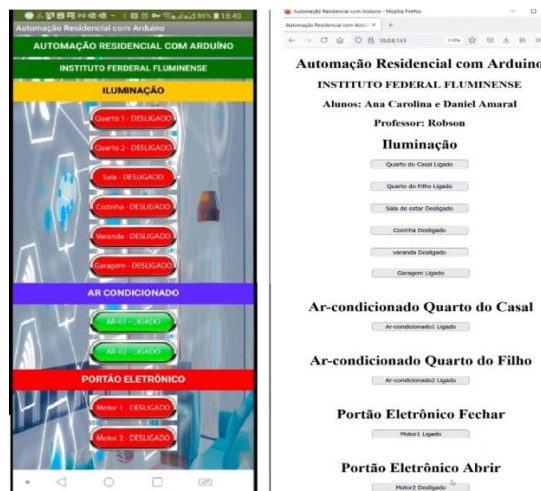


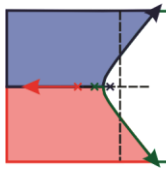
Figura 4. (a) Tela do aplicativo gerado para Android e IOS. (b) Página *web* em HTML criada para acionamento e monitoramento.

CONSIDERAÇÕES FINAIS

O protótipo apresentado neste documento possibilita ao aluno desenvolver conhecimentos em diversas áreas da automação residencial. A utilização dos sistemas de comunicação para visualização e comando utilizados (aplicativo móvel e página HTML), além de facilitar a interface do usuário com os sistemas, possui baixo custo de implementação, quando comparados a sistemas convencionais compostos por controladores lógicos programáveis e sistemas de supervisórios. Desta forma, o protótipo apresentado demonstrou ser uma solução simples, de baixo custo e fácil aplicação, podendo ser implementada em residências ou adaptado para controlar um sistema ou processo em ambientes industriais.

REFERÊNCIAS

- BONINO, Dario; CORNO, Fulvio; DE RUSSIS, Luigi. **Home energy consumption feedback: A user survey**. *Energy and Buildings*, v. 47, p. 383-393, 2012.
- NETO, José Aprígio Carneiro; JÚNIOR, Lúcio da Silva Gama; CARNEIRO, Luiz Felipe Costa Silva. **Um mapeamento tecnológico da Domótica no Brasil**. Anais do IX Simpósio Internacional de Inovação Tecnológica. Aracaju, 2018.
- COSTA, Rosana Gomes; PIEDADE, João Manuel Nunes. **Uso do aplicativo MIT app inventor na aprendizagem de programação: uma revisão sistemática da literatura entre 2011 e 2020**. *Revista Intersaberes*, v. 16, n. 37, p. 160-177, jan. 2021.



MODELAGEM E ANÁLISE DINÂMICA DE UM SISTEMA VIGA-HÉLICE

BISPO, Carlos A. S¹; YAMANAKA, Hugo F²; BREGANON, Ricardo³; ALMEIDA, João. P. L. S⁴; ALVES, Uiliam N. L. T⁵

¹ Discente em Engenharia de Controle e Automação, IFPR, Jacarezinho, Paraná, carlosalexandrehk@gmail.com

² Discente em Engenharia de Controle e Automação, IFPR, Jacarezinho, Paraná, hugoferndoyamanaka@gmail.com

³ Doutor em Engenharia Mecânica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, ricardo.breganon@ifpr.edu.br

⁴ Doutor em Engenharia Elétrica e Informática Industrial, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, joao.almeida@ifpr.edu.br

⁵ Doutor em Engenharia Elétrica, Instituto Federal do Paraná (IFPR), Jacarezinho, Paraná, uiliam.alves@ifpr.edu.br

RESUMO

Pêndulos são sistemas bem conhecidos pela engenharia de controle e possuem diversas variações. O presente trabalho apresenta a obtenção de um modelo matemático e a análise da dinâmica de um sistema Viga-hélice. O sistema mecânico Viga-hélice é um modelo de pêndulo que naturalmente possui dinâmica não linear, e seu funcionamento consiste na variação e estabilização da posição angular de uma haste fixa a um eixo de rotação, com um conjunto motor-hélice em sua outra extremidade. O modelo do sistema foi linearizado próximo a um ponto de operação utilizando a série de Taylor, e sua função de transferência foi obtida. A análise da dinâmica do sistema foi desenvolvida variando os parâmetros do sistema com objetivo de identificar com quais parâmetros o sistema tem um melhor desempenho e é mais estável.

PALAVRAS-CHAVE: Viga-hélice, análise dinâmica, sistema não linear, linearização.

INTRODUÇÃO

Os sistemas do tipo pêndulo são problemas clássicos estudados na área de controle. Esses sistemas são naturalmente não lineares e podem ser aproximados por um modelo linearizado (JOB; JOSE, 2015). O sistema Viga-hélice é um sistema do tipo pêndulo, que consiste em uma hélice acoplada ao eixo de um motor para produzir uma força de empuxo a fim de mover a haste para a posição desejada (SILVA et al, 2021). Utilizando a série de Taylor, foi obtido um modelo matemático linearizado para o sistema próximo a um ponto de operação. A análise da dinâmica do sistema consiste em calcular e comparar os diferentes valores que os polos do sistema assumem com a variação dos parâmetros físicos do protótipo do Viga-hélice, tais como massa do motor, a massa e o comprimento da haste, e o coeficiente de amortecimento viscoso. Tal análise tem como objetivo identificar com quais parâmetros o sistema possui um melhor desempenho e resposta.

SISTEMA VIGA-HÉLICE

O sistema Viga-hélice é um sistema mecânico que consiste em uma haste com rotação em torno de um eixo fixo. A extremidade da haste possui um conjunto motor hélice que proporciona o empuxo necessário para a variação do ângulo ϕ . Na Figura 1 é possível observar o diagrama de um sistema Viga-hélice e um protótipo deste tipo de sistema que está no laboratório de automação do IFPR, campus Jacarezinho.

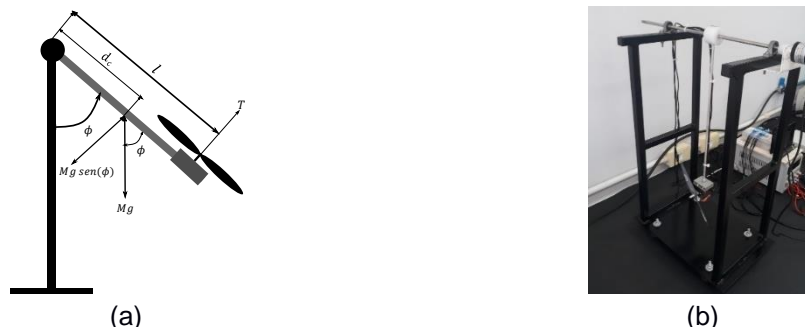
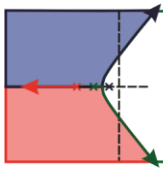


Figura 1 - (a) diagrama de corpo livre do sistema Viga-hélice e (b) protótipo Viga-hélice.



A partir do diagrama do sistema Viga-hélice apresentado na Figura 1 (a), fazendo o somatório das forças e torques em relação ao ponto de rotação, é possível obter a equação diferencial do sistema:

$$J \frac{d^2 \phi}{dt^2} + c \frac{d\phi}{dt} + mlgd_c \text{sen}(\phi) = T. \quad (1)$$

Sendo l a distância entre o conjunto motor hélice e o ponto de rotação, J o momento de inércia do sistema, c o coeficiente de amortecimento viscoso presente no eixo de rotação, T o torque exercido pelo motor, d_c a posição do centro de massa do sistema e m a massa do sistema.

Linearizando o termo não linear através da série de Taylor, e considerando ϕ_0 um ângulo constante como um ponto de operação para o sistema, tem-se a seguinte aproximação:

$$\text{sen}(\phi_0) \approx \text{sen}(\phi_0) + \frac{d\text{sen}(\phi_0)}{dt} \frac{\phi - \phi_0}{1!}. \quad (2)$$

Fazendo $\Delta_\phi = \phi - \phi_0$ e tomando sua derivada tem-se

$$\frac{d\Delta_\phi}{dt} = \frac{d\phi}{dt}. \quad (3)$$

Substituindo (2) e (3) em (1), segue que

$$\begin{aligned} J \frac{d^2 \Delta_\phi}{dt^2} + c \frac{d\Delta_\phi}{dt} + mlgd_c [\text{sen}(\phi_0) + \cos(\phi_0)\Delta_\phi] &= T. \\ J \frac{d^2 \Delta_\phi}{dt^2} + c \frac{d\Delta_\phi}{dt} + mlgd_c \cos(\phi_0)\Delta_\phi &= T - mlgd_c \text{sen}(\phi_0). \end{aligned} \quad (4)$$

Fazendo a substituição $\Delta_T = T - mlgd_c \text{sen}(\phi_0)$ em (4) resulta em

$$J \frac{d^2 \Delta_\phi}{dt^2} + c \frac{d\Delta_\phi}{dt} + mlgd_c \cos(\phi_0)\Delta_\phi = \Delta_T. \quad (5)$$

Aplicando a transformada de Laplace (NISE, 2018, p. 28) em (5) é obtida a expressão no domínio da frequência (6) e posteriormente organizando os termos de entrada do sistema sobre a saída do sistema, é obtida a função de transferência do sistema dado em (7).

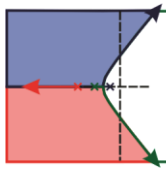
$$Js^2 \Delta_\phi(s) + cs \Delta_\phi(s) + mlgd_c \cos(\phi_0) \Delta_\phi(s) = \Delta_T(s). \quad (6)$$

$$\frac{\Delta_\phi}{\Delta_T(s)} = \frac{1}{s^2 + \frac{cs}{J} + \frac{mlgd_c \cos(\phi_0)}{J}}. \quad (7)$$

O protótipo físico do sistema Viga-hélice possui os parâmetros descritos na Tabela 1.

Tabela 1 - Parâmetros do sistema Viga-hélice.

Símbolo	Descrição	Valor	Unidade
g	Aceleração da gravidade	9,81	m/s^2
m_1	Massa do conjunto hélice motor	0,295	kg
m_2	Massa da haste	0,020	kg
l	Comprimento da haste	0,306	m
c	Coeficiente de amortecimento viscoso	0,0086	Nms/rad
$J = \frac{m_1}{3} l^2 + m_2 l^2$	Momento de inércia do sistema	0,0282	$kg.m^2$
d_c	Posição do centro de massa do sistema	0,2963	m



ANÁLISE DO SISTEMA E DISCUSSÕES

A fim de analisar a estabilidade e dinâmica do sistema Viga-hélice, foram considerados diferentes valores de parâmetros, sendo eles a massa do motor entre $0,050\text{ kg}$ e $0,350\text{ kg}$, comprimento da haste entre $0,20\text{ m}$ e $0,40\text{ m}$, o ângulo de operação entre 10° e 90° , coeficiente de amortecimento viscoso entre $0,0086\text{ Nms/rad}$ e 1 Nms/rad , e a massa da haste com valores entre $0,010\text{ kg}$ e $0,100\text{ kg}$. Desta forma, foram calculados os polos do sistema considerando os demais parâmetros dados na Tabela (1) e com os parâmetros nas faixas de valores escolhidos. É possível observar a influência da variação de cada parâmetro na dinâmica do sistema nas Figuras 2, 3 e 4.

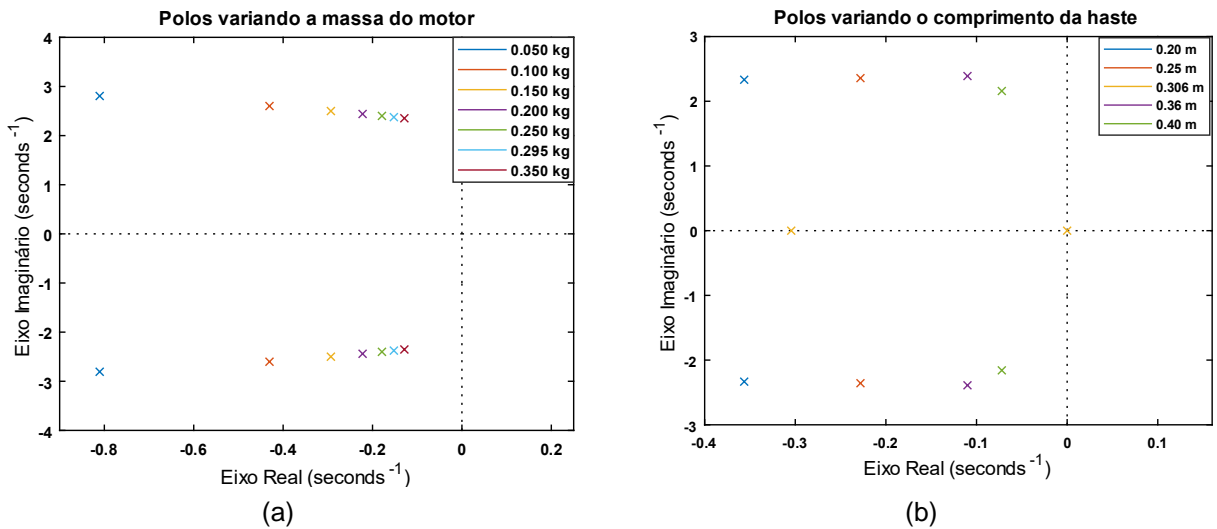


Figura 2 - Polos do sistema variando: (a) a massa do motor e (b) comprimento da haste.

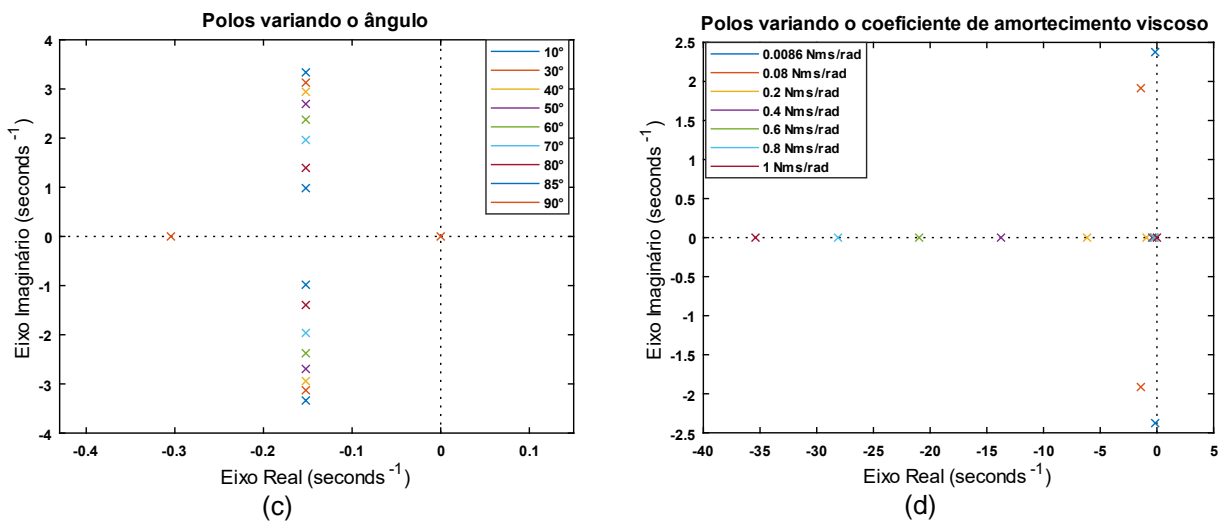


Figura 3 - Polos do sistema variando: (c) o ângulo e (d) coeficiente de amortecimento viscoso.

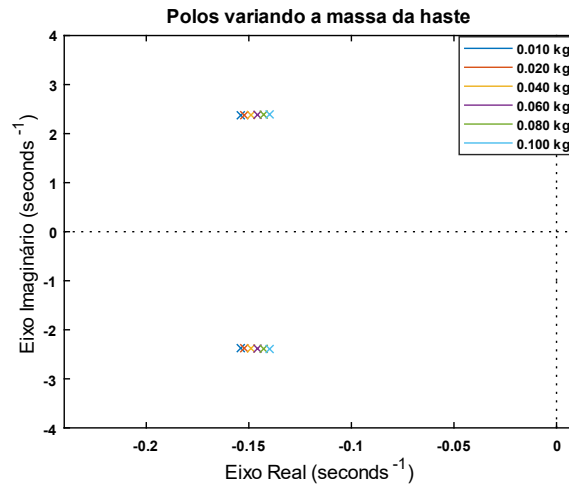
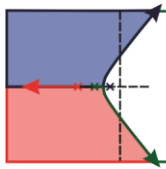


Figura 4 - Polos do sistema variando a massa da haste.

Analisando o comportamento dos polos com a variação dos parâmetros do sistema é possível observar que, ao aumentar a massa do motor, a componente real e imaginária dos polos decresce. Com massas menores as componentes real e imaginária dos polos aumentam. O mesmo ocorre com a massa da haste, como mostrado na Figura 2. Já, os polos variando o ângulo de operação, possuem componentes com parte real invariáveis até o ângulo de 85° e com parte imaginária crescente conforme o aumento do ângulo. Por esta análise, com o ângulo de 90° é possível observar um polo em $s = 0$ e outro polo com componente imaginária nula. Observando os polos resultantes das variações do coeficiente de amortecimento viscoso, nota-se que maiores valores de coeficiente de amortecimento aumentam a componente real dos polos, Figura 3, e os polos variando a massa da haste crescem sua parte real, conforme o aumento de sua massa, Figura 4. As componentes dos polos do sistema Viga-hélice influenciam a resposta do sistema na seguinte forma: polos com maiores componentes reais negativas proporcionam resposta mais rápidas e as componentes imaginárias são correspondentes as frequências de oscilação da resposta.

CONSIDERAÇÕES FINAIS

Considerando os parâmetros nominais de construção do sistema Viga-hélice e os parâmetros escolhidos para a análise dinâmica foi possível calcular em quais configurações o sistema obteve um melhor desempenho. Analisando a posição dos polos no semi-plano esquerdo, o sistema possui resposta menos oscilatória e mais estável com a massa do motor de $0,050 \text{ kg}$, massa da haste de $0,010 \text{ kg}$, utilizando o coeficiente de amortecimento viscoso de 1 Nms/rad e haste medindo $0,20 \text{ m}$. Concluindo que o sistema é mais estável e tem melhor resposta com massas menores, maior coeficiente de amortecimento viscoso, menor comprimento de haste, comparado a faixa de parâmetros escolhida e os parâmetros nominais do sistema Viga-hélice, facilitando assim o projeto de um controlador para o sistema.

REFERÊNCIAS

- JOB, M. M.; JOSE, P. S. H. **Modeling and control of mechatronic aeropendulum**. In: International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015, Coimbatore. Anais... Piscataway: IEEE, 2015. pp. 1-5.
- H. R. M. Silva, R. Cardim, M. C. M. Teixeira, E. Assunção and I. T. M. Ramos. **Switched Control and Tracking Application in Aeropendulum System using Fuzzy Models**. 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2021, pp. 1-6, doi: 10.1109/FUZZ45933.2021.9494543.
- NISE, Norman. **Engenharia de sistemas de controle**: Rio de Janeiro. 7 ed: LTC, 2018.



II SIMECA - IFPR

SIMPÓSIO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO

22 a 25 de novembro de 2021



**INSTITUTO
FEDERAL**
Paraná

Campus
Jacarezinho