

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

GABRIEL ROSSATO FRANCISCO

**SEGUIDOR SOLAR DE DOIS EIXOS COM SENSOR TRONCO
PIRAMIDAL APLICADO A UM SISTEMA DE GERAÇÃO DE
ENERGIA OFF-GRID**

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2022

GABRIEL ROSSATO FRANCISCO

**SEGUIDOR SOLAR DE DOIS EIXOS COM SENSOR TRONCO PIRAMIDAL
APLICADO A UM SISTEMA DE GERAÇÃO DE ENERGIA OFF-GRID**

**Two-axis solar tracker with pyramidal stem sensor applied to an Off-Grid power
generation system**

Dissertação apresentada como requisito para
obtenção do título de Mestre em Engenharia Elétrica
da Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientador: Dr. Alessandro do Nascimento Vargas.

CORNÉLIO PROCÓPIO

2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite o download e o compartilhamento da obra desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-la ou utilizá-la para fins comerciais.



**Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio**



GABRIEL ROSSATO FRANCISCO

SEGUIDOR SOLAR DE DOIS EIXOS COM SENSOR TRONCO PIRAMIDAL APLICADO A UM SISTEMA DE GERAÇÃO DE ENERGIA OFF-GRID

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Sistemas Eletrônicos Industriais.

Data de aprovação: 10 de Junho de 2022

Dr. Alessandro Do Nascimento Vargas, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Marcio Aurelio Furtado Montezuma, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Moacyr Aureliano Gomes De Brito, Doutorado - Universidade Federal de Mato Grosso do Sul (Ufms)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 10/06/2022.

Dedico esse trabalho a toda minha família. Principalmente ao meu pai, que contribuiu significativamente para o desenvolvimento desta universidade onde hoje eu curso meu Mestrado.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Alessandro do Nascimento Vargas e meu mentor o Prof. Marcio Aurelio Furtado Montezuma pela oportunidade de pesquisar e aprender ao lado deles.

Aos meus colegas do laboratório LaSisC e do Lab Control.

Agradeço também a minha família, pilar fundamental de toda minha formação acadêmica.

À CAPES pelo apoio financeiro provido para realização da pesquisa.

RESUMO

FRANCISCO, Gabriel. Seguidor solar de dois eixos com sensor tronco piramidal aplicado a um sistema de geração de energia Off-Grid. 2022. 51 f. Dissertação de Mestrado – Programa De Pós-Graduação Em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Esta dissertação apresenta contribuições para os sistemas seguidores solares, com especial ênfase na geração de energia fotovoltaica. É mostrado um sensor inovador com formato tronco piramidal para a leitura da direção da componente direta da irradiância solar. Sua função é permitir a integração de dois métodos de controle (controle clássico mesclado a uma lógica de histerese) para gerar economia de energia na rotação da estrutura e reduzir seu desgaste mecânico. O objetivo principal do trabalho é desenvolver e implementar um seguidor solar biaxial completo: estrutura mecatrônica, eletrônica embarcada, projeto de controladores e conversão da energia. Emprega-se um conversor Buck, junto a algoritmos para Seguimento do Ponto de Máxima Potência. Apresenta-se uma comparação da produção energética do seguidor solar em relação a um sistema fixo: experimentos indicam que o seguidor proporcionou um incremento próximo a 32% na produção energética.

Palavras-chave: Seguidor solar; Energia fotovoltaica; Eficiência energética; Controle de sistemas; Histerese; Instrumentação.

ABSTRACT

FRANCISCO, Gabriel. Two-axis solar tracker with pyramidal stem sensor applied to an Off-Grid power generation system. 2022. 51 f. Dissertação de Mestrado – Programa De Pós-Graduação Em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

This dissertation presents contributions to solar tracker systems with a special emphasis on photovoltaic energy generation. One innovation is a sensor with a pyramid-shaped trunk for reading the direct component direction of the solar irradiance. Its function is to allow the integration of two control methods (classic control and hysteresis logic) to generate energy savings in the rotation of the structure and reduce mechanical wear. The main goal of the work is to develop and implement a complete biaxial solar tracker that includes a mechatronic structure, embedded electronics, controller design, and energy conversion. A Buck converter and Maximum Power Point Tracking (MPPT) algorithms are also used. There is a comparison of the energy production of the solar tracker with a fixed system. Experiments show that the solar tracker increased energy production by almost 32 %.

Keywords: Solar tracker; Photovoltaic energy; Energy efficiency; Systems control; Hysteresis; Instrumentation.

LISTA DE FIGURAS

Figura 1 – Estrutura biaxial projetada para o seguidor solar	3
Figura 2 – Vista do eixo X da estrutura	3
Figura 3 – Vista do eixo Y da estrutura	4
Figura 4 – Estrutura com os painéis fotovoltaicos montados	4
Figura 5 – Fixação dos atuadores na estrutura biaxial	5
Figura 6 – Sensor para leitura da angulação de máxima incidência de irradiação solar.	6
Figura 7 – Saída e sinal de controle do eixo X.	7
Figura 8 – Saída e sinal de controle do eixo Y.	8
Figura 9 – Fluxograma da operação da lógica por histerese.	9
Figura 10 – Conversor Buck síncrono com painel fotovoltaico na entrada.	11
Figura 11 – Modelo da célula fotovoltaica	13
Figura 12 – Curvas de tensão por corrente dos painéis para diferentes condições	14
Figura 13 – Curvas de tensão por potência dos painéis para diferentes condições	15
Figura 14 – Fluxograma do algoritmo P&O com limitador para carregamento de bateria.	16
Figura 15 – Esquemático Buck síncrono com PV na entrada.	17
Figura 16 – Esquemático leitura da resistência dos LDRs	17
Figura 17 – Esquemático das pontes H com os atuadores	18
Figura 18 – Esquemático dos sensores magnéticos de meio de curso	18
Figura 19 – Placa de circuito impresso final utilizada no sistema.	19
Figura 20 – Lugar das raízes da função de transferência eixo X.	20
Figura 21 – Lugar das raízes da função de transferência eixo Y.	21
Figura 22 – Resposta no eixo X para perturbação.	22
Figura 23 – Resposta no eixo Y para perturbação.	22
Figura 24 – Controle PI atuando na tensão de entrada V_{pv} dos painéis	23
Figura 25 – Método P&O convencional realizando o MPPT no sistema	23
Figura 26 – Controle PI operando na tensão do painéis.	24
Figura 27 – Teste de carga bateria de chumbo-ácido.	24
Figura 28 – Potência produzida pelos painéis da estrutura fixa comparada com a seguidora.	25

LISTA DE TABELAS

Tabela 1 – Coeficientes dos filtros IIR de segunda ordem	10
Tabela 2 – Parâmetros dos componentes do conversor Buck	13
Tabela 3 – Parâmetros dos painéis Komaes Solar KMP50	14
Tabela 4 – Energia produzida pelos painéis fotovoltaicos durante o dia	26

LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog to Digital Converter
CC	Corrente Continua
CA	Corrente Alternada
MIPS	Milhões de Instruções Por Segundo
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MPPT	Maximum Power Point Tracker
LDR	Light Dependent Resistor
P&O	Perturba e Observa
PI	Proporcional Integral
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SRAM	Static Random Access Memory
TIG	Tungsten Inert Gas
PCB	Printed Circuit Board
IIR	Infinite Impulse Response
DsPIC	Digital Signal Processor from Microchip
SW1	Switch 1 Buck Converter
SW2	Switch 2 Buck Converter
ICSP	In Circuit Serial Programmer
STC	Standard Test Conditions
UART	Universal Asynchronous Receiver Transmitter

LISTA DE SÍMBOLOS

V_{oc}	Tensão de circuito aberto do painel fotovoltaico (V)
I_{sc}	Corrente de curto circuito do painel fotovoltaico (A)
V_{mpp}	Tensão de máxima potência em STC do painel fotovoltaico (V)
I_{mpp}	Corrente de máxima potência em STC do painel fotovoltaico (A)
R_s	Resistência série modelo painel (Ω)
R_p	Resistência paralela modelo painel (Ω)
α	Coefficiente de variação da corrente de curto circuito pela temperatura (A/°K)
N_s	Número de células fotovoltaicas em série do painel
V_{cell}	Tensão da célula fotovoltaica (V)
I_{cell}	Corrente da célula fotovoltaica (A)
G	Irradiação solar (W/m^2)
G_r	Irradiação solar de referência (W/m^2)
T	Temperatura do painel (°K)
T_r	Temperatura de referência (°K)
I_{ph}	Fotocorrente do painel (A)
I_r	Corrente de saturação reversa (A)
I_{rr}	Corrente de saturação reversa de referência (A)
η	Fator de idealidade do diodo
q	Carga do elétron (C)
k	Constante de boltzmann ($m^2 \cdot Kg \cdot s^{-2} \cdot K^{-1}$)
E_g	Band gap silício (eV)
K_{malha}	Ganho de realimentação da malha
W_n	Frequência natural do sistema (rad/s)

t_a	Tempo de acomodação (s)
ξ	Fator de amortecimento
$\%Over$	Percentual de Overshoot (%)
Deg	Amplitude do degrau
$MaxPico$	Máximo pico da resposta da saída
$G_X(s)$	Função de transferência do eixo X
$G_Y(s)$	Função de transferência do eixo Y
$x(n)$	Entrado do filtro IIR no instante atual
$y(n)$	Saída do filtro IIR no instante atual
R_{pv}	Resistência que representa o painel para modelagem (Ω)
R_o	Resistência de saída do conversor (Ω)
V_{pv}	Tensão do painel fotovoltaico (V)
I_{pv}	Corrente do painel fotovoltaico (A)
T_S	Período de chaveamento do conversor (s)
f_S	Frequência de chaveamento do conversor (Hz)
d	Razão cíclica do conversor
L	Indutor do conversor (H)
C_{pv}	Capacitor de entrada do conversor (F)
C_o	Capacitor de saída do conversor (F)
R_L	Resistência do indutor (Ω)
V_L	Tensão do Indutor (V)
I_L	Corrente do Indutor (A)
V_o	Tensão da saída do conversor (V)
I_{C_o}	Corrente do capacitor de saída (A)
$I_{C_{pv}}$	Corrente do capacitor de entrada (A)
$G_{vd}(s)$	Função de transferência do conversor

ΔV_o	Ripple de tensão do capacitor saída (V)
ΔI_L	Ripple de corrente do indutor (A)
ΔV_{pv}	Ripple de tensão do capacitor entrada (V)
T_{adc}	Período interno do conversor ADC (s)
f_c	Frequência de corte do filtro (Hz)
Q	Fator de qualidade do filtro
ρ	Posição do polo no eixo real
K_i	Ganho integrativo do controle PI
K_p	Ganho proporcional do controle PI
ΔV_{ref}	Passo de incremento do P&O (V)

SUMÁRIO

1 – INTRODUÇÃO	1
2 – MATERIAIS E MÉTODOS	3
2.1 Construção da estrutura biaxial	3
2.2 Sensor projetado para leitura da angulação de máxima incidência de irradiação solar	5
2.3 Identificação da função de transferência de cada eixo do seguidor solar	6
2.4 Lógica por histerese que comuta o controlador clássico	8
2.5 Modelagem do conversor Buck acoplado ao painel fotovoltaico	10
2.6 Determinação da indutância e das capacitâncias do conversor Buck síncrono	12
2.7 Modelo da célula fotovoltaica	13
2.8 Curvas dos painéis fotovoltaico baseadas no modelo	14
2.9 Algoritmo para rastreamento do ponto de máxima potência dos painéis fotovoltaicos, com limitador para carregamento de bateria	15
2.10 Instrumentação eletrônica, microcontrolador e circuitos de potência empregados na confecção da placa de circuito impresso.	16
3 – RESULTADOS E DISCUSSÕES	20
3.1 Projeto do controlador proporcional para o seguidor solar	20
3.2 Experimento: controladores atuando no seguidor solar	21
3.3 Projeto e teste do controlador PI do conversor Buck síncrono	21
3.4 Experimentos: Métodos de MPPT em operação e sistema Off-Grid carregando uma bateria	23
3.5 Experimento para a potência produzida: comparação entre o seguidor solar e estrutura fixa	25
4 – CONCLUSÃO	27
Referências	28
Apêndices	31
APÊNDICE A – Código em C++ implementado no dsPIC33EV256GM104 compilador XC16 V1.5	32

APÊNDICE B–Algoritmo implementado no MATLAB para comunicação com dsPIC	50
--	----

1 INTRODUÇÃO

Considerando que a energia solar fotovoltaica representa somente 1,22% da matriz energética brasileira, segundo publicação governamental recente (BEN, 2020), observa-se a existência de um amplo potencial para sua expansão. Nesse sentido, pesquisadores tem buscado maneiras de aprimorar a eficiência dos sistemas fotovoltaicos. Por exemplo, a literatura sugere três principais formas para atingir tal objetivo: A primeira é aperfeiçoando os módulos fotovoltaicos, por meio da utilização de novas tecnologias, ou aprimorando os conversores estáticos empregados na conversão da potência produzida. A segunda consiste em aprimorar os métodos de *Maximum Power Point Tracking* (MPPT), para proporcionar uma extração mais eficiente da potência dos módulos, exemplos são apresentados pelos autores (Oliveira et al., 2015; de Brito et al., 2013). A terceira consiste em melhorar a eficiência da captação da luz solar nos módulos, com uso dos sistemas seguidores solar (AL-ROUSAN; ISA; DESA, 2018). Nesse contexto, este trabalho apresenta uma contribuição para o aprimoramento da eficiência dos sistemas fotovoltaicos via seguidor solar, conforme discutido adiante.

Um seguidor solar é um dispositivo mecatrônico que modifica continuamente a posição dos painéis fotovoltaicos, de modo a mantê-los perpendicular em relação a componente direta da irradiância solar, semelhante ao efeito girassol (e.g., (PRINSLOO; DOBSON, 2015)). Note que a posição perpendicular proporciona a máxima produção energética possível, considerando um mesmo arranjo de painéis sob as mesmas condições climáticas (Lee; Rahim, 2013).

A estratégia de mover os painéis, automaticamente, de modo que sigam a luz do sol não é nova. As primeiras pesquisas relacionadas com os sistemas seguidores solar, surgiram nas décadas de 80 e 90, publicadas pelos autores (Pritchard, 1983; Koyuncu; Balasubramanian, 1991). Cabe ressaltar que esses trabalhos foram desenvolvidos com hardware limitado, disponível naquela época, sendo que não se conseguia extrair a máxima potência dos painéis fotovoltaicos de forma consistente. Por volta do ano 2000, com aprimoramento dos conversores estáticos de potência, foi possível aplicar os métodos conhecidos como MPPT nos sistemas seguidores solar, constatando-se então ganhos expressivos na produção energética (Huang; Tien; Or, 1998; Luque-Heredia et al., 2003).

A literatura sugere dois tipos de algoritmos para implementar o mecanismo automático dos seguidores solar. O primeiro é do tipo astronômico, ou seja, o algoritmo utiliza dados da localização geográfica dos painéis para determinar o ângulo de azimute e altitude do sol, e assim determina os movimentos para onde será rotacionado os painéis (e.g., (Zhan et al., 2013)). O segundo tipo de algoritmo opera rastreando a angulação de máxima incidência de irradiação solar em tempo real, com auxílio de sensores de luminosidade montados juntos aos painéis (e.g., (Mustafa et al., 2018)). O segundo tipo de algoritmo promove uma maior eficiência energética, quando comparado com o primeiro algoritmo (o astronômico), em situações nas quais o mecanismo seguidor solar é incorretamente alinhado com os polos magnéticos. A vantagem do

segundo algoritmo se deve ao fato deste operar em malha fechada, proporcionando menores erros de regime permanente na orientação dos painéis (KABALCI, 2020).

Os seguidores solares mais simples são construídos com um único eixo de rotação. Normalmente tais dispositivos são aplicados em países de baixas latitudes, como é o caso do Brasil. Este único eixo frequentemente é alinhado ao longo do Meridiano Norte verdadeiro (BAHRAMI; OKOYE; ATIKOL, 2016; Portolan dos Santos; RÜTHER, 2014). Porém, dotar o sistema com dois eixos de rotação tem as seguintes vantagens: (1) mobilidade e (2) ganho em eficiência. O sistema seguidor solar mostrado neste trabalho contém dois eixos, ou seja, é biaxial e opera em tempo real rastreando a angulação de máxima irradiação solar. Neste trabalho, a mobilidade dos painéis possibilita que sejam transportados para locais remotos e colocado em funcionamento sem a necessidade do usuário se preocupar com o correto alinhamento com os polos magnéticos.

Em relação ao ganho de eficiência proporcionado pelos seguidores solares com dois eixos de rotação, tais sistemas começam a ganhar uma maior atenção, amparado pelos artigos (Yazidi et al., 2006; Beltran A.; Gonzalez Rubio S.; Garcia-Beltran, 2007) onde os autores obtêm incrementos na produção energética da ordem de 26%. Por exemplo, em (Sawant et al., 2018), o autor realiza uma comparação de um sistema de dois eixos com outro de um único eixo, e obtêm com o sistema biaxial um incremento de eficiência próximo a 25%. Este trabalho segue na linha desses autores, conforme será detalhado a seguir.

O objetivo deste trabalho é mostrar o desenvolvimento e implementação de um sistema seguidor solar biaxial através do projeto e confecção de uma estrutura mecatrônica. O seguidor solar exige uma instrumentação cuidadosa, projeto dos controladores e processamento embarcado em um microcontrolador. Serão apresentados dados experimentais da produção energética do seguidor solar e comparados aos dados coletados em um sistema fixo (sem seguidor solar). Para extrair a potência dos painéis fotovoltaicos serão empregados um conversor estático de potência, juntamente com um sistema de controle usando MPPT. Por fim, será conectado ao sistema uma bateria para armazenar a energia produzida pelos painéis, constituindo-se um sistema do tipo *Off-Grid* (e.g., (ALDOSARY et al., 2021)).

Este trabalho também apresenta uma contribuição do ponto de vista de sensores. Tendo como base o sensor sugerido por (Mustafa et al., 2018), que possui um formato de pirâmide de base quadrada com 4 faces, mostra-se um sensor inovador com formato de tronco de pirâmide de base quadrada, que contém 5 faces. Tal sensor de 5 faces é utilizado neste trabalho para o rastreamento da angulação de máxima incidência de irradiação solar. Sua função é permitir a integração de dois métodos de controle distintos, gerando benefícios ao sistema, conforme descritos no decorrer dessa dissertação.

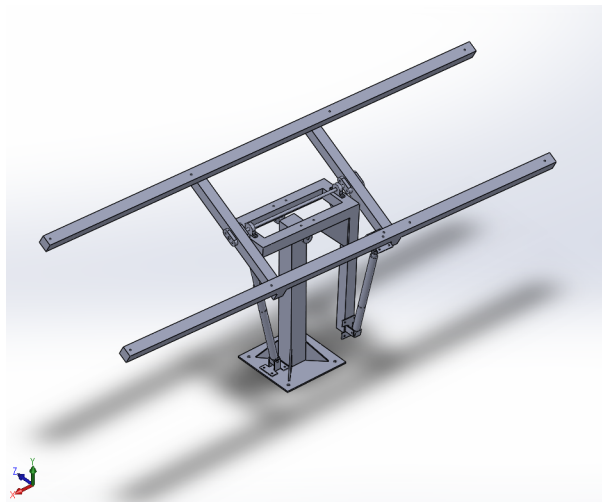
2 MATERIAIS E MÉTODOS

Esta seção descreve os materiais utilizados na construção da plataforma, a estratégia de controle empregada, assim como os métodos de controle desenvolvidos.

2.1 Construção da estrutura biaxial

Utilizando o software de projeto tridimensional SolidWorks desenvolveu-se uma estrutura biaxial, capaz de rotacionar dois painéis fotovoltaicos em até 45° para ambos os lados e sob ambos os eixos, em relação ao solo. Essa angulação limite de 45° ocorre devido as limitações dos componentes disponíveis para confecção da estrutura. Uma visão geral do projeto é mostrada na Figura 1. Já as vistas laterais dos dois eixos rotacionados até seus ângulos limites, são ilustradas nas Figuras 2 e 3.

Figura 1 – Estrutura biaxial projetada para o seguidor solar



Fonte: Autoria própria

Figura 2 – Vista do eixo X da estrutura

(a) Rotacionado -45°



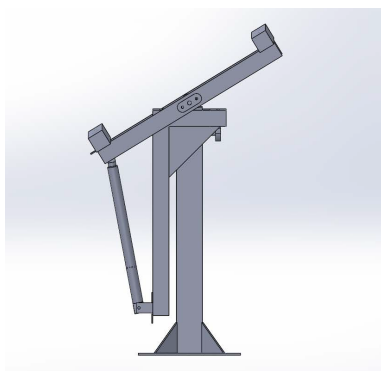
Fonte: Autoria própria

(b) Rotacionado $+45^\circ$

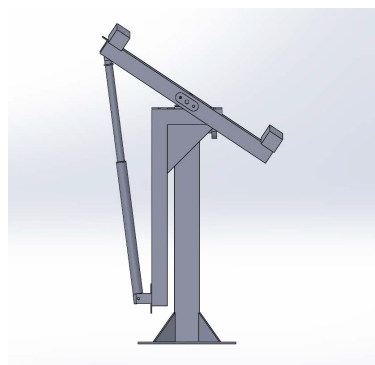


Fonte: Autoria própria

Figura 3 – Vista do eixo Y da estrutura

(a) Rotacionado -45° 

Fonte: Autoria própria

(b) Rotacionado $+45^\circ$ 

Fonte: Autoria própria

A estrutura foi confeccionada com tubos de alumínio unidos por meio do processo de soldagem *Tungsten Inert Gas* (TIG), esta comporta dois painéis da KomaesSolar de 50 watts cada, fixados lado-a-lado no seu topo. Toda fabricação: processos de medição, corte e soldagem foram feitos internamente na intuição, na oficina de engenharia mecânica, conforme mostrado na Figura 4.

A estrutura possui dois sensores do tipo botoeira magnética, um em cada eixo, ambos são acionados quando os painéis estão posicionados a exatos 20° no sentido Norte. O intuito é simular uma orientação fixa da estrutura em uma angulação ótima de apontamento (Portolan dos Santos; RüTHER, 2014). Tal estratégia será utilizada nos resultados para realizar uma comparação com o sistema do tipo seguidor solar.

Figura 4 – Estrutura com os painéis fotovoltaicos montados



Fonte: Autoria própria

Para alteração da angulação dos eixos da estrutura empregaram-se dois atuadores elétricos lineares pivotantes, compostos por motores de corrente contínua (CC) de imã permanente, que trabalham com uma tensão nominal de 12 V, possuem um curso de 200 mm, uma velocidade de deslocamento sem carga igual a 10 mm/s, e são capazes de aplicar forças de até 500 N. Cada atuador apresenta uma potência nominal de 24 W, porém para rotação dos painéis eles consumiram uma potência máxima de 2 W. Uma fonte externa foi utilizada para alimentar esses atuadores. A fixação dos atuadores na estrutura é mostrada em detalhes na Figura 5.

Figura 5 – Fixação dos atuadores na estrutura biaxial



Fonte: Autoria própria

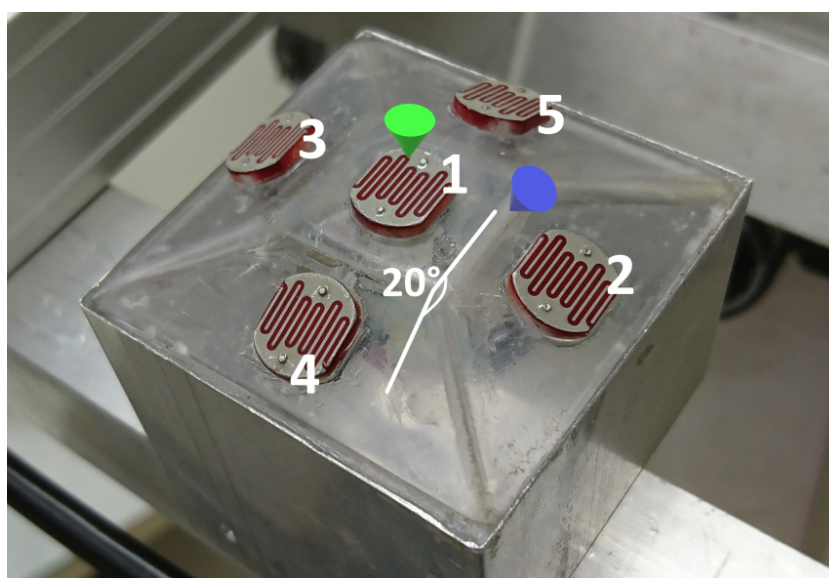
2.2 Sensor projetado para leitura da angulação de máxima incidência de irradiação solar

Em seu artigo (Mustafa et al., 2018), os autores utilizam um sensor com formato de pirâmide de base quadrada, contendo quatro faces, no qual cada face contém um *Light Dependent Resistor* (LDR). Os autores aplicam um controle clássico por eixo, implementado por meio de circuitos analógicos, que tem como entrada a diferença de luminosidade entre os LDRs dispostos em cada eixo. O formato do sensor sugerido por (Mustafa et al., 2018) foi adaptado para este trabalho, da seguinte forma: acrescentou-se uma nova face no seu topo para acoplar um LDR adicional, consequentemente cada uma das 5 faces contém um único LDR. O formato do novo sensor é de um tronco de pirâmide de base quadrada, sendo o ângulo entre o topo e face lateral igual a 20° (A escolha dessa angulação será detalhada na subseção 2.4), o sensor já confeccionado é apresentado na Figura 6. O projeto do sensor com 5 faces possibilita um aprimoramento do controle empregado por (Mustafa et al., 2018).

O algoritmo associado ao sensor de 5 faces funciona da seguinte maneira. Não é de interesse que o erro angular lido pelo sensor seja afetado por variações na intensidade da

irradiação solar, que ocorrem ao longo do dia. O objetivo aqui é somente realizar a leitura da direção do vetor de máxima incidência solar. Por exemplo, se no eixo X temos 800 W/m^2 incidindo a 8° , a saída do sensor deve apresentar a mesma leitura sob a situação em que temos 100 W/m^2 incidindo aos mesmos 8° . A informação relevante aqui é a direção, portanto, a diferença de intensidade luminosa entre os LDRs 3 e 2 dividida pela média dos LDRs de 1 a 5 representa o erro angular no eixo X. Já a diferença entre os LDRs 4 e 5, também normalizada pela média de todos os LDRs, representa o erro no eixo Y. Sob esta estratégia, o sistema de controle interpreta em qual direção o mecanismo deve se mover, ou seja, na direção do eixo X ou Y.

Figura 6 – Sensor para leitura da angulação de máxima incidência de irradiação solar.



Fonte: Autoria própria

2.3 Identificação da função de transferência de cada eixo do seguidor solar

O sensor de incidência solar mostrado na Figura 6 foi fixado junto aos painéis solares, no topo da estrutura, em sua parte central. Assumiu-se que o sistema mecatrônico que movimenta os painéis obedece a um comportamento linear e que ambos os eixos são independentes. Um atuador composto por um motor de corrente contínua (CC) é responsável por gerar o movimento em cada um dos dois eixos do sistema seguidor solar.

A dinâmica de um motor CC geralmente é representada por uma equação de segunda ordem. Porém como as dimensões dos motores dos atuadores utilizados são pequenas, é possível representá-los por uma equação de primeira ordem somente, sendo a sua entrada a tensão e a saída a velocidade (CORPORATION, 1972). Deseja-se que a saída da Função de Transferência (FT) seja a posição do atuador, por isso é necessário integrar a velocidade para se obter a

posição. A FT que descreve cada um dos dois eixos é mostrada na Equação 1.

$$G(s) = \frac{B}{s \cdot (s + A)} \quad (1)$$

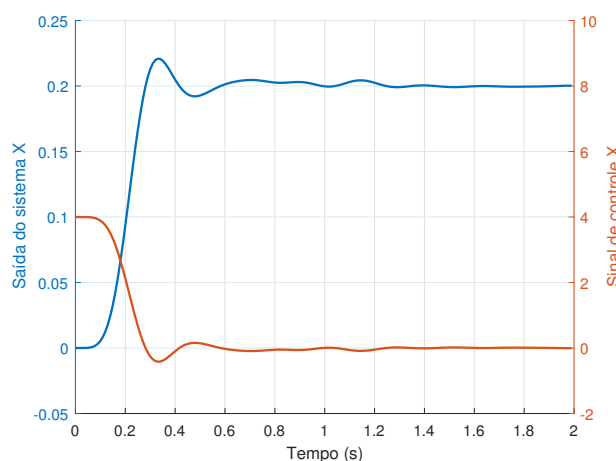
É possível identificar os parâmetros da Equação 1 realizando o seguinte procedimento: multiplicando o sistema $G(s)$ por um ganho proporcional K_p e fechando sua malha, tal ganho deve forçar uma resposta subamortecida no sistema. Aplicando um degrau com amplitude Deg se obtém do sistema o seu pico máximo $MaxPico$ e o tempo de acomodação t_a . O fator de amortecimento ξ é facilmente obtido pelo percentual de overshoot $\%Over = \frac{MaxPico \cdot 100}{Deg} - 100$ e a frequência natural do sistema é obtida pela Equação $W_n = \frac{4}{\xi \cdot t_a}$ (OGATA, 2009).

Aplicando ξ e W_n na função de transferência de malha fechada, mostrada na Equação 2, ao se desfazer a malha que havia sido fechada inicialmente, obtém-se a $G(s)$, onde $A = 2 \cdot \xi \cdot W_n$ e $B = \frac{W_n^2}{K_p}$.

$$G(s)_{MalhaFechada} = \frac{W_n^2}{s^2 + 2 \cdot \xi \cdot W_n \cdot s + W_n^2} \quad (2)$$

Utilizando como ponto de partida o sistema alinhado com a angulação de máxima irradiação solar, aplicou-se individualmente nos eixos X e Y um degrau de amplitude Deg igual a 0,2 no sistema em malha fechada, considerando $K_p = 20$. As saídas e sinais de controle dos eixos X e Y são mostradas respectivamente nas Figuras 7 e 8. Note que o eixo da saída dos sistemas X e Y (Azul) dos gráficos são adimensionais, devido ao processo de normalização descrito na subseção 2.2.

Figura 7 – Saída e sinal de controle do eixo X.

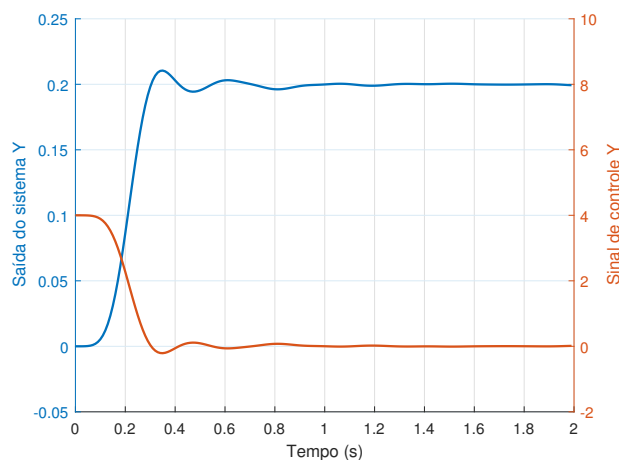


Fonte: Autoria própria

Para o eixo X obteve-se um $MaxPico = 0,221$ e $t_a = 1,2$ s, já para o eixo Y obteve-se um $MaxPico = 0,210$ e $t_a = 0,9$ s. Resultando nas funções de transferência mostradas respectivamente nas Equações 3 e 4.

$$G_X(s) = \frac{1,635}{s \cdot (s + 6,667)} \quad (3)$$

Figura 8 – Saída e sinal de controle do eixo Y.



Fonte: Autoria própria

$$G_Y(s) = \frac{2,074}{s \cdot (s + 8,889)} \quad (4)$$

2.4 Lógica por histerese que comuta o controlador clássico

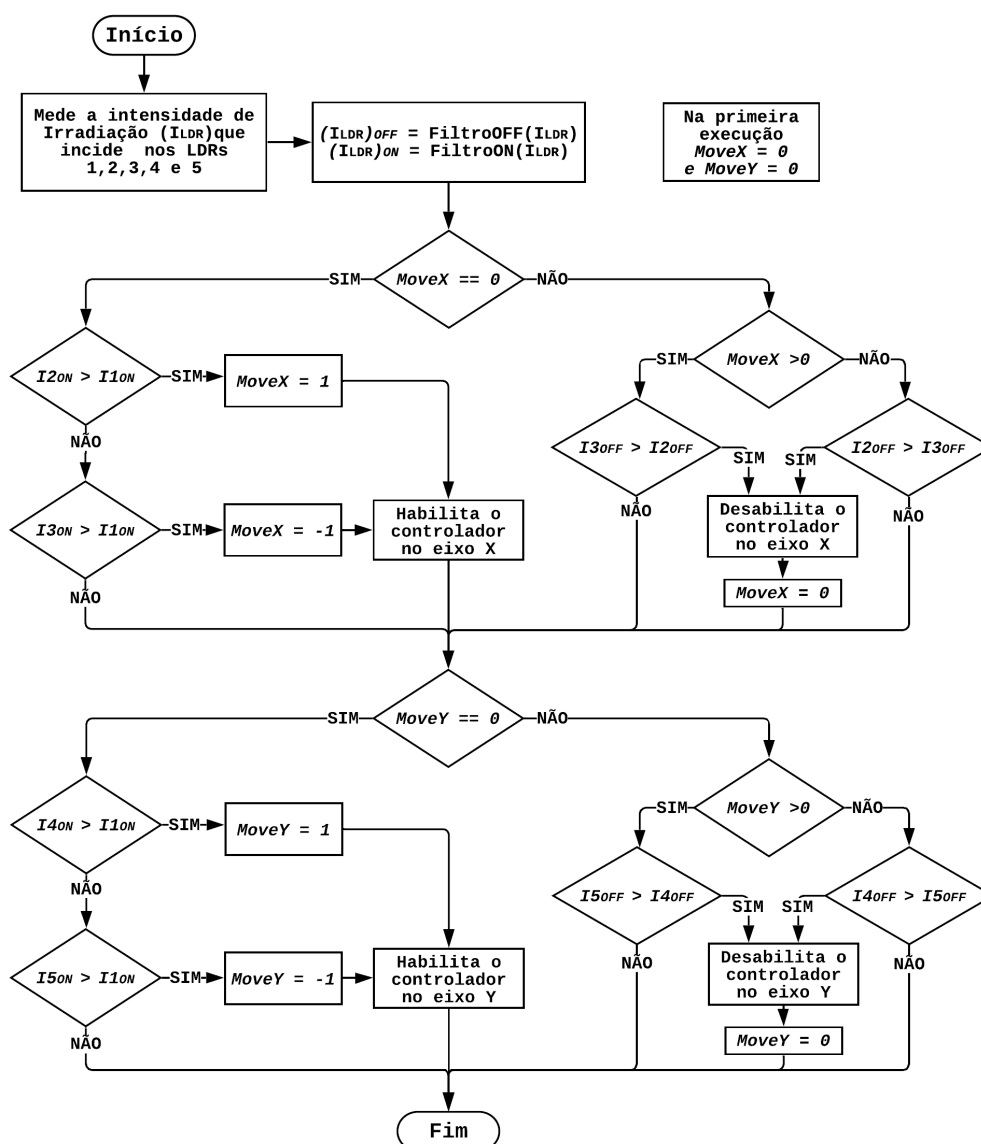
Empregou-se uma lógica por histerese, com intuito de se reduzir o consumo de energia na movimentação dos painéis solares no decorrer do dia, e também diminuir o desgaste mecânico da estrutura. A ideia é associar a lógica por histerese ao controlador clássico proporcional (P). A lógica por histerese tem a tarefa de habilitar e desabilitar o controlador P, conforme discutido a seguir. A literatura demonstra que folgas nos atuadores, nas engrenagens e nos mancais podem gerar sinais de pequena amplitude e alta frequência, quando o mecanismo correspondente é operado por controle linear (e.g., (Ghaffari; Mohammadiasl, 2013; Kumar et al., 2009)). A lógica de histerese associada ao controle linear tem como objetivo suprimir tais oscilações.

Adotou-se nesse trabalho uma histerese de 10° , que foi definido como a metade do ângulo entre o topo e a face lateral do sensor mostrado na Figura 6. Isso significa que, ao ligar o equipamento, o mecanismo pode realizar no máximo 9 acionamentos liga-desliga ao longo do dia, porque a estrutura tem leque de rotação igual a 90° (45° para cada lado). A perda de potência teórica para essa angulação é de no máximo 1,5%. Este valor foi calculado por meio da decomposição vetorial da irradiação solar quando incide a 10° nos painéis, sendo sua resultante igual a $\cos(10^\circ) = 0,985$.

Para exemplificar o funcionamento da lógica por histerese, vamos considerar a seguinte situação: o sistema é iniciado e apresenta um erro de 6° no eixo X no sentido horário. No decorrer do dia este erro vai crescendo até atingir 10° , situação na qual a luminosidade lida pelo LDR 2 é maior que a do LDR 1 (ilustrada pela seta azul na Figura 6). Habilita-se então o controlador do eixo X, que movimenta a estrutura no sentido horário, até que o erro do eixo X atinja um valor próximo a zero.

Quando o erro atinge um valor ligeiramente negativo (próximo a zero), a luminosidade lida pelo LDR 3 será maior que a do LDR 2 (Situação ilustrada pela seta verde na Figura 6). Desabilita-se então o controlador do eixo X, cessando toda a movimentação desse eixo. Note que este eixo será habilitado novamente somente quando existir um erro superior a 10°. Este exemplo ilustra a atuação da lógica por histerese no eixo X e em um único sentido. Situação análoga ocorre no sentido anti-horário e também no eixo Y, em ambos os sentidos de rotação, operando de acordo com o diagrama mostrado na Figura 9.

Figura 9 – Fluxograma da operação da lógica por histerese.



Fonte: Autoria própria

Com a finalidade de se evitar acionamentos e interrupções não desejadas no funcionamento da lógica de histerese, implementou-se filtros digitais IIR Chebyshev do tipo 2 de segunda ordem nas medições da irradiação feita nos LDRs. Nos comparadores que habilitam o controlador utilizou-se uma frequência de corte de 0,1 Hz (Denominado de FiltroON), já nos comparadores que desabilitam o controlador utilizou-se uma frequência de corte de 10 Hz

(Denominado de FiltroOFF). Ambos foram projetados seguindo a metodologia mostrada em (OPPENHEIM, 2013), eles trabalham a um frequência amostral de 2 kHz e seus coeficientes são mostrados na Tabela 1, é mostrada também sua implementação na Equação 5, sendo $x(n)$ a entrada e $y(n)$ a saída deles.

$$a_1 \cdot y(n) = b_1 \cdot x(n) + b_2 \cdot x(n - 1) + b_3 \cdot x(n - 2) - a_2 \cdot y(n - 1) - a_3 \cdot y(n - 2) \quad (5)$$

Tabela 1 – Coeficientes dos filtros IIR de segunda ordem

FiltroON	FiltroOFF
$a=[1,00000000 -1,99937482 0,99937502]$	$a=[1,00000000 -1,93697498 0,93891957]$
$b=[0,00999692 -0,01999365 0,00999692]$	$b=[0,01017588 -0,01840718 0,01017588]$

2.5 Modelagem do conversor Buck acoplado ao painel fotovoltaico

Dentre as diversas topologias de conversores CC-CC não isolados existentes na literatura, optou-se por utilizar um conversor Buck devido a necessidade de se reduzir a tensão do arranjo de painéis para a carga, ele tem sido bastante empregado em sistemas do tipo *Off-Grid* (BARBI; MARTINS, 2006). Tal conversor apresenta também uma construção simplificada e demanda um projeto de controle mais trivial, pois seu modelo não tem zeros no semiplano direito, assim ele possui fase mínima (ERICKSON; MAKSIMOVIC, 2001).

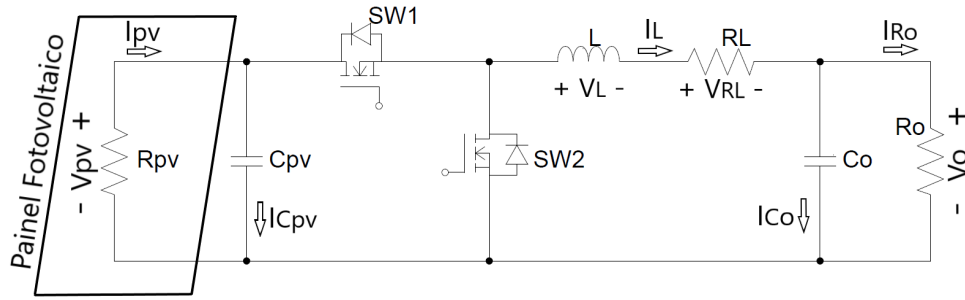
Empregou-se um conversor Buck do tipo síncrono, porque ele tende a possuir uma melhor eficiência (desde que escolhido um MOSFET apropriado para subsistir o diodo), ele tem como característica sempre operar no modo condução contínua (MCC), visto que não possui um diodo que depende da corrente fluindo no indutor para se polarizar. Considere o conversor mostrado na Figura 10, na sua entrada foi conectado os painéis fotovoltaicos, representados pela resistência R_{pv} , conforme sugerido por (Femia et al., 2009). O arranjo de painéis pode ser interpretado como uma fonte de potência, fornecendo uma tensão V_{pv} e uma corrente I_{pv} para o conversor. Na saída do conversor foi conectada uma resistência R_o . A razão cíclica d representa a parcela do período de chaveamento T_S em que $SW1$ está em condução.

Realizando o equacionamento em MCC da tensão do indutor e das correntes dos dois capacitores, para cada uma das duas etapas e posteriormente ponderando seus valores pela parcela do período de chaveamento de cada uma das etapas, obtemos as medias das etapas ($\langle X \rangle$) nas Equações 6, 7 e 8.

$$\langle V_L \rangle = L \cdot \frac{dI_L}{dt} = V_{pv} \cdot d - R_L \cdot I_L - V_o \quad (6)$$

$$\langle I_{C_{pv}} \rangle = C_{pv} \cdot \frac{dV_{pv}}{dt} = -\frac{V_{pv}}{R_{pv}} - I_L \cdot d \quad (7)$$

Figura 10 – Conversor Buck síncrono com painel fotovoltaico na entrada.



Fonte: Autoria própria

$$\langle I_{Co} \rangle = C_o \cdot \frac{dV_o}{dt} = I_L - \frac{V_o}{R_o} \quad (8)$$

A partir das Equações de 6 a 8 é possível realizar a modelagem CA para pequenos sinais, onde as variáveis são substituídas por seu valor médio (\bar{x}) somado a uma parcela CA (\hat{x}). Os termos resultantes CC (Ex: $\bar{x} \cdot \bar{y}$) que representam a resposta em regime permanente, se anulam e assumimos que os termos CA de 2ª ordem (Ex: $\hat{x} \cdot \hat{y}$) são pequenos, quando comparados aos de 1ª ordem (Ex: $\bar{x} \cdot \hat{y}$), a ponto de poderem ser desprezados (ERICKSON; MAKSIMOVIC, 2001). Esta estratégia resulta nas equações dos estados do sistema 9, 10 e 11.

$$L \cdot \frac{d(\hat{I}_L)}{dt} = \bar{V}_{pv} \cdot \hat{d} + \hat{V}_{pv} \cdot \bar{d} - R_L \cdot \hat{I}_L - \hat{V}_o \quad (9)$$

$$C_{pv} \cdot \frac{d(\hat{V}_{pv})}{dt} = -\frac{\hat{V}_{pv}}{R_{pv}} - \bar{I}_L \cdot \hat{d} - \hat{I}_L \cdot \bar{d} \quad (10)$$

$$C_o \cdot \frac{d(\hat{V}_o)}{dt} = \hat{I}_L - \frac{\hat{V}_o}{R_o} \quad (11)$$

Aplicando a transformada de Laplace nas equações de 9 a 11 e manipulando seus termos obtém-se uma função de transferência que tem como saída \hat{V}_{pv} e entrada \hat{d} , mostrada na Equação 12.

$$\begin{aligned} N_2 &= \frac{\bar{I}_L}{C_{pv}} & D_2 &= \frac{R_L}{L} + \frac{1}{R_{pv} \cdot C_{pv}} + \frac{1}{R_o \cdot C_o} \\ N_1 &= \frac{R_L \cdot \bar{I}_L + \bar{d} \cdot \bar{V}_{pv}}{L \cdot C_{pv}} + \frac{\bar{I}_L}{R_o \cdot C_o \cdot C_{pv}} & D_1 &= \frac{\bar{d}^2}{C_{pv} \cdot L} + \frac{R_L}{R_{pv} \cdot C_{pv} \cdot L} + \frac{R_L}{R_o \cdot C_o \cdot L} + \frac{1}{C_o \cdot L} + \frac{1}{R_o \cdot C_o \cdot R_{pv} \cdot C_{pv}} \\ N_0 &= \frac{\bar{I}_L}{C_o \cdot L \cdot C_{pv}} + \frac{R_L \cdot \bar{I}_L + \bar{d} \cdot \bar{V}_{pv}}{R_o \cdot C_o \cdot L \cdot C_{pv}} & D_0 &= \frac{R_{pv} \cdot \bar{d}^2 + R_L + R_o}{R_o \cdot C_o \cdot L \cdot R_{pv} \cdot C_{pv}} \end{aligned}$$

$$G_{vd}(s) = \frac{\hat{V}_{pv}}{\hat{d}} = -\frac{N_2 \cdot s^2 + N_1 \cdot s + N_0}{s^3 + D_2 \cdot s^2 + D_1 \cdot s + D_0} \quad (12)$$

Considerando um conversor Buck Ideal é possível correlacionarmos os valores médios \bar{I}_L , \bar{d} e a R_{pv} com as variáveis medias \bar{V}_{pv} , \bar{I}_{pv} , \bar{V}_o e a R_o pela igualdade das potencias de

entrada e de saída, pelo ganho estático do conversor e pela resistência do painel fotovoltaico, resultando nas Equações de 13 a 16.

$$\overline{V_o} = \sqrt{R_o \cdot \overline{V_{pv}} \cdot \overline{I_{pv}}} \quad (13)$$

$$\overline{I_L} = \frac{\overline{V_{pv}} \cdot \overline{I_{pv}}}{\overline{V_o}} \quad (14)$$

$$\overline{d} = \frac{\overline{V_o}}{\overline{V_{pv}}} \quad (15)$$

$$R_{pv} = \frac{\overline{V_{pv}}}{\overline{I_{pv}}} \quad (16)$$

2.6 Determinação da indutância e das capacitâncias do conversor Buck síncrono

Considerando o conversor Buck operando no modo condução contínua com uma carga R_o , vai existir uma ondulação de tensão no capacitor de saída ΔV_o e uma outra de corrente no indutor ΔI_L . Essas ondulações, também conhecidas como *ripple*, são oriundas do processo de chaveamento. O controlador não possui nenhuma ação sobre tais ondulações. Seus valores máximos são determinados no dimensionamento do conversor. São definidos os valores da indutância L , e da capacitância C_o , conforme as Equações 17 e 18. A razão cíclica $D = 0,5$ é onde ocorre os maiores *ripple* de corrente e tensão, considera-se ela para o cálculo desses componentes (BARBI; MARTINS, 2006).

$$L = \frac{\overline{V_{pv_{max}}} \cdot D \cdot (1 - D)}{\Delta I_L \cdot f_s} \quad (17)$$

$$C_o = \frac{\overline{V_{pv_{max}}} \cdot D \cdot (1 - D)}{8 \cdot L \cdot \Delta V_o \cdot f_s^2} \quad (18)$$

A capacitância acoplada ao painel C_{pv} é calculada por meio da corrente que o painel fotovoltaico fornece ao capacitor quando o $SW1$ está aberto, na segunda etapa de operação do conversor Buck, dada pela Equação $I_{pv} = C_{pv} \cdot \frac{dV_{pv}}{dt} \approx C_{pv} \cdot \frac{\Delta V_{pv}}{(1-D) \cdot T_s}$. Resultando na Equação 19. Observe que nesta equação a razão cíclica próxima a $D = 0$ gera o maior *ripple* na tensão do painel fotovoltaico.

$$C_{pv} = \frac{\overline{I_{pv_{max}}} \cdot (1 - D)}{\Delta V_{pv} \cdot f_s} \quad (19)$$

Optou-se por operar o conversor a uma frequência de chaveamento $f_s = 100$ kHz. Deseja-se que ele tenha um *ripple* na tensão de saída máxima igual a $\Delta V_o = 3,45$ mV e um *ripple* na corrente do indutor máximo de $\Delta I_L = 275$ mA, tem-se que a $\overline{V_{pv_{max}}} = 2 \cdot V_{oc} = 43,12$ V, considerando $D = 0,5$ e aplicando as Equações 17 e 18, obtém-se os valores dos componentes do conversor Buck. Na tensão do painel fotovoltaico deseja-se um *ripple* máximo

igual a $\Delta V_{pv} = 65$ mV, tem-se que a $\overline{I_{pv_{max}}} = I_{sc} = 3,04$ A, considerando $D = 0$ e aplicando a Equação 19, obtém-se a capacitância acoplada ao painel. Os resultados são mostrados na Tabela 2.

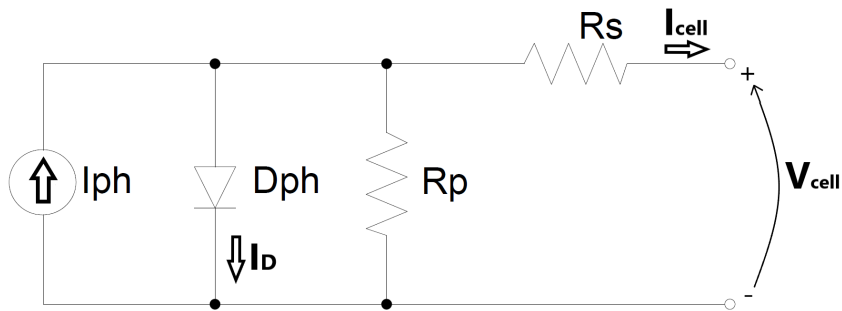
Tabela 2 – Parâmetros dos componentes do conversor Buck

L	390 μ H
C_o	100 μ F
C_{pv}	470 μ F

2.7 Modelo da célula fotovoltaica

Utilizando o modelo mostrado na Figura 11, que foi proposto por (Villalva; Gazoli; Filho, 2009), com um diodo, uma resistência série e uma outro paralela, é possível representar a célula fotovoltaica. Pelas Equações de 20 a 23 é calculada a corrente da célula fotovoltaica para diferentes condições de irradiância e temperatura.

Figura 11 – Modelo da célula fotovoltaica



Fonte: Autoria própria

$$I_{cell} = I_{ph} - I_r \cdot \left(e^{\frac{q \cdot (V_{cell} + R_s \cdot I_{cell})}{\eta \cdot k \cdot T}} - 1 \right) - \frac{V_{cell} + R_s \cdot I_{cell}}{R_p} \quad (20)$$

$$I_{ph} = [I_{sc} + \alpha \cdot (T - T_r)] \cdot \frac{G}{G_r} \quad (21)$$

$$I_r = I_{rr} \cdot \left(\frac{T}{T_r} \right)^3 \cdot e^{\left[\frac{q \cdot E_g}{\eta \cdot k} \cdot \left(\frac{1}{T_r} - \frac{1}{T} \right) \right]} \quad (22)$$

$$I_{rr} = \frac{I_{sc} \cdot R_p - V_{oc}}{R_p \cdot \left(e^{\frac{q \cdot V_{oc}}{\eta \cdot k \cdot T_r}} - 1 \right)} \quad (23)$$

2.8 Curvas dos painéis fotovoltaico baseadas no modelo

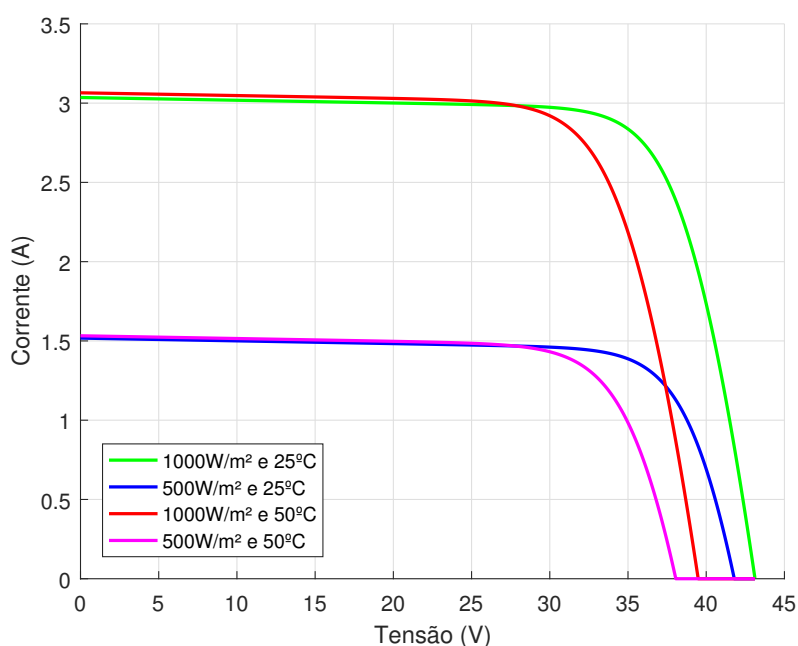
Os dois painéis da KomaesSolar de 50 W foram associados em série. Seus parâmetros são mostrados na Tabela 3, a resistência série e paralela do modelo da célula fotovoltaica foi calculada com base no modelo mostrado pela autor Sera (Sera; Teodorescu; Rodriguez, 2007).

Tabela 3 – Parâmetros dos painéis Komaes Solar KMP50

V_{oc}	21,56 V
I_{sc}	3,04 A
V_{mpp}	17,74 V
I_{mpp}	2,84 A
R_s	12 mΩ
R_p	8 Ω
η	1,0
α	1,1856e-3 A/°C
N_s	36 células

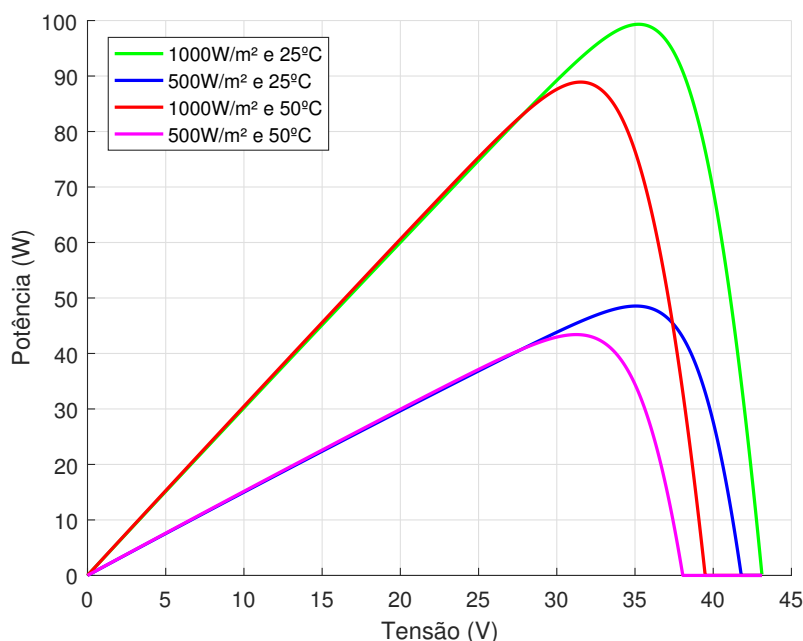
Implementou-se o método de Newton Raphson para Equação 20, em conjunto com as Equações de 21 a 23 para diferentes condições de irradiação solar e temperatura nos painéis fotovoltaicos, a $V_{pv} = 2 \cdot N_s \cdot V_{cell}$ e a $I_{pv} = I_{cell}$. As curvas do arranjo de painéis são mostradas nas Figuras 12 e 13, considerando-se as mesmas condições de irradiância e temperatura nos módulos.

Figura 12 – Curvas de tensão por corrente dos painéis para diferentes condições



Fonte: Autoria própria

Figura 13 – Curvas de tensão por potência dos painéis para diferentes condições



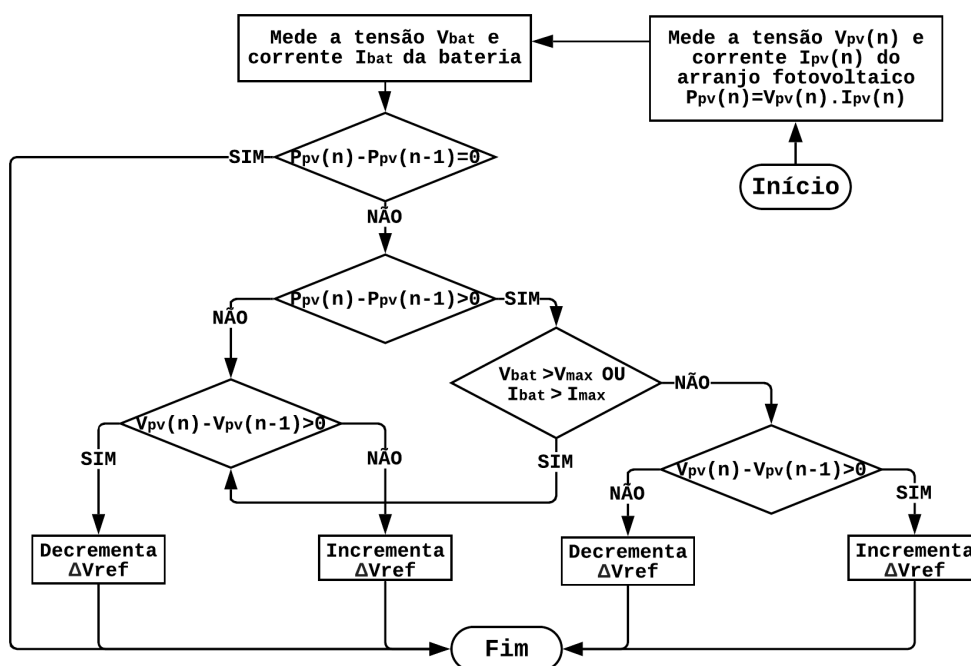
Fonte: Autoria própria

2.9 Algoritmo para rastreamento do ponto de máxima potência dos painéis fotovoltaicos, com limitador para carregamento de bateria

O algoritmo utilizado para realizar o *Maximum Power Point Tracking* (MPPT) foi o Perturbe e Observe (P&O). O P&O convencional é funcional em situações onde não ocorre sombreamento parcial dos painéis. Quando ocorre sombreamento, há outras técnicas de MPPT conforme discutido em (Oliveira et al., 2015). Portanto, durante a realização dos experimentos é fundamental garantir que os painéis sejam posicionados sempre em uma área aberta, de modo a não ocorrer sombreamento por qualquer estrutura urbana (Ex: paredes, muros etc) (FERNÁNDEZ-AHUMADA, 2020).

Em sistemas do tipo *Off-Grid* a bateria é um fator limitante da potência produzida pelos painéis fotovoltaicos, pois para se preservar a vida útil da mesma deve-se respeitar sua corrente máxima de carga I_{max} (na etapa de corrente constante) e sua tensão máxima V_{max} de carga (na etapa tensão constante) (SERHAN; AHMED, 2018). Em nossa aplicação, modificou-se o algoritmo padrão do P&O, originando um novo algoritmo P&O com uma etapa limitadora de potência, com o intuito de realizar o carregamento da bateria, conforme o fluxograma da Figura 14.

Figura 14 – Fluxograma do algoritmo P&O com limitador para carregamento de bateria.



Fonte: Autoria própria

2.10 Instrumentação eletrônica, microcontrolador e circuitos de potência empregados na confecção da placa de circuito impresso.

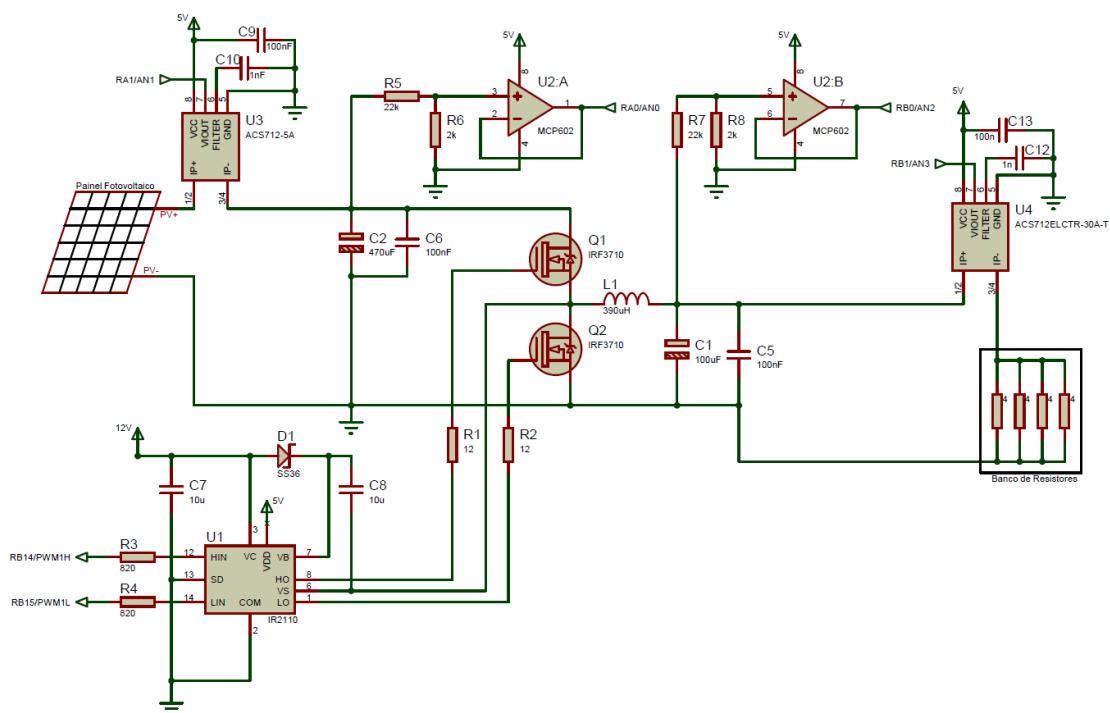
Para embarcar todos os controles e algoritmos optou-se por um microcontrolador da Microchip modelo dsPIC33EV256GM104. O seu ADC foi configurado para uma frequência amostral de 514,7 kcps e precisão de 12 bits. Para acionamento do conversor Buck síncrono configurou-se o PWM1 a uma frequência de 100 KHz, já para acionamento dos drivers dos atuadores dos eixos X e Y, foram utilizados os PWM2 e PWM3 operando a 2,14 KHz. O TIMER1 gera uma interrupção periódica de 10 KHz, nela executa-se o controlador PI da tensão dos painéis fotovoltaicos e os filtros IIR, já o TIMER2 opera a 500 Hz e executa o controlador responsável por rastrear a angulação de máxima incidência solar e o método de MPPT. Para comunicar o dsPIC com o MATLAB, utilizou-se a comunicação UART com um *BaudRate* 921,6 Kbps, juntamente com um conversor USB serial da Prolific modelo PL2303 HXD, que é compatível com essa elevada velocidade de comunicação.

A interface de programação utilizada foi a proprietária da Microchip o MPLABX IDE V5.35 e o compilador foi o XC16 V1.5. O código completo em linguagem C++ que o microcontrolador executa é mostrado no apêndice A. O algoritmo implementado no MATLAB para comunicação com o dsPIC é mostrado no apêndice B.

No conversor Buck síncrono optou-se por MOSFETs canal N da International Rectifier modelo IRF3710, para seu acionamento utilizou-se um driver da mesma fabricante modelo IR2110. Para leitura das tensões utiliza-se atenuadores resistivos, seguidos de *Buffers* e para leitura das correntes aplicou-se sensores do tipo *Hall*. O circuito completo pode ser visto na

Figura 15.

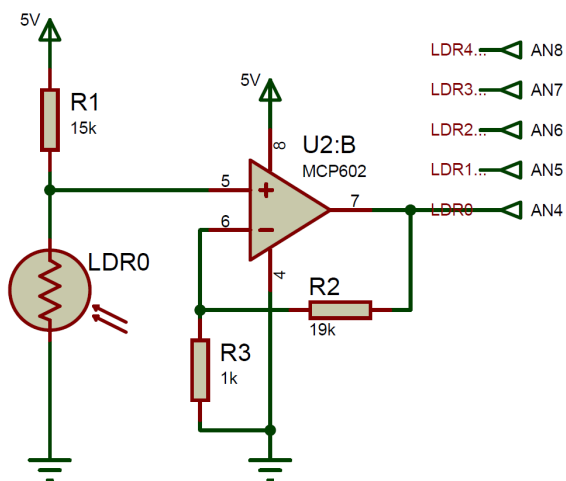
Figura 15 – Esquemático Buck síncrono com PV na entrada.



Fonte: Autoria própria

Para realizar a leitura dos cinco LDRs montou-se um divisor resistivo com cada um deles e a tensão do divisor é amplificada com ganho $G = 20$. O intervalo de leitura dos LDRs pode variar de 0Ω a aproximadamente 780Ω , fato esse que permitiu adotar a aproximação $V_{div} = 5 \cdot \frac{R_{LDR}}{R_{LDR} + 15000} = R_{LDR} \cdot 3,168 \cdot 10^{-4}$ para leitura da tensão do divisor resistivo. O esquemático do circuito está ilustrado na Figura 16.

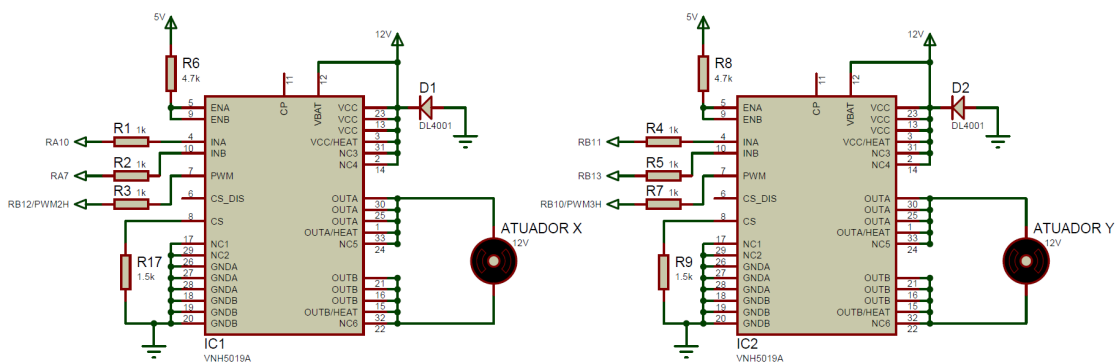
Figura 16 – Esquemático leitura da resistência dos LDRs



Fonte: Autoria própria

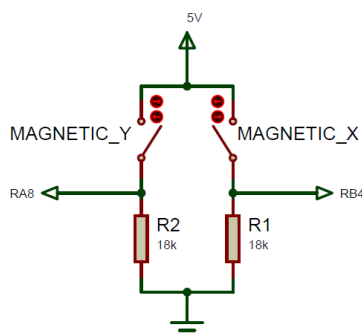
Para acionamento dos atuadores utilizou-se duas pontes H de classificação automotiva da marca STMicroelectronics modelo VNH5019. As conexões dos drivers e dos sensores magnéticos com o microcontrolador são mostradas respectivamente nas Figuras 17 e 18.

Figura 17 – Esquemático das pontes H com os atuadores



Fonte: Autoria própria

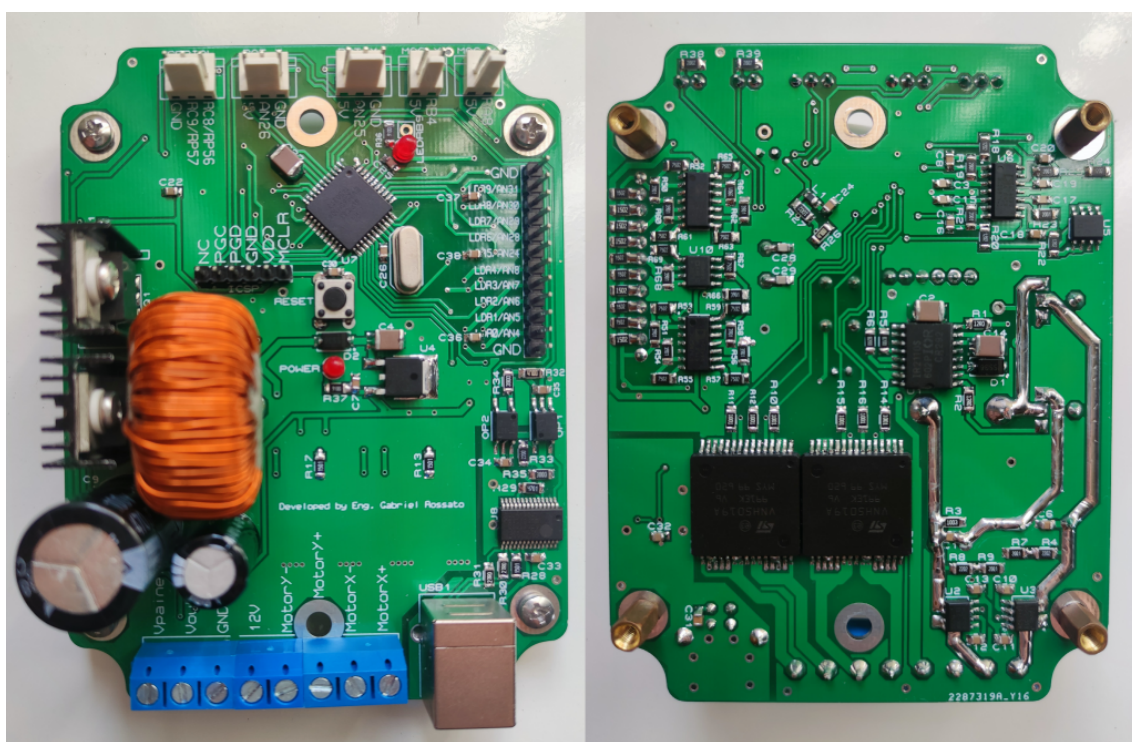
Figura 18 – Esquemático dos sensores magnéticos de meio de curso



Fonte: Autoria própria

Em um software de projeto de placas de circuito impresso confeccionou-se uma PCB, contendo todos os circuitos eletrônicos descritos nessa subseção. Na Figura 19 é mostrada a placa fabricada, já com todos seus componentes montados.

Figura 19 – Placa de circuito impresso final utilizada no sistema.



Fonte: Autoria própria

3 RESULTADOS E DISCUSSÕES

Nessa seção serão apresentados os resultados experimentais, juntamente com discussão e análise.

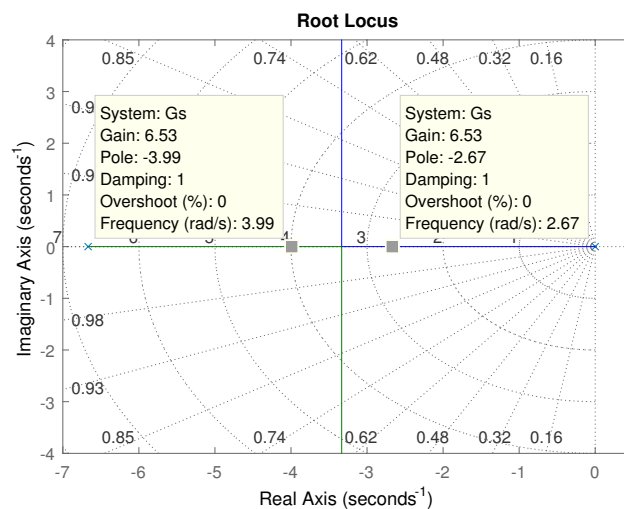
3.1 Projeto do controlador proporcional para o seguidor solar

Conforme mostrado na identificação do sistema na subseção 2.3, os eixos X e Y são representados respectivamente pelas funções de transferência (3) e (4). Note que o sistema é do tipo 1, ou seja, um simples controle proporcional gera erro nulo no regime permanente para uma entrada do tipo degrau.

Tem-se como objetivo uma resposta superamortecida em malha fechada, ou seja, sem a presença de *overshoot* (polos localizados sobre o eixo real). O tempo de acomodação t_a do sistema é ditado pela posição ρ do polo mais próximo da origem, conforme Equação $t_a = \frac{4}{\rho}$ (OGATA, 2009). Assim é possível ajustar o ganho proporcional Kp pela posição do polo dominante no método do lugar das raízes.

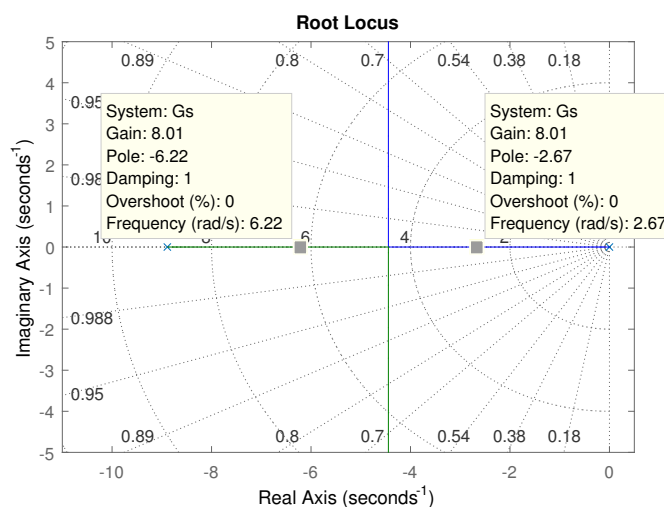
Deseja-se que o sistema tenha um $t_a = 1,5$ s, para isso o polo dominante deve estar posicionado no eixo real em $\rho = -2,67$. Por meio do lugar das raízes das funções de transferência obtém-se ganho proporcional: para o eixo X $Kp_X = 6,53$ e para o eixo Y $Kp_Y = 8,01$, respectivamente ilustrados nas Figuras 20 e 21.

Figura 20 – Lugar das raízes da função de transferência eixo X.



Fonte: Autoria própria

Figura 21 – Lugar das raízes da função de transferência eixo Y.



Fonte: Autoria própria

3.2 Experimento: controladores atuando no seguidor solar

A seguir é mostrado os resultados dos experimentos para o controlador proporcional projetado na subseção 3.1, que opera associada com a lógica de histerese, mostrada na subseção 2.4. O controlador proporcional com histerese foi implementado no hardware projetado. Perturbou-se o sistema aleatoriamente, e para cada um dos eixos, os dados experimentais foram coletados (realizou-se a leitura dos sensores de angulação, a saída do controlador clássico proporcional e da lógica de histerese), veja as Figuras 22 e 23 (O eixo da saída dos sistemas X e Y (Azul) dos gráficos são adimensionais, devido normalização mostrada na subseção 2.2).

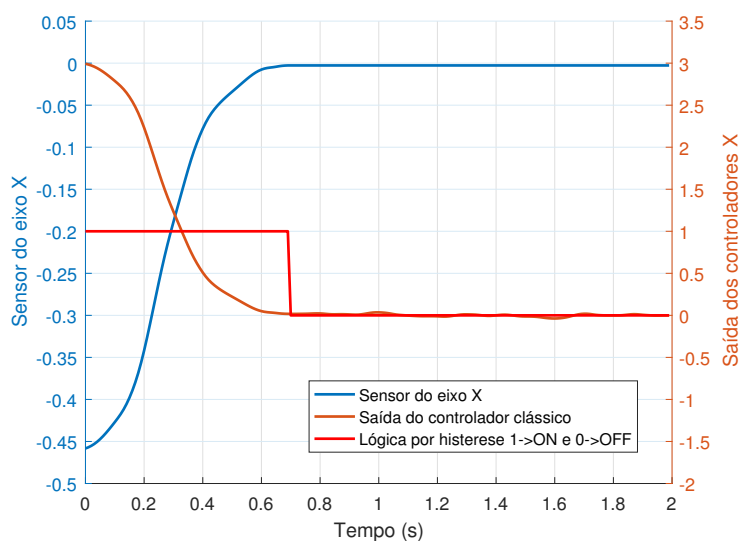
Observa-se que o controle obteve êxito em alcançar a referência nula em ambos os eixos, mostrando que o arranjo de painéis fotovoltaicos foi alinhado com a direção da máxima incidência de irradiação solar, em aproximadamente 0,7 segundos para eixo X e 1,0 segundo para o Y. É possível observar também a lógica de histerese em ação: quando a posição do eixo alcançou erro próximo a zero (valor ligeiramente negativo), então a lógica desabilitou o controlador clássico, conforme projetado.

A plataforma seguidora solar em funcionamento é mostrada em detalhes no vídeo, por meio do link: <<https://www.youtube.com/watch?v=ezxvYEWXvUM>>

3.3 Projeto e teste do controlador PI do conversor Buck síncrono

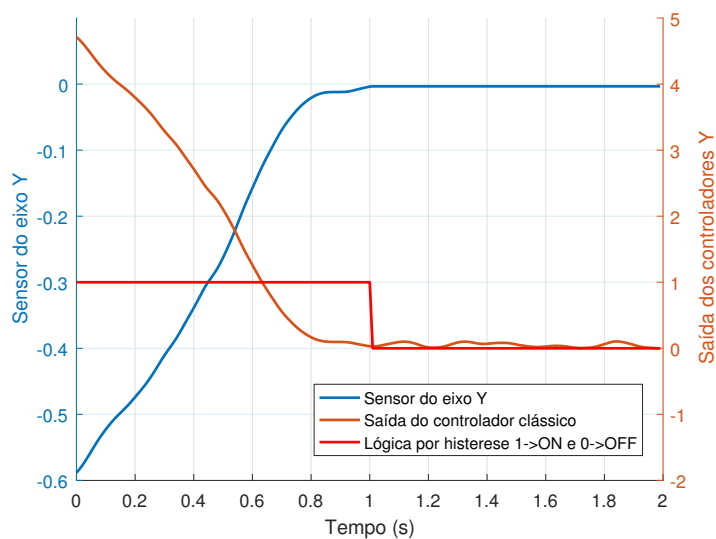
Para o projeto do controlador PI do conversor Buck, utilizou-se a função de transferência mostrada em Equação 12 (O sinal negativo indica uma razão inversamente proporcional entre \widehat{V}_{pv} e \widehat{d}). Para projeto do controlador considerou-se $\overline{V}_{pv} = 2 \cdot V_{mpp} = 35,48 \text{ V}$, $\overline{I}_{pv} = I_{mpp} = 2,84 \text{ A}$ e $R_o = 1 \Omega$. Foram adotadas as capacitâncias e a indutância mostradas na Tabela 2, a resistência do indutor $R_L = 0,1 \Omega$ e utilizou-se uma carga de saída $R_o = 1 \Omega$. Aplicando as

Figura 22 – Resposta no eixo X para perturbação.



Fonte: Autoria própria

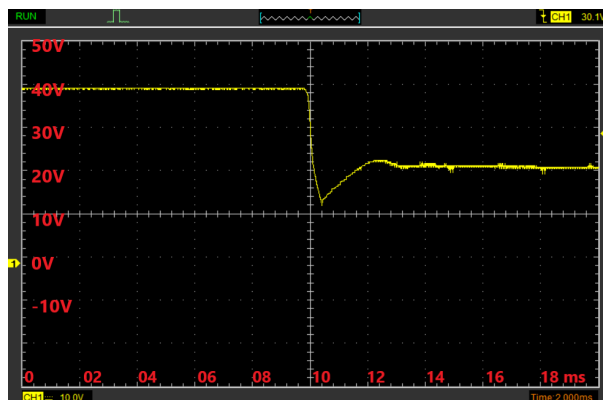
Figura 23 – Resposta no eixo Y para perturbação.



Fonte: Autoria própria

Equações de 13 a 16 obtém-se os valores médios associados a função de transferência.

Deseja-se que o sistema em malha aberta tenha uma margem de fase igual 75° , e uma frequência de cruzamento com o ganho 0db igual a 500 Hz. Baseado no método de projeto desenvolvido por (Angélico; Campanhol; Oliveira da Silva, 2014), calculou-se os ganhos para o controlador, $K_i = 9,8012$ e $K_p = 0,0893$. Na Figura 24 é mostrado em funcionamento no sistema real o controle PI, programou-se uma referência de tensão igual a 20 V no dsPIC e se mensurou a tensão dos painéis fotovoltaicos. Observa-se que a resposta do controlador ficou dentro das especificações projetadas acima.

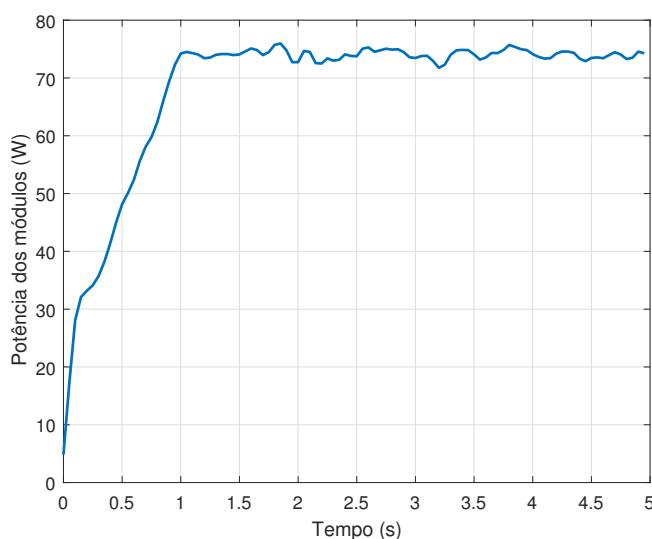
Figura 24 – Controle PI atuando na tensão de entrada V_{pv} dos painéis

Fonte: Autoria própria

3.4 Experimentos: Métodos de MPPT em operação e sistema Off-Grid carregando uma bateria

Primeiramente realizou-se teste do algoritmo de MPPT P&O convencional. Para este teste, utilizou-se uma carga na saída do conversor Buck $R_o = 1 \Omega$. Foi adotado um $\Delta V_{ref} = 0,05 \text{ V}$ e a uma frequência de atualização igual a 500 Hz. Na Figura 25 é apresentada a potência extraída do arranjo de painéis, já na Figura 26 é mostrado em funcionamento o controle PI, com o método P&O comandando sua tensão de referência. Observa-se que o método se estabiliza e consegue buscar a máxima potência para as condições nas quais foi realizado o teste.

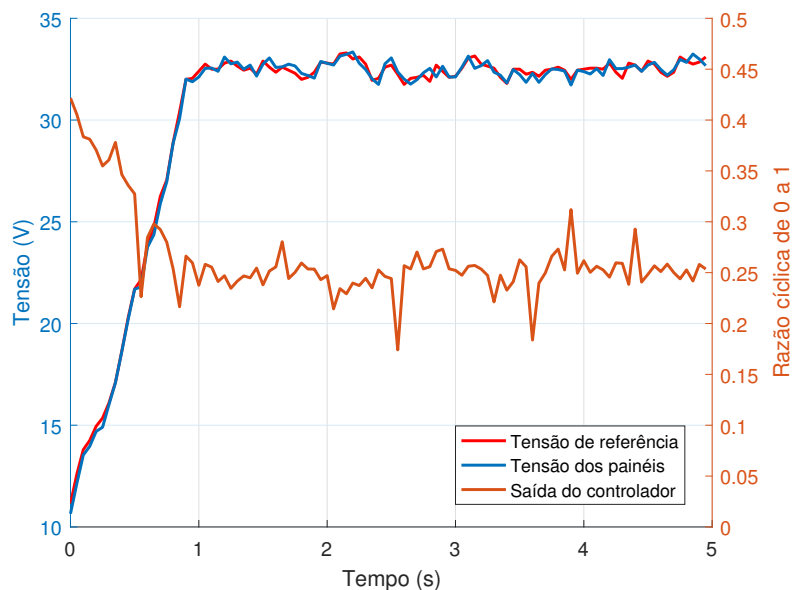
Figura 25 – Método P&O convencional realizando o MPPT no sistema



Fonte: Autoria própria

Aplicou-se o conversor Buck para carregar uma bateria de chumbo-ácido de 6 células e com capacidade de 6 Ah. Foi utilizado o algoritmo P&O modificado, com a etapa que limita a potência dos painéis fotovoltaicos, conforme o fluxograma da Figura 14. Inicialmente a

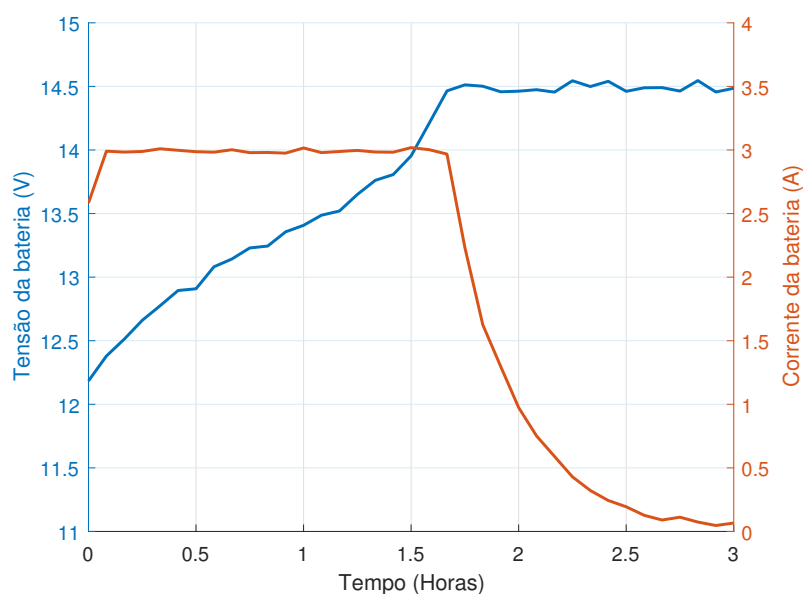
Figura 26 – Controle PI operando na tensão do painéis.



Fonte: Autoria própria

bateria estava semi carregada, contendo uma tensão de 12,0V. Adotou-se um saturador para a corrente de carregamento $I_{max} = 3,0A$ e para tensão $V_{max} = 14,5V$. A Figura 27 apresenta os dados experimentais da tensão e corrente da bateria durante o experimento. Pode-se observar que foi possível carregar a bateria corretamente, fato que corrobora a eficácia do algoritmo apresentado.

Figura 27 – Teste de carga bateria de chumbo-ácido.



Fonte: Autoria própria

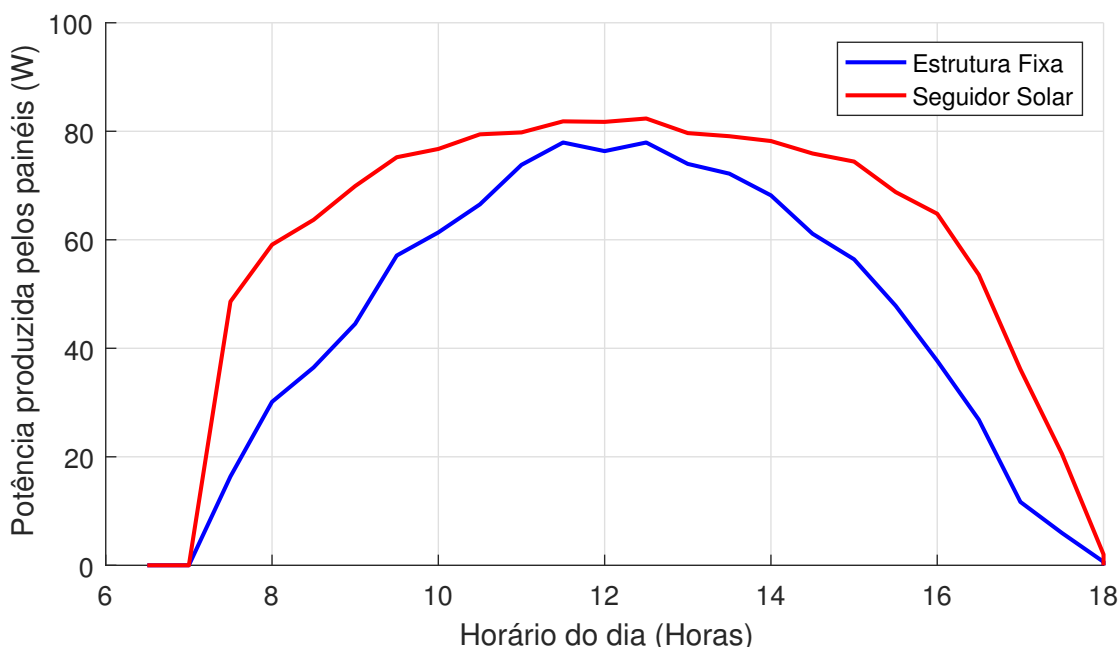
3.5 Experimento para a potência produzida: comparação entre o seguidor solar e estrutura fixa

Realizou-se um ensaio na cidade de Cornélio Procópio/PR (23°10'50"S 50°39'20"W) no dia 22 de agosto de 2021, com início às 6:30 AM e término às 6:00 PM. Os dados experimentais foram coletados a cada 30 minutos. Utilizou-se uma carga resistiva constante $R_o = 1 \Omega$ (100 watts) na saída do conversor, com intuito de não limitar a potência produzida pelos painéis no decorrer desse teste. O algoritmo de MPPT utilizado foi o P&O convencional.

A metodologia aplicada foi a seguinte: primeiramente realizou-se o teste com estrutura fixa e imediatamente após habilitou-se o seguidor solar. Para a estrutura fixa, os painéis foram orientados na angulação ótima e permaneceram fixos, ou seja, foram mantidos estáticos independente do horário do dia. Esta angulação ótima está relacionada aos dados geodésicos do local, onde o experimento foi realizado: painel fixado em ângulo de 20° e alinhado ao Norte magnético do planeta (aferido através de bússola manual).

Na outra situação do sistema seguidor solar, comandou-se o rastreamento da angulação de máxima incidência de irradiação solar. O resultado comparativo da potência produzida pelos painéis para ambos os testes é mostrado na Figura 28, é importante ressaltar que se considerou os dados para o cálculo da potência média somente após o MPPT se estabilizar.

Figura 28 – Potência produzida pelos painéis da estrutura fixa comparada com a seguidora.



Fonte: Autoria própria

Após computar a área sob as curvas mostradas na Figura 28, obteve-se a energia produzida ao longo do dia, que é resumida na Tabela 4. Nota-se que o sistema seguidor solar obteve um incremento de **32,42 %** na energia produzida, quando comparado a estrutura fixa. Esta evidência experimental encontra suporte na literatura, pois outros sistemas seguidores solar

biaxiais propostos foram capazes de prover incrementos de 25% em (Lee; Rahim, 2013)(Malásia) e 35% em (Mustafa et al., 2018)(Bagdá/Iraque). O ganho obtido aqui foi levemente menor que o sensor piramidal com base quadrada, porém o sensor tronco piramidal proposto aqui possui as vantagens já descritas no decorrer dessa dissertação e temos que levar em consideração o fato dos sistemas estarem em regiões geográficas distintas.

O consumo dos atuadores lineares não excedeu 2W, seu tempo de acionamento sempre foi inferior a 10 segundos e durante o dia todo são realizados 9 acionamentos, devido a lógica de histerese. Assim podemos estimar um gasto energético máximo de 0,05Wh (que representa menos de 0,007% da energia total produzida pelo sistema seguidor), comprovando assim a elevada eficiência do método de controle proposto.

Tabela 4 – Energia produzida pelos painéis fotovoltaicos durante o dia

Estrutura Fixa	540,50 Wh
Seguidor Solar	715,73 Wh

4 CONCLUSÃO

Esta dissertação apresentou o projeto e a construção de um sistema seguidor solar de dois eixos. Tal seguidor tem como função rotacionar dois painéis fotovoltaicos, de modo a mantê-los perpendicular com relação a incidência de luz solar. Esta estratégia visa maximizar a produção de energia elétrica. O projeto envolve conceitos de instrumentação, controladores, sistemas microcontrolados e conversor estático de potência, associado a um método de MPPT.

A estrutura mecatrônica se comportou conforme o projetado, seus atuadores foram suficientes para realizar a rotação dos painéis. O método de controle proposto se mostrou eficiente em gerar uma economia de energia na rotação da estrutura e possui a vantagem de diminuir desgaste mecânico devido a histerese associada. O sensor com formato de tronco de pirâmide de base quadrada se mostrou eficaz para este tipo de aplicação. O algoritmo de MPPT com etapa de limitação de potência realizou o carregamento da bateria de maneira adequada, respeitando as etapas de tensão e corrente constante. Todos os circuitos eletrônicos projetados e implementados apresentaram um correto funcionamento. De maneira geral, podemos afirmar que o sistema atendeu todos os objetivos propostos.

Dados experimentais indicam que a estrutura seguidora solar proporcionou um incremento significativo na energia produzida, de aproximadamente 32%, quando comparado com uma estrutura fixa. Para trabalhos futuros, sugere-se realizar testes com uma duração prolongada, para aprimorar análise estatística dos dados. O sistema poderia se beneficiar também da utilização de outros tipos de sensores para leitura da luminosidade. Com intuito de aproveitar melhor a energia produzida pela painéis fotovoltaicos, futuramente pode-se injetá-la rede elétrica com utilização de sistemas do tipo *On-Grid*.

Referências

- AL-ROUSAN, N.; ISA, N. A. M.; DESA, M. K. M. Advances in solar photovoltaic tracking systems: A review. **Renewable and Sustainable Energy Reviews**, v. 82, p. 2548–2569, 2018. Citado na página 1.
- ALDOSARY, A. et al. A modified shuffled frog algorithm to improve MPPT controller in PV system with storage batteries under variable atmospheric conditions. **Control Engineering Practice**, v. 112, p. 104831, 2021. Citado na página 2.
- Angélico, B. A.; Campanhol, L. B. G.; Oliveira da Silva, S. A. Proportional–integral/proportional–integral–derivative tuning procedure of a single-phase shunt active power filter using bode diagram. **IET Power Electronics**, v. 7, n. 10, p. 2647–2659, 2014. Citado na página 22.
- BAHRAMI, A.; OKOYE, C. O.; ATIKOL, U. The effect of latitude on the performance of different solar trackers in europe and africa. **Applied Energy**, v. 177, p. 896–906, 2016. ISSN 0306-2619. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0306261916307085>>. Citado na página 2.
- BARBI, I.; MARTINS, D. C. **Conversores CC-CC Básicos Não Isolados**. 2ed. ed. Florianópolis, Santa Catarina, BR: Universidade Federal de Santa Catarina, 2006. 15 - 29 p. Citado 2 vezes nas páginas 10 e 12.
- Beltran A., J.; Gonzalez Rubio S., J. L.; Garcia-Beltran, C. D. Design, manufacturing and performance test of a solar tracker made by a embedded control. In: **Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007)**. [S.l.: s.n.], 2007. p. 129–134. Citado na página 2.
- BEN. **Balanco Energético Nacional 2020 / Matriz Energética Nacional 2020 - Ano Base 2019**. 2020. Disponível em: <<https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/balanco-energetico-nacional-2020>>. Acesso em: 02 de Agosto de 2020. Citado na página 1.
- CORPORATION, E. C. **DC Motors, Speed Controls, Servo Systems**. 3ed. ed. Oxford: Pergamon Press, 1972. 395 - 418 p. ISBN 978-0-08-021714-7. Citado na página 6.
- de Brito, M. A. G. et al. Evaluation of the main MPPT techniques for photovoltaic applications. **IEEE Trans. Ind. Electr.**, v. 60, n. 3, p. 1156–1167, 2013. Citado na página 1.
- ERICKSON, R. W.; MAKSIMOVIC, D. **Fundamentals of Power Electronics**. 2ed. ed. The Salmon Tower Building, Nova Iorque, Nova York, EUA: Springer, 2001. 187 - 200 p. ISBN 0792372700. Citado 2 vezes nas páginas 10 e 11.
- Femia, N. et al. A technique for improving PO MPPT performances of double-stage grid-connected photovoltaic systems. **IEEE Trans. Ind. Electr.**, v. 56, n. 11, p. 4473–4482, 2009. Citado na página 10.
- FERNÁNDEZ-AHUMADA, L. A novel backtracking approach for two-axis solar pv tracking plants. **Renewable Energy**, v. 145, p. 1214–1221, 2020. ISSN 0960-1481. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0960148119308900>>. Citado na página 15.

Ghaffari, A.; Mohammadiasl, E. Calculating the frequency of oscillation for servo axes distressed by clearance or preloading. **IEEE/ASME Transactions on Mechatronics**, v. 18, n. 3, p. 922–931, 2013. Citado na página 8.

Huang, F.; Tien, D.; Or, J. A microcontroller based automatic sun tracker combined with a new solar energy conversion unit. In: **1998 International Conference on Power Electronic Drives and Energy Systems for Industrial Growth, 1998. Proceedings**. [S.l.: s.n.], 1998. v. 1, p. 488–492 Vol.1. Citado na página 1.

KABALCI, E. Design and implementation of control algorithms for single-axis sun tracking systems. **Journal of Power Technologies**, v. 100, n. 1, p. 32–42, 2020. ISSN 2083-4195. Disponível em: <<https://papers.itc.pw.edu.pl/index.php/JPT/article/view/1460>>. Citado na página 2.

Koyuncu, B.; Balasubramanian, K. A microprocessor controlled automatic sun tracker. **IEEE Transactions on Consumer Electronics**, v. 37, n. 4, p. 913–917, 1991. Citado na página 1.

Kumar, G. et al. A study of a gun-turret assembly in an armored tank using model predictive control. In: **2009 American Control Conference**. [S.l.: s.n.], 2009. p. 4848–4853. Citado na página 8.

Lee, J. F.; Rahim, N. A. Performance comparison of dual-axis solar tracker vs static solar system in malaysia. In: **2013 IEEE Conference on Clean Energy and Technology (CEAT)**. [S.l.: s.n.], 2013. p. 102–107. Citado 2 vezes nas páginas 1 e 26.

Luque-Heredia, I. et al. A subdegree precision sun tracker for 1000x microconcentrator modules. In: **3rd World Conf. Photovoltaic Energy Conversion, 2003. Proceedings of**. [S.l.: s.n.], 2003. v. 1, p. 857–860 Vol.1. Citado na página 1.

Mustafa, F. I. et al. Simple design and implementation of solar tracking system two axis with four sensors for baghdad city. In: **2018 9th International Renewable Energy Congress (IREC)**. [S.l.: s.n.], 2018. p. 1–5. Citado 4 vezes nas páginas 1, 2, 5 e 26.

OGATA, K. **Modern Control Engineering**. 5ed. ed. Upper Saddle River, Nova Jersey, EUA: Prentice Hall, 2009. 159 - 178, 269 - 289 p. ISBN 0136156738. Citado 2 vezes nas páginas 7 e 20.

Oliveira, F. M. et al. Application of pso method for maximum power point extraction in photovoltaic systems under partial shading conditions. In: **2015 IEEE 13th Brazilian Power Electronics Conference and 1st Southern Power Electronics Conference (COBEP/S-PEC)**. [S.l.: s.n.], 2015. p. 1–6. Citado 2 vezes nas páginas 1 e 15.

OPPENHEIM, A. V. **Processamento em Tempo Discreto de Sinais**. 3ed. ed. Campinas, São Paulo, BR: Pearson Universidades, 2013. 294 - 314 p. ISBN 858143102X. Citado na página 10.

Portolan dos Santos Ísis; RüTHER, R. Limitations in solar module azimuth and tilt angles in building integrated photovoltaics at low latitude tropical sites in brazil. **Renewable Energy**, v. 63, p. 116–124, 2014. ISSN 0960-1481. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0960148113004758>>. Citado 2 vezes nas páginas 2 e 4.

PRINSLOO, G.; DOBSON, R. **Solar Tracking, Sun Tracking, Sun Tracker, Solar Tracker, Follow Sun, Sun Position**. [S.l.: s.n.], 2015. 34–42 p. ISBN 978-0-620-61576-1. Citado na página 1.

- Pritchard, D. Sun tracking by peak power positioning for photovoltaic concentrator arrays. **IEEE Control Systems Magazine**, v. 3, n. 3, p. 2–8, 1983. Citado na página 1.
- Sawant, A. et al. Design and analysis of automated dual axis solar tracker based on light sensors. In: **2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC**. [S.l.: s.n.], 2018. p. 454–459. Citado na página 2.
- Sera, D.; Teodorescu, R.; Rodriguez, P. PV panel model based on datasheet values. In: **2007 IEEE International Symposium on Ind. Electr.** [S.l.: s.n.], 2007. p. 2392–2396. Citado na página 14.
- SERHAN, H. A.; AHMED, E. M. Effect of the different charging techniques on battery life-time: Review. In: **2018 International Conference on Innovative Trends in Computer Engineering (ITCE)**. [S.l.: s.n.], 2018. p. 421–426. Citado na página 15.
- Villalva, M. G.; Gazoli, J. R.; Filho, E. R. Comprehensive approach to modeling and simulation of photovoltaic arrays. **IEEE Transactions on Power Electronics**, v. 24, n. 5, p. 1198–1208, 2009. Citado na página 13.
- Yazidi, A. et al. Low cost two-axis solar tracker with high precision positioning. In: **2006 First International Symposium on Environment Identities and Mediterranean Area**. [S.l.: s.n.], 2006. p. 211–216. Citado na página 2.
- Zhan, T. et al. Design and implementation of the dual-axis solar tracking system. In: **2013 IEEE 37th Annual Computer Software and Applications Conference**. [S.l.: s.n.], 2013. p. 276–277. Citado na página 1.

Apêndices

APÊNDICE A – Código em C++ implementado no dsPIC33EV256GM104 compilador XC16 V1.5

```

1
2 /*
3  * File:   SolarTCC.c
4  * Author: Gabriel Rossato Francisco
5  */
6 // DSPIC33EV256GM104 Configuration Bit Settings
7 #pragma config BWRP = OFF           // Boot Segment may be written
8 #pragma config BSS = DISABLED      // No Protection (other than BWRP)
9 #pragma config BSS2 = OFF          // Boot Segment Control Bit (No Boot Segment)
10 #pragma config GWRP = OFF         // General Segment may be written
11 #pragma config GSS = DISABLED     // No Protection (other than GWRP)
12 #pragma config CWRP = OFF        // Configuration Segment may be written
13 #pragma config CSS = DISABLED     // No Protection (other than CWRP)
14 #pragma config AIVTDIS = DISABLE // Disable Alternate Vector Table
15 #pragma config BSLIM = 0x1FFF     // Boot Segment Flash Page Address Limit (0-0x1FFF)
16 #pragma config FNOSC = FRC        // Initial oscillator (Internal Fast RC (FRC))
17 #pragma config IESO = ON          // Start with FRC automatically switch selected
18 #pragma config POSCMD = HS        // Primary Oscillator HS Crystal Oscillator mode
19 #pragma config OSCIOFNC = OFF     // Pin I/O Function Enable Bit OSC2 is clock output
20 #pragma config IOL1WAY = ON       // Peripheral Pin Select (Allow Only One reconfiguration)
21 #pragma config FCKSM = CSECMD     // Clock Switching is enabled, Fail-safe is disabled
22 #pragma config PLLKEN = ON        // PLL Lock Enable Bit (Clock switch to PLL)
23 #pragma config WDTPOST = PS32768 // Watchdog Timer Postscaler Bits (1:32,768)
24 #pragma config WDTPRE = PR128     // Watchdog Timer Prescaler Bit (1:128)
25 #pragma config FWDTEN = OFF       // Watchdog Timer Enable Bits (WDT and SWDTEN Disabled)
26 #pragma config WINDIS = OFF       // Watchdog timer in Non-Window Mode
27 #pragma config WDTWIN = WIN25     // WDT Window is 25% of WDT period
28 #pragma config BOREN0 = ON        // Brown Out Reset Detection Bit (BOR is Enabled)
29 #pragma config ICS = PGD2         // ICSP PINS Communicate on PGEC2 and PGED2 !!!!!!!!!!!!!!!
30 #pragma config DMTIVTL = 0xFFFF   // Lower 16 bits of 32 bit DMT window interval
31 #pragma config DMTIVTH = 0xFFFF   // Upper 16 bits of 32 bit DMT window interval
32 #pragma config DMTCNTH = 0xFFFF   // Lower 16 bits of 32 bit DMT instruction count time-out
33 #pragma config DMTCNTH = 0xFFFF   // Upper 16 Bits of 32 Bit DMT Instruction Count Time-Out
34 #pragma config DMTEN = ENABLE     // Dead Man Timer is Enabled and cannot be disabled
35 #pragma config PWMLOCK = OFF      // PWM Lock Disable Bit !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
36 #pragma config ALTI2C1 = OFF      // I2C1 mapped to SDA1/SCL1 pins
37 #pragma config CTXT1 = NONE       // Interrupt Priority Level Register Set 1 (Not Assigned)
38 #pragma config CTXT2 = NONE       // Interrupt Priority Level Register Set 2 (Not Assigned)
39
40
41 #include "xc.h"
42 #include "math.h"
43
44 char i=0;
45 char ReceivedChar = 0;
46 char ReceiverBuffer[10];
47
48 union Data1A{
49     char SendChar [4];
50     float TensionPainel;
51 };
52
53 union Data1B{

```

```
54     char SendChar [4];
55     float ReferenciaVPainel;
56 };
57
58 union Data1C{
59     char SendChar [4];
60     float ControlVPainelOut;
61 };
62
63 union Data2{
64     char SendChar [4];
65     float CurrentPainel;
66 };
67
68 union Data3A{
69     char SendChar [4];
70     float TensionLoad;
71 };
72
73 union Data3B{
74     char SendChar [4];
75     float CurrentLoad;
76 };
77
78
79 union Data4{
80     char SendChar [4];
81     float SensorX;
82 };
83
84 union Data5{
85     char SendChar [4];
86     float SensorY;
87 };
88
89 union Data6{
90     char SendChar [4];
91     float ControlX;
92 };
93
94 union Data7{
95     char SendChar [4];
96     float ControlY;
97 };
98
99 union Data1A data1A;
100 union Data1B data1B;
101 union Data1C data1C;
102 union Data2 data2;
103 union Data3A data3A;
104 union Data3B data3B;
105
106 union Data4 data4;
107 union Data5 data5;
108 union Data6 data6;
109 union Data7 data7;
110
111
112 int EnableMPPT = 0;
113 int EnableControlVpv = 0;
```



```
114 int EnableCharging = 0;
115
116 double TensionPV = 0;
117 double TensionPV_1 = 0;
118 double CurrentPV = 0;
119 double CurrentPV_1 = 0;
120 double PowerPV = 0;
121 double PowerPV_1 = 0;
122 double TensionLoad = 0;
123 double CurrentLoad = 0;
124
125 //PI conversor Buck
126 double ReferenciaVpv = 0;
127 double ErroVpv = 0;
128 double ErroVpv_1 = 0;
129 double IntegratorPI = 0;
130 double AntiWindup = 0;
131 double ControlPIoutput = 0;
132
133
134 int p=0;
135 int n=0;
136 int n_1=0;
137 int n_2=0;
138 double Sensor_LDR[5][3];
139 double FilterON_LDR [5][3];
140 double FilterOFF_LDR [5][3];
141
142 double SumLDRs = 0;
143 double XaxisError = 0;
144 double XaxisControl = 0;
145 double YaxisError = 0;
146 double YaxisControl = 0;
147
148 int enableFunctionTransfer=0; //Levantar FuncaoDeTransferencia
149
150 int enableRastreamentoDoSol=0; //Realiza o rastreamento do sol
151
152 int moveX=0;
153 int moveY=0;
154
155 int enableX=0;
156 int enableY=0;
157
158 int perpenXRight = 0;
159 int perpenXLeft = 0;
160 int perpenYRight = 0;
161 int penpenYLeft = 0;
162
163
164 void __attribute__((__interrupt__, no_auto_psv)) _AD1Interrupt(void)
165 {
166
167     AD1CON1bits.ASAM = 0; //Desabilita o AutoSAMP ate que seja escrito 1 novamente
168
169
170     //Adianta a comutacao um tempo de amostragem
171     //Porque eu leio a ADC, depois comuto o MUX
172     switch((p+1)%5)
173     {
```

```
174     case 0: AD1CSSH = 0x0000;
175             AD1CSSL = 0x001F; // Auto Scan AN0 to AN3 and AN4
176             break;
177
178     case 1: AD1CSSH = 0x0000;
179             AD1CSSL = 0x002F; // Auto Scan AN0 to AN3 and AN5
180             break;
181
182     case 2: AD1CSSH = 0x0000;
183             AD1CSSL = 0x004F; // Auto Scan AN0 to AN3 and AN6
184             break;
185
186     case 3: AD1CSSH = 0x0000;
187             AD1CSSL = 0x008F; // Auto Scan AN0 to AN3 and AN7
188             break;
189
190     case 4: AD1CSSH = 0x0000;
191             AD1CSSL = 0x010F; // Auto Scan AN0 to AN3 and AN8
192             break;
193
194 }
195
196
197 IFS0bits.AD1IF = 0; //Clear adc Flag interrupt
198
199 }
200
201
202 void __attribute__((__interrupt__, no_auto_psv)) _T1Interrupt(void) //10kHz sample
203 {
204     //LATBbits.LATB9 = i;
205     //i=~i;
206
207     //LDR reading ADC mux comutate
208
209     Sensor_LDR[p][n] = ADC1BUF4;
210
211     //Filters IIR 2 order
212     switch (n)
213     {
214         case 0: n_1 = 2;
215                 n_2 = 1; break;
216
217         case 1: n_1 = 0;
218                 n_2 = 2; break;
219
220         case 2: n_1 = 1;
221                 n_2 = 0; break;
222     }
223
224
225     //2 order IIR LowPass Chebyshev Filter
226     //fsample = 2kHz;
227
228     //fc = 0.1Hz
229     FilterON_LDR[p][n] = 0.00999692396850083*Sensor_LDR[p][n]
230                        - 0.0199936506062813*Sensor_LDR[p][n_1]
231                        + 0.00999692396850083*Sensor_LDR[p][n_2]
232                        + 1.99937482849879*FilterON_LDR[p][n_1]
233                        - 0.999375025829507*FilterON_LDR[p][n_2];
```

```

234
235 //fc = 10Hz
236 FilterOFF_LDR[p][n] = 0.0101758827036283*Sensor_LDR[p][n]
237                       - 0.0184071801973730*Sensor_LDR[p][n_1]
238                       + 0.0101758827036283*Sensor_LDR[p][n_2]
239                       + 1.93697498762654*FilterOFF_LDR[p][n_1]
240                       - 0.938919572836420*FilterOFF_LDR[p][n_2];
241
242
243 //Commute LDRs ports for reading the sensors
244 if (p>=4)
245 {
246     p=0;
247     if (n>=2) n=0;
248     else n = n+1;
249 }
250 else p = p+1;
251
252
253
254 //Current Tension Mesure
255 TensionPV = ADC1BUF0*0.01465201465; //5*12/4095 Resistive Divisor G=1/12
256
257 //Deve ser feito assim. Nao juntar com a de baixo
258 CurrentPV = ADC1BUF1;
259 //(ADC-2047)*(5*1000)/(185*4095) Current Sensor ACS712 5A (185 mV/A)
260 CurrentPV = (CurrentPV - 2047)*0.0066000066;
261
262 TensionLoad = ADC1BUF2*0.01465201465; //5*12/4095 Resistive Divisor G=1/12
263
264 //Deve ser feito assim. Nao juntar com a de baixo
265 CurrentLoad = ADC1BUF3;
266 //(ADC-2047)*(5*1000)/(66*4095) Current Sensor ACS712 30A (66 mV/A)
267 CurrentLoad =(CurrentLoad -2047)*0.0185000185;
268
269
270 AD1CON1bits.ASAM = 1; //Habilita o AutoSAMP novamente para realizar a nova leitura das ADCs
271
272
273 //Control PI para controlar tencao Vpv
274 if (EnableControlVpv == 1)
275 {
276     //fs = 10kHz
277     //Ki = 9.8012
278     //Kp = 0.0893
279     //ReferenciaVpv = 20;
280
281     //ErroVpv = (ReferenciaVpv - TensionPV);
282     ErroVpv = (ReferenciaVpv - TensionPV) + AntiWindup/1000; //KantiWindUp = 1400*9.8012/1000
283
284     IntegratorPI = IntegratorPI + (ErroVpv + ErroVpv_1)/(2*10000); //Trapezio 10kHz
285
286     //ControlPIoutput = -(ErroVpv*0.0141 + IntegratorPI*5.8738)*1400 //Testado
287     ControlPIoutput = -(ErroVpv*0.0893 + IntegratorPI*9.8012)*1400;
288
289
290     //Saturador de 100(evitar queima do MOSFET baixo com bateria) a 1350
291     if ((ControlPIoutput >=100)&&(ControlPIoutput <=1350))
292     {
293         PDC1 = ControlPIoutput;

```

```
294         AntiWindup = 0;
295     }
296     else
297     {
298         if (ControlPloutput < 100)
299         {
300             PDC1 = 100;
301             AntiWindup = ControlPloutput - 100;
302         }
303
304         if (ControlPloutput > 1350)
305         {
306             PDC1 = 1350;
307             AntiWindup = ControlPloutput - 1350;
308         }
309     }
310
311     ErroVpv_1 = ErroVpv;
312
313 }
314
315
316 IFS0bits.T1IF = 0; // Clear Timer1 Interrupt Flag
317 }
318
319
320 void __attribute__((__interrupt__, no_auto_psv)) _T2Interrupt(void) //500Hz periodic
321 {
322
323     //LATBbits.LATB9 = i;
324     //i=~i;
325
326     /*
327     //Obter a FuncaoDeTransferencia////////////////////////////////////
328     if (enableFunctionTransfer==1)
329     {
330         SumLDRs = (Sensor_LDR[0][n] + Sensor_LDR[1][n] + Sensor_LDR[2][n]
331                 + Sensor_LDR[3][n] + Sensor_LDR[4][n])/5;
332
333         //Eixo X
334         //Degrau aplicado tempo 0 com amplitude = 0.2
335         //Ganho Kp aplicado no sistema em malha fechada = 20
336
337         //Erro normalizado pela media de todos os sensores
338         XaxisError = (Sensor_LDR[1][n] - Sensor_LDR[2][n])/SumLDRs;
339
340         //XaxisControl = 5*XaxisError*65535; //Normaliza a saida de 0 a 1
341         XaxisControl = 20*(0.2-XaxisError)*65535; //Normaliza a saida de 0 a 1
342
343         if (XaxisControl >= 0)
344         {
345             LATAbits.LATA10 = 1;
346             LATAbits.LATA7 = 0;
347         }
348
349         if (XaxisControl < 0)
350         {
351             LATAbits.LATA10 = 0;
352             LATAbits.LATA7 = 1;
353         }
354     }
355 }
```

```

354
355     if ((fabs(XaxisControl)>0)&&(fabs(XaxisControl)<65535)) PDC2 = fabs(XaxisControl);
356     if (fabs(XaxisControl)<=0) PDC2 = 0;
357     if (fabs(XaxisControl)>=65535) PDC2 = 65535;
358
359
360     //Eixo Y
361     //Degrau aplicado tempo 0 com amplitude = 0.2
362     //Ganho Kp aplicado no sistema em malha fechada = 20
363     //Erro normalizado pela media de todos os sensores
364     YaxisError = (Sensor_LDR[3][n] - Sensor_LDR[4][n])/SumLDRs;
365
366     YaxisControl = 20*(0.2-YaxisError)*65535; //Normaliza a saida de 0 a 1
367
368     if (YaxisControl >=0)
369     {
370         LATBbits.LATB11 = 1;
371         LATBbits.LATB13 = 0;
372     }
373
374     if (YaxisControl <0)
375     {
376         LATBbits.LATB11 = 0;
377         LATBbits.LATB13 = 1;
378     }
379
380     if ((fabs(YaxisControl)>0)&&(fabs(YaxisControl)<65535)) PDC3 = fabs(YaxisControl);
381     if (fabs(YaxisControl)<=0) PDC3 = 0;
382     if (fabs(YaxisControl)>=65535) PDC3 = 65535;
383
384 }
385 */
386
387 //Rastrear Solar efeito girassol////////////////////////////////////
388 //Controle Proporcional K
389
390 //enableRastreamentoDoSol = 1;
391
392 if (enableRastreamentoDoSol == 1)
393 {
394     //Controle Proporcional K
395     SumLDRs = (Sensor_LDR[0][n] + Sensor_LDR[1][n] + Sensor_LDR[2][n]
396             + Sensor_LDR[3][n] + Sensor_LDR[4][n])/5;
397
398     if (enableX==1)
399     {
400         //Axis X Control
401         //Erro normalizado pela media de todos os sensores
402         XaxisError = (Sensor_LDR[1][n] - Sensor_LDR[2][n])/SumLDRs;
403
404         //XaxisControl = -8.29*XaxisError*65535; //Normaliza a saida de 0 a 1
405         XaxisControl = -6.53*XaxisError*65535;
406
407         if (XaxisControl >=0)
408         {
409             LATAbits.LATA10 = 1;
410             LATAbits.LATA7 = 0;
411         }
412
413         if (XaxisControl <0)

```

```
414     {
415         LATAbits.LATA10 = 0;
416         LATAbits.LATA7 = 1;
417     }
418
419     if ((fabs(XaxisControl)>8000)&&(fabs(XaxisControl)<65535)) PDC2 = fabs(XaxisControl);
420     if (fabs(XaxisControl)<=8000) PDC2 = 8000;
421     if (fabs(XaxisControl)>=65535) PDC2 = 65535;
422 }
423
424 if (enableY==1)
425 {
426     //Axis Y Control
427     //Erro normalizado pela media de todos os sensores
428     YaxisError = (Sensor_LDR[3][n] - Sensor_LDR[4][n])/SumLDRs;
429
430     //YaxisControl = -8.03*YaxisError*65535; //Normaliza a saida de 0 a 1
431     YaxisControl = -8.01*YaxisError*65535;
432
433     if (YaxisControl >=0)
434     {
435         LATBbits.LATB11 = 1;
436         LATBbits.LATB13 = 0;
437     }
438
439     if (YaxisControl <0)
440     {
441         LATBbits.LATB11 = 0;
442         LATBbits.LATB13 = 1;
443     }
444
445     if ((fabs(YaxisControl)>4000)&&(fabs(YaxisControl)<65535)) PDC3 = fabs(YaxisControl);
446     if (fabs(YaxisControl)<=4000) PDC3 = 4000;
447     if (fabs(YaxisControl)>=65535) PDC3 = 65535;
448 }
449 }
450
451
452
453 //Controle por histerese mesclado ao classico////////////////////////////////////
454
455 //Eixo X
456 if ((FilterON_LDR[1][n]<FilterON_LDR[0][n])&&(moveX==0))
457 {
458     //LATAbits.LATA10 = 1;
459     //LATAbits.LATA7 = 0;
460     //PDC2 = 65535;
461     moveX = 1;
462     enableX = 1;
463 }
464
465 if ((FilterOFF_LDR[2][n]<FilterOFF_LDR[1][n])&&(moveX==1))
466 {
467     LATAbits.LATA10 = 0;
468     LATAbits.LATA7 = 0;
469     PDC2 = 0;
470     moveX = 0;
471     enableX = 0;
472 }
473
```

```
474     if (( FilterON_LDR [2][n]<FilterON_LDR [0][n])&&(moveX==0))
475     {
476         //LATAbits.LATA10 = 0;
477         //LATAbits.LATA7 = 1;
478         //PDC2 = 65535;
479         moveX = -1;
480         enableX = 1;
481     }
482
483     if (( FilterOFF_LDR [1][n]<FilterOFF_LDR [2][n])&&(moveX== -1))
484     {
485         LATAbits.LATA10 = 0;
486         LATAbits.LATA7 = 0;
487         PDC2 = 0;
488         moveX = 0;
489         enableX = 0;
490     }
491
492     //Eixo Y
493     if (( FilterON_LDR [3][n]<FilterON_LDR [0][n])&&(moveY==0))
494     {
495         //LATBbits.LATB11 = 1;
496         //LATBbits.LATB13 = 0;
497         //PDC3 = 65535;
498         moveY = 1;
499         enableY = 1;
500     }
501
502     if (( FilterOFF_LDR [4][n]<FilterOFF_LDR [3][n])&&(moveY==1))
503     {
504         LATBbits.LATB11 = 0;
505         LATBbits.LATB13 = 0;
506         PDC3 = 0;
507         moveY = 0;
508         enableY = 0;
509     }
510
511     if (( FilterON_LDR [4][n]<FilterON_LDR [0][n])&&(moveY==0))
512     {
513         //LATBbits.LATB11 = 0;
514         //LATBbits.LATB13 = 1;
515         //PDC3 = 65535;
516         moveY = -1;
517         enableY = 1;
518     }
519
520     if (( FilterOFF_LDR [3][n]<FilterOFF_LDR [4][n])&&(moveY== -1))
521     {
522         LATBbits.LATB11 = 0;
523         LATBbits.LATB13 = 0;
524         PDC3 = 0;
525         moveY = 0;
526         enableY = 0;
527     }
528
529
530     //Perpendicular solo ////////////////////////////////////////
531
532     if (perpenXRight == 1)
533     {
```

```
534     switch (PORTBbits.RB4)
535     {
536         case 0:
537             PDC2 = 65535;
538             LATAbits.LATA10 = 1;
539             LATAbits.LATA7 = 0;
540             break;
541
542         case 1:
543             PDC2 = 0;
544             LATAbits.LATA10 = 0;
545             LATAbits.LATA7 = 0;
546             perpenXRight = 0;
547             break;
548     }
549 }
550
551 if(perpenXLeft == 1)
552 {
553     switch (PORTBbits.RB4)
554     {
555         case 0:
556             PDC2 = 65535;
557             LATAbits.LATA10 = 0;
558             LATAbits.LATA7 = 1;
559             break;
560
561         case 1:
562             PDC2 = 0;
563             LATAbits.LATA10 = 0;
564             LATAbits.LATA7 = 0;
565             perpenXLeft = 0;
566             break;
567     }
568 }
569
570
571 if(perpenYRight == 1)
572 {
573     switch (PORTAbits.RA8)
574     {
575         case 0:
576             PDC3 = 65535;
577             LATBbits.LATB11 = 1;
578             LATBbits.LATB13 = 0;
579             break;
580
581         case 1:
582             PDC3 = 0;
583             LATBbits.LATB11 = 0;
584             LATBbits.LATB13 = 0;
585             perpenYRight = 0;
586             break;
587     }
588 }
589
590
591 if(perpenYLeft == 1)
592 {
593     switch (PORTAbits.RA8)
```



```
594     {
595         case 0:
596             PDC3 = 65535;
597             LATBbits.LATB11 = 0;
598             LATBbits.LATB13 = 1;
599             break;
600
601         case 1:
602             PDC3 = 0;
603             LATBbits.LATB11 = 0;
604             LATBbits.LATB13 = 0;
605             penpenYLeft = 0;
606             break;
607     }
608 }
609 }
610
611
612 //Rastreo Maximo Ponto Potencia MPPT////////////////////////////////////
613
614 if (EnableMPPT == 1)
615 {
616     //LATBbits.LATB9 = 1;
617     //DeltaVpv = 0.05V;
618
619     PowerPV = TensionPV*CurrentPV;
620
621     if(PowerPV - PowerPV_1 > 0)
622     {
623
624         if(TensionPV - TensionPV_1 > 0)
625         {
626             ReferenciaVpv = ReferenciaVpv + 0.05;
627         }
628
629         if(TensionPV - TensionPV_1 <= 0)
630         {
631             ReferenciaVpv = ReferenciaVpv - 0.05;
632         }
633     }
634 }
635
636 if(PowerPV - PowerPV_1 < 0)
637 {
638
639     if(TensionPV - TensionPV_1 >= 0)
640     {
641         ReferenciaVpv = ReferenciaVpv - 0.05;
642     }
643
644     if(TensionPV - TensionPV_1 < 0)
645     {
646         ReferenciaVpv = ReferenciaVpv + 0.05;
647     }
648 }
649 }
650
651 TensionPV_1 = TensionPV;
652 CurrentPV_1 = CurrentPV;
653 PowerPV_1 = PowerPV;
```

```
654
655     //Saturador entre 5V e 50V para P&O
656     if(ReferenciaVpv < 5) ReferenciaVpv = 5;
657     if(ReferenciaVpv > 50) ReferenciaVpv = 50;
658
659 }
660
661 //Carregamento da Bateria de Chumbo-Acido com P&O modificado////////////////
662
663 if (EnableCharging == 1)
664 {
665     //LATBbits.LATB9 = 1;
666     //DeltaVpv = 0.02V;
667     //Vmax = 14.5 V
668     //Imax = 3 A
669
670
671     PowerPV = TensionPV*CurrentPV;
672
673
674     if((PowerPV - PowerPV_1 > 0)&&(TensionLoad <= 14.5)&&(CurrentLoad <= 3))
675     {
676
677         if(TensionPV - TensionPV_1 > 0)
678         {
679             ReferenciaVpv = ReferenciaVpv + 0.02;
680         }
681
682         if(TensionPV - TensionPV_1 <= 0)
683         {
684             ReferenciaVpv = ReferenciaVpv - 0.02;
685         }
686     }
687
688
689     if((PowerPV - PowerPV_1 < 0)|| (TensionLoad > 14.5)|| (CurrentLoad > 3))
690     {
691
692         if(TensionPV - TensionPV_1 >= 0)
693         {
694             ReferenciaVpv = ReferenciaVpv - 0.02;
695         }
696
697         if(TensionPV - TensionPV_1 < 0)
698         {
699             ReferenciaVpv = ReferenciaVpv + 0.02;
700         }
701     }
702
703
704
705     TensionPV_1 = TensionPV;
706     CurrentPV_1 = CurrentPV;
707     PowerPV_1 = PowerPV;
708
709     //Saturador entre 15V e 50V para P&O
710     if(ReferenciaVpv < 15) ReferenciaVpv = 15;
711     if(ReferenciaVpv > 50) ReferenciaVpv = 50;
712
713 }
```

```
714
715
716     IFS0bits.T2IF = 0; // Clear Timer2 Interrupt Flag
717 }
718
719
720 void __attribute__((__interrupt__, no_auto_psv)) _U1RXInterrupt(void)
721 {
722     //U1TXREG = ReceivedChar;
723     if(U1STAbits.OERR == 1) U1STAbits.OERR = 0;
724
725     ReceivedChar = 0; //Clear Buffer
726     ReceivedChar = U1RXREG; //Receiver Char
727
728     switch (ReceivedChar){
729
730         //Envio variaveis internas do DsPic/////////////////////////////////
731         case '{':
732             data1A.TensionPainel = TensionPV;
733             for(i=0;i<4;i++) U1TXREG = data1A.SendChar[i];
734             break;
735
736         case '$':
737             data1B.ReferenciaVPainel = ReferenciaVpv;
738             for(i=0;i<4;i++) U1TXREG = data1B.SendChar[i];
739             break;
740
741         case '#':
742             data1C.ControlVPainelOut = ControlPloutput;
743             for(i=0;i<4;i++) U1TXREG = data1C.SendChar[i];
744             break;
745
746         case '}':
747             data2.CurrentPainel = CurrentPV;
748             for(i=0;i<4;i++) U1TXREG = data2.SendChar[i];
749             break;
750
751         case '!':
752             data3A.TensionLoad = TensionLoad;
753             for(i=0;i<4;i++) U1TXREG = data3A.SendChar[i];
754             break;
755
756         case '@':
757             data3B.CurrentLoad = CurrentLoad;
758             for(i=0;i<4;i++) U1TXREG = data3B.SendChar[i];
759             break;
760
761         case 'a':
762             data4.SensorX = XaxisError;
763             for(i=0;i<4;i++) U1TXREG = data4.SendChar[i];
764             break;
765
766         case 'b':
767             data5.SensorY = YaxisError;
768             for(i=0;i<4;i++) U1TXREG = data5.SendChar[i];
769             break;
770
771         case 'c':
772             data6.ControlX = XaxisControl;
773             for(i=0;i<4;i++) U1TXREG = data6.SendChar[i];
```

```
774         break;
775
776     case 'd':
777         data7.ControlY = YaxisControl;
778         for(i=0;i<4;i++) UITXREG = data7.SendChar[i];
779         break;
780
781     //Controle da acoes do sistema////////////////////////////////////
782     case 'm':
783         EnableMPPT = 1;
784         break;
785
786     case 'v':
787         ReferenciaVpv = 10; //Condicao Inicial
788         EnableControlVpv = 1;
789         break;
790
791     case 'r':
792         //LATBbits.LATB9 = 1;
793         enableRastreamentoDoSol = 1;
794         enableX=1;
795         enableY=1;
796         break;
797
798     case 's':
799         enableRastreamentoDoSol = 0;
800         enableX=0;
801         enableY=0;
802         PDC2 = 0;
803         PDC3 = 0;
804         break;
805
806     case 'p':
807         //LATBbits.LATB9 = 1;
808         perpenXRight = 1;
809         perpenXLeft = 0;
810         perpenYRight = 1;
811         penpenYLeft = 0;
812         break;
813
814     case 't':
815         perpenXRight = 0;
816         perpenXLeft = 1;
817         perpenYRight = 0;
818         penpenYLeft = 1;
819         break;
820
821     case 'f':
822         enableFunctionTransfer = 1;
823         break;
824
825     case 'g':
826         ReferenciaVpv = 14; //Condicao Inicial
827         EnableCharging = 1;
828         break;
829
830     //Teste LDRs
831     //case 'z':
832     //    data7.ControlY = Sensor_LDR[4][n];
833     //    for(i=0;i<4;i++) UITXREG = data7.SendChar[i];
```

```
834         // break;
835
836     }
837
838
839     IFS0bits.U1RXIF = 0;
840 }
841
842
843 int main(void) {
844
845     // Configure PLL prescaler , PLL postscaler , PLL divisor
846
847     CLKDIVbits.PLLPRE = 0;          // N1 = PLLPRE+2 = 2
848     PLLFBD = 33;                   // M = PLLDIV + 2 =35
849     CLKDIVbits.PLLPOST = 0;        // N2= 2 x (PLLPOST + 1) =2
850
851                                     //16M*35/2*2=140MHz
852
853     // Initiate Clock Switch to Primary Oscillator with PLL (NOSC = 0b011)
854     __builtin_write_OSCCONH(0x03);
855     __builtin_write_OSCCONL(0x01);
856
857     // Wait for Clock switch to occur
858     while (OSCCONbits.COSC != 0b011);
859
860     // Wait for PLL to lock
861     while(OSCCONbits.LOCK != 1);
862
863
864     // Initialization Code Timer1 Counter Mode
865
866     T1CONbits.TON = 0;              // Disable Timer
867     T1CONbits.TCS = 0;              // Select internal instruction cycle clock (Fosc/2)
868     T1CONbits.TCKPS = 0b00;        // Select 1:1 Prescaler
869     TMR1 = 0x00;                   // Clear timer register
870     //PR1 = 1399;                   // Load the period value T=PR1+1 freq= 50kHz
871     PR1 = 6999;                    // Load the period value T=PR1+1 freq= 10kHz
872     IPC0bits.T1IP = 5;             // Set Timer1 Interrupt Priority Level
873     IFS0bits.T1IF = 0;             // Clear Timer1 Interrupt Flag
874     IEC0bits.T1IE = 1;            // Enable Timer1 interrupt
875     T1CONbits.TON = 1;            // Start Timer
876
877
878     // Initialization Code Timer2 Counter Mode
879
880     T2CONbits.TON = 0;              // Disable Timer
881     T2CONbits.TCS = 0;              // Select internal instruction cycle clock (Fosc/2)
882     T2CONbits.TCKPS = 0b01;        // Select 1:8 Prescaler
883     //TMR2 = 0x00;                 // Clear timer register
884     //PR2 = 8749;                   // Load the period value T=PR1 +1 freq= 1kHz
885     PR2 = 17499;                   // Load the period value T=PR1 +1 freq= 500Hz
886     IPC1bits.T2IP = 6;             // Set Timer1 Interrupt Priority Level
887     IFS0bits.T2IF = 0;             // Clear Timer1 Interrupt Flag
888     IEC0bits.T2IE = 1;            // Enable Timer1 interrupt
889     T2CONbits.TON = 1;            // Start Timer
890
891
892     //Serial UART configure//////////////////////////////////////
893
```

```

894     TRISCbits.TRISC6 = 0; //Configure RP54 Output
895     TRISCbits.TRISC7 = 1; //Configure RP55 Input
896     ANSELCbits.ANSC7 = 0; //The ANSELx record has a default value of 0xFFFF
897
898     __builtin_write_OSCCONL(OSCCON & ~(1<<6)); // Unlock Registers
899     RPOR6bits.RP54R = 1; //Pin RP54 to the UART1 TX output registre
900     RPINR18bits.U1RXR = 55; //Pin RP55 to the UART1 RX input register
901     __builtin_write_OSCCONL(OSCCON | (1<<6)); // Lock Registers
902
903     //U1MODEbits.STSEL = 0; // 1–Stop bit
904     // U1MODEbits.PDSEL = 0; // No Parity , 8–Data bits
905     //U1MODEbits.ABAUD = 0; // Auto–Baud disabled
906     //U1MODEbits.BRGH = 0; // Standard–Speed mode
907
908     U1MODE = 0; //clear mode register
909     U1STA = 0; //clear status register
910
911     //U1BRG = 18; // (70M/230.4k/16)–1 BaudRate 230.4k
912     //U1MODEbits.BRGH = 0; // Standard–Speed mode (16x baud clock , Standard mode)
913     U1BRG = 18; // (70M/921.6k/4)–1 BaudRate 921.6k
914     U1MODEbits.BRGH = 1; // High–Speed mode (4x baud clock , High–Speed mode)
915
916     U1MODEbits.WAKE = 1; // Wake–up enabled
917     IEC0bits.U1TXIE = 0; // Disable UART TX interrupt
918     IPC3bits.U1TXIP = 0; //Set Send Interrupt Priority Level
919     IEC0bits.U1RXIE = 1; //Interrupt after one RX character is receive
920     IFS0bits.U1RXIF = 0; //clear the receive flag
921     IPC2bits.U1RXIP = 7; //Set Receive Interrupt Priority Level
922     U1MODEbits.UARTEN = 1; // Enable UART
923     U1STAbits.UTXEN = 1; // Enable UART TX
924
925
926     //Configure ADC////////////////////////////////////
927
928     //Voltage and Current
929     TRISAbits.TRISA0 = 1; //Defined RA0 input analog AN0
930     ANSELAbits.ANSA0 = 1;
931     TRISAbits.TRISA1 = 1; //Defined RA1 input analog AN1
932     ANSELAbits.ANSA1 = 1;
933     TRISBbits.TRISB0 = 1; //Defined RB0 input analog AN2
934     ANSELBbits.ANSB0 = 1;
935     TRISBbits.TRISB1 = 1; //Defined RB1 input analog AN3
936     ANSELBbits.ANSB1 = 1;
937
938     //LDRs acquisition
939     TRISBbits.TRISB2 = 1; //Defined RB2 input analog AN4 ->LDR0
940     ANSELBbits.ANSB2 = 1;
941     TRISBbits.TRISB3 = 1; //Defined RB3 input analog AN5 ->LDR1
942     ANSELBbits.ANSB3 = 1;
943     TRISCbits.TRISC0 = 1; //Defined RC0 input analog AN6 ->LDR2
944     ANSELCbits.ANSC0 = 1;
945     TRISCbits.TRISC1 = 1; //Defined RC1 input analog AN7 ->LDR3
946     ANSELCbits.ANSC1 = 1;
947     TRISCbits.TRISC2 = 1; //Defined RC2 input analog AN8 ->LDR4
948     ANSELCbits.ANSC2 = 1;
949     TRISAbits.TRISA4 = 1; //Defined RA4 input analog AN24
950     ANSELAbits.ANSA4 = 1;
951     TRISAbits.TRISA9 = 1; //Defined RA9 input analog AN28
952     ANSELAbits.ANSA9 = 1;
953     TRISCbits.TRISC3 = 1; //Defined RC3 input analog AN29

```

```

954 ANSELBits.ANSC3 = 1;
955 TRISBits.TRISC4 = 1; //Defined RC4 input analog AN30
956 ANSELBits.ANSC4 = 1;
957 TRISBits.TRISC5 = 1; //Defined RC5 input analog AN31
958 ANSELBits.ANSC5 = 1;
959
960 //Potentiometers X and Y
961 TRISBits.TRISB7 = 1; //Defined RB7 input analog AN25
962 ANSELBits.ANSB7 = 1;
963 TRISBits.TRISB8 = 1; //Defined RB8 input analog AN26
964 ANSELBits.ANSB8 = 1;
965
966
967
968 // Initialize ADC module
969 // Enable 12-bit mode, auto-sample and auto-conversion. Sampling begins when SAMP bit is set
970 AD1CON1 = 0x04E0;
971 // Sample 5 channels alternately using channel scanning
972 AD1CON2 = 0x0410;
973 // Tad = 8/70M = 114.3ns. Sample time = 3*TAD. Conversion 514.7 ksps (14+3)/Tad
974 AD1CON3 = 0x0307;
975 AD1CON4 = 0x0000;
976
977 AD1CSSH = 0x0000;
978 AD1CSSL = 0x001F; // Select AN0 to AN4 for input scan //INICIAL LDR sample
979
980 // Assign MUXA inputs
981 AD1CHS0bits.CH0SA = 0; // CH0SA bits ignored for CH0 +ve input selection
982 AD1CHS0bits.CH0NA = 0; // Select VREF- for CH0 -ve input
983
984 //ADC interrupt config
985 IPC3bits.AD1IP = 6; //Set priority ADC interrupt
986 IFS0bits.AD1IF = 0; //Clear adc Flag interrupt
987 IEC0bits.AD1IE = 1; //Enable ADC interrupt
988 AD1CON1bits.ADON = 1; //Enable ADC module
989
990
991 //Configure PWM////////////////////////////////////
992
993 //PWM Buck scrycrono Complementary
994 PHASE1 = 1400; //Fpwm=140M/1400 = 100kHz
995 //PDC1 = 700; //Set Duty Cycles
996 //PDC1 = 0; //Set Duty Cycles
997 PDC1 = 1350; //Evitar queima do MOSFET baixo com bateria (Inicia limite superior)
998 //DTR1 = 28; //Set Dead time 200ns;
999 //ALTDTR1 = 28; //Set Dead time 200ns;
1000 DTR1 = 14; //Set Dead time 100ns;
1001 ALTDTR1 = 14; //Set Dead time 100ns;
1002 IOCON1 = 0xC000; //Set PWM Mode to Complementary and active PWM1H, PWM1L pins
1003 // Set Independent Time Bases, Edge-Aligned Mode and Independent Duty Cycles
1004 PWMCON1 = 0x0200;
1005 FCLCON1 = 0x0003; //Disabel Faults Pins
1006
1007 //PWM Motor control Independent
1008
1009 PHASE2 = 65535; //Fpwm=140M/65535 = 2.14kHz
1010 PHASE3 = 65535;
1011 PDC2 = 0; //Set Duty Cycles
1012 PDC3 = 0;
1013 DTR2 = DTR3 = 0; //Set Dead Time Values

```

```
1014     ALTDTR2 = ALTDTR3 = 0;
1015     IOCON2 = IOCON3 = 0x8400; //Set PWM Mode to Redundant and PWMxL disable
1016     //Set Independent Time Bases, Edge-Aligned Mode and Independent Duty Cycles
1017     PWMCON2 = PWMCON3 = 0x0200;
1018     FCLCON2 = FCLCON3 = 0x0003; //Configure Faults
1019
1020     PTCON2 = 0x0000; //1:1 Prescaler
1021     PTCON = 0x8000; // Enable PWM Module
1022
1023
1024     //Output pots configure
1025
1026
1027     //Driver H Bridge X
1028     TRISAbits.TRISA10 = 0;
1029     LATAbits.LATA10 = 0;
1030
1031     TRISAbits.TRISA7 = 0;
1032     LATAbits.LATA7 = 0;
1033
1034     //Driver H Bridge Y
1035     TRISBbits.TRISB11 = 0;
1036     LATBbits.LATB11 = 0;
1037
1038     TRISBbits.TRISB13 = 0;
1039     LATBbits.LATB13 = 0;
1040
1041
1042
1043     //Center Magnect sensor input
1044     TRISBbits.TRISB4 = 1;
1045     TRISAbits.TRISA8 = 1;
1046
1047     //LED RB9 test
1048     TRISBbits.TRISB9 = 0;
1049     LATBbits.LATB9 = 0;
1050
1051     while(1)
1052     {
1053         //Infinite loop
1054     }
1055
1056     return 0;
1057
1058 }
```

APÊNDICE B – Algoritmo implementado no MATLAB para comunicação com dsPIC

```

1
2 serial = serial('COM4');
3 set(serial, 'BaudRate', 921600);
4 set(serial, 'InputBufferSize', 4);
5 set(serial, 'Timeout', 1);
6 fopen(serial);
7
8 %Enable ControleVpv%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 fprintf(serial, 'v');
10
11 %Enable MPPT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 fprintf(serial, 'm');
13
14 %Ensaiar funcao de transferencia%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 %fprintf(serial, 'f')
16
17 %Habilita P&O modificado p/ carregar a bateria%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18 fprintf(serial, 'g')
19
20 N=500;
21 for n=1:N
22
23     fprintf(serial, '{');
24     TensionPainel(n) = fread(serial, 1, 'float32');
25
26     fprintf(serial, '$');
27     ReferenciaVPainel(n) = fread(serial, 1, 'float32');
28
29     fprintf(serial, '#');
30     ControlVPainelOut(n) = fread(serial, 1, 'float32');
31
32     fprintf(serial, '}');
33     CurrentPainel(n) = fread(serial, 1, 'float32');
34
35     fprintf(serial, '!');
36     TensionLoad(n) = fread(serial, 1, 'float32');
37
38     fprintf(serial, '@');
39     CurrentLoad(n) = fread(serial, 1, 'float32');
40
41
42     fprintf(serial, 'a');
43     SensorX(n) = fread(serial, 1, 'float32');
44
45     fprintf(serial, 'b');
46     SensorY(n) = fread(serial, 1, 'float32');
47
48     fprintf(serial, 'c');
49     ControlX(n) = fread(serial, 1, 'float32');
50
51     fprintf(serial, 'd');
52     ControlY(n) = fread(serial, 1, 'float32');
53

```

```
54     pause(0.01); %fs = 100Hz
55
56 end
57
58 tempo1 = 0:0.01:N*0.01-0.01; %fs = 100Hz
59
60
61 %Rastreia sol%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
62 fprintf(serial, 'r');
63
64 %Stop rastreamento sol
65 fprintf(serial, 's'); %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
66
67 %Posicao perpendicular %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68 fprintf(serial, 't');
69 pause (8);
70 fprintf(serial, 'p');
71
72
73 %Close serial port
74 fclose(serial);
```
