



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

CAIO JORDAN OLIVEIRA DE SIQUEIRA

**ANÁLISE DO COMPORTAMENTO DINÂMICO DE UM MOTOR DE CORRENTE
CONTÍNUA POR MEIO DE TÉCNICAS BASEADAS NA TEORIA DO CAOS
APLICADAS A SINAIS DE CORRENTE ELÉTRICA**

QUIXADÁ

2026

CAIO JORDAN OLIVEIRA DE SIQUEIRA

ANÁLISE DO COMPORTAMENTO DINÂMICO DE UM MOTOR DE CORRENTE
CONTÍNUA POR MEIO DE TÉCNICAS BASEADAS NA TEORIA DO CAOS APLICADAS
A SINAIS DE CORRENTE ELÉTRICA

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Com-
putação do Campus Quixadá da Universidade
Federal do Ceará, como requisito parcial à
obtenção do grau de bacharel em Engenharia de
Computação.

Orientador: Prof. Dr. Thiago Werley
Bandeira da Silva.

QUIXADÁ

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- S629a Siqueira, Caio Jordan Oliveira de.
Análise do comportamento dinâmico de um motor de corrente contínua por meio de técnicas baseadas na teoria do caos aplicadas a sinais de corrente elétrica / Caio Jordan Oliveira de Siqueira. – 2026.
82 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Computação, Quixadá, 2026.
Orientação: Prof. Dr. Thiago Werlley Bandeira da Silva.
1. Motor DC. 2. Corrente Elétrica. 3. Teoria do Caos. 4. Análise Dinâmica. 5. Sistema Embarcado. I.
Título.

CDD 621.39

CAIO JORDAN OLIVEIRA DE SIQUEIRA

ANÁLISE DO COMPORTAMENTO DINÂMICO DE UM MOTOR DE CORRENTE
CONTÍNUA POR MEIO DE TÉCNICAS BASEADAS NA TEORIA DO CAOS APLICADAS
A SINAIS DE CORRENTE ELÉTRICA

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Com-
putação do Campus Quixadá da Universidade
Federal do Ceará, como requisito parcial à
obtenção do grau de bacharel em Engenharia de
Computação.

Aprovada em: 26/01/2026.

BANCA EXAMINADORA

Prof. Dr. Thiago Werlley Bandeira da
Silva (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Antonio Joel Ramiro de Castro
Universidade Federal do Ceará (UFC)

Prof. Ms. Francisco Evando Nascimento dos Santos
Secretaria da Educação (SEDUC)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

AGRADECIMENTOS

À Universidade Federal do Ceará (UFC), por proporcionar a estrutura, o ambiente acadêmico e as oportunidades necessárias para minha formação profissional e pessoal.

Ao Prof. Dr. Thiago Werley Bandeira da Silva, pela orientação, disponibilidade e contribuições ao longo do desenvolvimento deste trabalho, fundamentais para a condução e consolidação desta pesquisa.

A todos os professores que contribuíram ao longo da minha trajetória acadêmica, pelo conhecimento compartilhado, pela orientação, incentivo e dedicação ao ensino, essenciais para o meu desenvolvimento durante o curso.

Aos colegas de graduação, com os quais tive a oportunidade de aprender e também de ensinar, pelas trocas de experiências, apoio mútuo e convivência ao longo dessa caminhada acadêmica.

Aos meus pais, pelo apoio incondicional, confiança e constantes sacrifícios realizados ao longo de toda a minha trajetória, sempre priorizando minha educação e possibilitando as melhores condições para meu desenvolvimento.

“A persistência é o caminho do êxito.”
(Chaplin)

RESUMO

O monitoramento do estado de funcionamento de motores elétricos é fundamental para garantir confiabilidade e eficiência em sistemas eletromecânicos. Entre as abordagens existentes, o uso da corrente elétrica como variável observável destaca-se por permitir a análise indireta do comportamento do sistema, sem a necessidade de sensores mecânicos adicionais. No entanto, métodos tradicionais baseados apenas nos domínios do tempo e da frequência apresentam limitações quando aplicados a sistemas que exibem dinâmicas não lineares e variações transitórias. Nesse contexto, este trabalho tem como objetivo coletar sinais de corrente elétrica de um motor de corrente contínua, em seguida, analisar o sinal utilizando a técnica baseada na teoria do caos, visando caracterizar o comportamento dinâmico do sistema sob diferentes condições de operação. A análise utilizou-se de método que consiste na aquisição experimental dos sinais de corrente por meio de um sensor do tipo ACS712 integrado a um sistema de aquisição embarcado baseado no microcontrolador ESP32. Os experimentos consideram diferentes regimes de funcionamento, incluindo operação normal, aplicação de carga mecânica contínua, carga mecânica intermitente e estado desligado. Os sinais adquiridos foram submetidos a etapas de pré-processamento, segmentação temporal e reconstrução do espaço de fases. A partir dessa reconstrução, métricas baseadas em análise de recorrência e dinâmica não linear foram extraídas e comparadas a uma baseline de funcionamento normal, permitindo a avaliação de desvios no comportamento dinâmico do sistema. Os resultados indicam que a corrente elétrica contém informações relevantes sobre o regime de funcionamento do motor, sendo possível observar diferenças consistentes entre as condições analisadas. Embora o desempenho do método apresente limitações inerentes ao caráter exploratório e à instrumentação de baixo custo utilizada, os resultados evidenciam o potencial das técnicas baseadas na teoria do caos como abordagem complementar às análises tradicionais no monitoramento de motores de corrente contínua.

Palavras-chave: motor de corrente contínua; corrente elétrica; teoria do caos; análise dinâmica; sistema embarcado.

ABSTRACT

The monitoring of the operating condition of electric motors is fundamental to ensure reliability and efficiency in electromechanical systems. Among the existing approaches, the use of electric current as an observable variable stands out for enabling indirect analysis of the system's behavior without the need for additional mechanical sensors. However, traditional methods based solely on the time and frequency domains present limitations when applied to systems that exhibit nonlinear dynamics and transient variations. In this context, this work aims to collect electric current signals from a direct current motor and then analyze the signal using a technique based on chaos theory, with the objective of characterizing the dynamic behavior of the system under different operating conditions. The analysis employed a method consisting of the experimental acquisition of current signals by means of an ACS712-type sensor integrated into an embedded acquisition system based on the ESP32 microcontroller. The experiments consider different operating regimes, including normal operation, application of continuous mechanical load, intermittent mechanical load, and the off state. The acquired signals were subjected to preprocessing, temporal segmentation, and phase space reconstruction steps. From this reconstruction, metrics based on recurrence analysis and nonlinear dynamics were extracted and compared to a baseline of normal operation, allowing the evaluation of deviations in the dynamic behavior of the system. The results indicate that the electric current contains relevant information about the motor's operating regime, making it possible to observe consistent differences among the analyzed conditions. Although the performance of the method presents limitations inherent to its exploratory nature and the use of low-cost instrumentation, the results highlight the potential of techniques based on chaos theory as a complementary approach to traditional analyses in the monitoring of direct current motors.

Keywords: direct current motor; electric current; chaos theory; dynamic analysis; embedded system.

LISTA DE FIGURAS

Figura 1 – Motor de corrente contínua com caixa de redução.	18
Figura 2 – Representação conceitual da reconstrução do espaço de fases a partir de uma série temporal, evidenciando a estrutura dinâmica do sistema.	26
Figura 3 – Módulo sensor de corrente baseado no efeito Hall (ACS712).	28
Figura 4 – Placa de desenvolvimento baseada no microcontrolador ESP32.	29
Figura 5 – Arquitetura geral do sistema experimental, incluindo circuito de potência, medição, aquisição via ESP32 e processamento em computador.	34
Figura 6 – Circuito de potência do sistema experimental, destacando a ligação em série entre bateria, sensor ACS712 e motor Corrente contínua (CC).	34
Figura 7 – Filtro RC e desacoplamento da alimentação aplicados ao sensor ACS712 para redução de ruído e estabilização do sinal.	35
Figura 8 – Fluxo de aquisição, armazenamento e processamento dos sinais de corrente elétrica.	37

LISTA DE TABELAS

Tabela 1 – Parâmetros da reconstrução do espaço de fases por condição de operação, calculados conforme as Equações (2.6) e (2.7).	41
Tabela 2 – Valores médios das métricas dinâmicas por condição de operação.	41
Tabela 3 – Resultados da avaliação automática por aquisição.	44
Tabela 4 – Resumo estatístico dos escores por condição de operação.	44

LISTA DE QUADROS

Quadro 1 – Análise comparativa entre os trabalhos relacionados e o presente estudo. . .	32
---	----

LISTA DE ABREVIATURAS E SIGLAS

ACS712	Sensor de corrente baseado no efeito Hall
CC	Corrente contínua
ESP32	Microcontrolador de 32 bits da Espressif Systems
FFT	<i>Fast Fourier Transform</i> / Transformada Rápida de Fourier

LISTA DE SÍMBOLOS

θ	Defasagem entre tensão e corrente
f	Frequência
t	Tempo
$x[n]$	Sinal discreto no tempo
$X[k]$	Coeficientes espectrais do sinal
S_{med}	Mediana do escore global de desvio dinâmico
S_{mean}	Média do escore global de desvio dinâmico
S_{max}	Valor máximo do escore global de desvio dinâmico
τ	Atraso temporal utilizado na reconstrução do espaço de fases
m	Dimensão de <i>embedding</i>
λ	Expoente de Lyapunov
e	Base do logaritmo natural
j	Unidade imaginária

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo Geral	16
1.1.1	<i>Objetivos Específicos</i>	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Motores de corrente contínua com caixa de redução	17
2.2	Medição por corrente elétrica	20
2.3	Aquisição e processamento de sinais	21
2.4	Análise no domínio da frequência	23
2.5	Teoria do Caos aplicada à análise de sinais	24
2.6	Sistema de aquisição embarcado	26
2.6.1	<i>Sensor de corrente por efeito Hall</i>	27
2.6.2	<i>Microcontrolador ESP32 como plataforma de aquisição</i>	27
2.6.3	<i>Conversão analógico-digital e limitações práticas</i>	28
3	TRABALHOS RELACIONADOS	30
3.1	Técnica baseada em teoria do caos para análise dinâmica de sistemas (Ramon, 2019)	30
3.2	Aplicação da teoria do caos no diagnóstico de falhas em máquinas rotati- vas (Li <i>et al.</i>, 2018)	31
3.3	Análise dinâmica não linear aplicada a sinais de vibração (Peng <i>et al.</i>, 2012)	31
3.4	Análise Comparativa	32
4	METODOLOGIA	33
4.1	Materiais, fonte de alimentação e arquitetura do sistema	33
4.2	Montagem experimental e circuito elétrico	33
4.3	Condicionamento analógico do sinal	35
4.4	Condições de operação do motor	35
4.5	Aquisição dos sinais de corrente elétrica	36
4.6	Pipeline computacional e baseline	36
4.6.1	<i>Pré-processamento do sinal e métricas no domínio do tempo</i>	37
4.6.2	<i>Reconstrução do espaço de fases</i>	38
4.6.3	<i>Extração de métricas não lineares</i>	38

4.6.4	<i>Escore global de desvio dinâmico</i>	39
4.6.5	<i>Relação com os resultados experimentais</i>	39
5	RESULTADOS	40
5.1	Visão geral das aquisições	40
5.2	Parâmetros da reconstrução do espaço de fases	41
5.3	Métricas dinâmicas por condição de operação	41
5.4	Resultados por condição de operação	42
5.4.1	<i>Funcionamento normal</i>	42
5.4.2	<i>Funcionamento sob carga mecânica contínua</i>	42
5.4.3	<i>Estado desligado</i>	43
5.4.4	<i>Funcionamento com carga mecânica intermitente</i>	43
5.5	Resultados quantitativos	43
6	CONCLUSÃO E TRABALHOS FUTUROS	45
	REFERÊNCIAS	48
	APÊNDICE A – Código-fonte do pipeline de análise caótica	50
	APÊNDICE B – Código-fonte do firmware de aquisição (ESP32)	75

1 INTRODUÇÃO

O uso de sistemas eletromecânicos tem crescido de forma significativa em diferentes áreas da engenharia, incluindo automação industrial, robótica e sistemas embarcados. Nesses contextos, motores elétricos desempenham papel fundamental, sendo responsáveis pela conversão de energia elétrica em energia mecânica necessária ao funcionamento dos sistemas. A confiabilidade desses equipamentos é um fator crítico, uma vez que falhas podem resultar em interrupções de operação, redução de desempenho e aumento de custos de manutenção. Estratégias adequadas de monitoramento e análise do comportamento dos sistemas são essenciais para garantir maior disponibilidade e eficiência operacional ao longo do tempo (Jardine *et al.*, 2013).

A necessidade de monitoramento contínuo de motores elétricos tem motivado o desenvolvimento de diferentes técnicas voltadas à detecção precoce de falhas e à avaliação das condições de operação desses sistemas. Métodos tradicionais baseados em inspeções diretas ou no uso de sensores mecânicos adicionais nem sempre são viáveis, especialmente em aplicações experimentais, educacionais ou de baixo custo, nas quais simplicidade, flexibilidade e viabilidade prática são fatores determinantes. Nesse contexto, abordagens indiretas baseadas na análise de sinais elétricos têm se destacado como alternativas eficientes para o monitoramento de motores. Estudos recentes evidenciam o uso crescente de sinais elétricos para diagnóstico de falhas (Verma *et al.*, 2021), enquanto trabalhos clássicos demonstram que a corrente elétrica pode refletir alterações no comportamento mecânico e operacional do motor, reforçando sua viabilidade como variável observável (Thorsen; Dalva, 1995).

Embora técnicas baseadas na análise no domínio do tempo e da frequência sejam amplamente utilizadas no monitoramento de motores elétricos devido à sua eficiência computacional, maturidade e robustez frente a ruídos, essas abordagens apresentam limitações quando aplicadas a sistemas reais, nos quais efeitos não lineares e variações dinâmicas estão presentes. Métodos tradicionais frequentemente assumem comportamentos estacionários e linearidade do sistema, o que pode dificultar a identificação de alterações sutis no regime de funcionamento do motor. Além disso, transformações para o domínio da frequência tendem a suprimir informações temporais relevantes, reduzindo a capacidade de caracterizar fenômenos transitórios ou dinâmicas complexas. Essas limitações têm motivado a investigação de abordagens alternativas capazes de capturar de forma mais completa o comportamento dinâmico de sistemas eletromecânicos, inclusive quando os sinais são adquiridos por sistemas de instrumentação simplificados (Lei *et*

al., 2014).

As limitações observadas nas abordagens tradicionais para a análise de sistemas eletromecânicos motivam a investigação de métodos capazes de considerar a dinâmica não linear presente nesses sistemas. Nesse cenário, técnicas baseadas na teoria do caos têm sido utilizadas para a análise de séries temporais, permitindo caracterizar o comportamento dinâmico de um sistema a partir da estrutura do sinal observado. Sistemas determinísticos não lineares podem exibir comportamentos complexos que não são adequadamente descritos por análises puramente estatísticas ou espectrais (Strogatz, 2015). Além disso, estudos têm explorado o uso de medidas e representações associadas à teoria do caos em tarefas de diagnóstico de falhas em máquinas rotativas, reforçando o potencial dessas técnicas no contexto de monitoramento (Li *et al.*, 2018).

Dessa forma, a análise da corrente elétrica sob uma perspectiva dinâmica configura-se como uma abordagem complementar para investigar variações no regime de funcionamento de motores elétricos. Neste trabalho, essa abordagem é explorada em um contexto experimental e exploratório, utilizando um sistema de aquisição embarcado de baixo custo para a obtenção dos sinais de corrente elétrica. O foco reside na avaliação da viabilidade da aplicação de técnicas baseadas na teoria do caos para a caracterização do comportamento dinâmico de um motor de corrente contínua, sem a pretensão de validação industrial, mas com ênfase em aplicações educacionais e experimentais.

1.1 Objetivo Geral

O presente trabalho tem como objetivo geral analisar sinais de corrente elétrica de um motor de corrente contínua por meio de técnicas baseadas na teoria do caos, visando caracterizar o comportamento dinâmico do sistema sob diferentes condições de operação.

1.1.1 Objetivos Específicos

- realizar a aquisição de sinais de corrente elétrica do motor em diferentes regimes de funcionamento;
- analisar a relação entre a corrente elétrica medida e o comportamento dinâmico do motor;
- aplicar técnicas de análise de séries temporais baseadas na teoria do caos;
- avaliar o potencial dessas técnicas como abordagem complementar às análises tradicionais no domínio do tempo e da frequência, em um contexto experimental e exploratório.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os conceitos teóricos necessários para a compreensão do desenvolvimento deste trabalho. Inicialmente, são abordados aspectos básicos relacionados ao funcionamento de motores elétricos e à relação entre corrente elétrica e esforço mecânico, fundamentais para o entendimento do comportamento do sistema analisado. Em seguida, são discutidos conceitos referentes ao monitoramento de sinais elétricos, com ênfase na análise da corrente como variável representativa das condições de operação do motor. Por fim, são apresentados fundamentos de processamento de sinais, incluindo técnicas de análise no domínio do tempo e da frequência, que servem de base para as análises realizadas ao longo do trabalho.

2.1 Motores de corrente contínua com caixa de redução

Motores de corrente contínua com caixa de redução são amplamente utilizados em aplicações de robótica educacional e sistemas eletromecânicos de pequeno porte, devido à sua simplicidade construtiva, baixo custo e facilidade de controle. Esse tipo de motor é frequentemente empregado em plataformas móveis, robôs seguidores de linha e projetos didáticos, pois permite a obtenção de torque adequado em baixas velocidades, característica essencial para aplicações que exigem precisão e estabilidade de movimento. Além disso, sua operação em baixa tensão e a compatibilidade com controladores simples, como pontes H e microcontroladores, tornam esse motor uma escolha recorrente em ambientes de ensino e prototipagem, sendo considerado uma boa prática para a introdução de conceitos relacionados ao acionamento e controle de motores elétricos.

O motor de corrente contínua utilizado neste trabalho é composto por um estator, responsável pela geração do campo magnético, e um rotor, no qual se encontram as bobinas que produzem o movimento rotacional quando alimentadas eletricamente, conforme ilustrado na Figura 1. A comutação da corrente elétrica é realizada por meio de um comutador em conjunto com escovas, que promovem a inversão periódica da corrente nas bobinas do rotor ao longo da rotação, assegurando a continuidade do movimento. De acordo com Chapman (Chapman, 2011), esse princípio de funcionamento resulta na conversão direta de energia elétrica em energia mecânica. Do ponto de vista físico, o torque desenvolvido por motores de corrente contínua é diretamente proporcional à corrente elétrica que percorre suas bobinas, enquanto a velocidade de

Figura 1 – Motor de corrente contínua com caixa de redução.



Fonte: Adaptado de (RoboMix, 2024).

rotação está associada à força contraeletromotriz gerada durante o funcionamento do motor.

Essa relação fundamental entre torque e corrente pode ser expressa de forma simplificada pela Equação (2.1), na qual T representa o torque eletromagnético desenvolvido pelo motor, I a corrente elétrica aplicada ao rotor e k_t a constante de torque do motor.

$$T = k_t \cdot I \quad (2.1)$$

A dinâmica elétrica do motor de corrente contínua também pode ser descrita, de forma simplificada, pela equação de tensão do circuito do rotor, apresentada na Equação (2.2), em que V é a tensão aplicada, R a resistência elétrica, L a indutância do enrolamento, I a corrente elétrica e $k_e \omega$ o termo associado à força contraeletromotriz, proporcional à velocidade angular do eixo.

$$V = RI + L \frac{dI}{dt} + k_e \omega \quad (2.2)$$

A caixa de redução mecânica acoplada ao motor de corrente contínua é constituída por um conjunto de engrenagens responsável por modificar as características de rotação do eixo, reduzindo a velocidade angular e ampliando o torque disponível na saída. Esse mecanismo

permite que motores de alta rotação sejam empregados em aplicações que exigem maior força e melhor controle de movimento, como sistemas de locomoção em robótica educacional. Contudo, a presença da caixa de redução introduz efeitos mecânicos adicionais ao sistema, como atrito entre engrenagens, folgas mecânicas e variações no esforço transmitido ao eixo, os quais tornam o comportamento do conjunto motor–carga mais complexo. Como consequência, essas interações mecânicas influenciam diretamente o esforço requerido do motor durante a operação, refletindo-se em variações no comportamento dinâmico da corrente elétrica consumida.

Do ponto de vista eletromecânico, o torque desenvolvido por motores de corrente contínua está diretamente relacionado à corrente elétrica que percorre as bobinas do rotor, uma vez que a força responsável pela geração do movimento resulta da interação entre o campo magnético do estator e a corrente aplicada. De acordo com Krishnan (Krishnan, 2001), essa relação constitui a base da modelagem de motores DC em sistemas de acionamento, sendo amplamente utilizada para representar o comportamento do torque em função da corrente. Assim, variações no esforço mecânico exigido pelo sistema, como aumentos de carga ou resistência ao movimento, demandam maior torque no eixo, o que se reflete em um aumento do consumo de corrente elétrica, característica também discutida por Boldea (Boldea, 2016) no contexto de acionamentos elétricos modernos.

Em motores de corrente contínua com caixa de redução, falhas e degradações mecânicas tendem a se manifestar de forma progressiva, alterando gradualmente as condições de operação do sistema. Fenômenos como aumento de atrito no eixo, desgaste ou folgas nas engrenagens, desalinhamentos mecânicos e irregularidades no contato das escovas introduzem variações no torque requerido ao longo do tempo, mesmo quando o motor continua operando aparentemente de forma normal. Essas irregularidades resultam em oscilações no esforço mecânico aplicado ao motor, que se refletem diretamente na forma de onda da corrente elétrica consumida, produzindo variações dinâmicas que vão além de simples alterações no valor médio. Trabalhos recentes demonstram que tais oscilações de torque podem ser observadas e analisadas a partir do sinal de corrente, evidenciando que a corrente elétrica carrega informações relevantes sobre irregularidades mecânicas e dinâmicas do sistema, como discutido em estudos sobre *torque ripple* e comutação em motores elétricos.

2.2 Medição por corrente elétrica

A corrente elétrica constitui uma das grandezas mais acessíveis para a observação do estado de operação de motores elétricos, podendo ser monitorada de forma contínua durante o funcionamento do sistema, sem a necessidade de acesso direto aos componentes mecânicos. Em sistemas de acionamento, a corrente está diretamente associada ao esforço exigido do motor, variando conforme as condições de carga e o regime de operação. De acordo com Chapman (Chapman, 2011), o torque desenvolvido por motores de corrente contínua é proporcional à corrente que percorre suas bobinas, o que estabelece uma relação direta entre o esforço mecânico requerido e o consumo de corrente elétrica.

Alterações na carga mecânica, como aumento do atrito, variações no esforço exigido ou aplicação de cargas externas, refletem-se em mudanças no comportamento da corrente elétrica ao longo do tempo. Quando o motor é submetido a maiores esforços, observa-se um aumento no valor da corrente consumida, enquanto condições de operação mais leves resultam em menores níveis de corrente. Essa relação é amplamente explorada na modelagem e análise de sistemas de acionamento elétrico, conforme discutido por Krishnan (Krishnan, 2001) e Boldea (Boldea, 2016), que destacam a corrente como uma variável fundamental para a avaliação do desempenho e das condições operacionais de motores elétricos.

De forma geral, o sinal de corrente elétrica observado pode ser compreendido como uma variável dependente do estado dinâmico do sistema eletromecânico, sendo influenciado por grandezas como o torque de carga, a velocidade de rotação e as condições de alimentação elétrica. Essa dependência pode ser expressa de maneira conceitual pela Equação (2.3), na qual a corrente elétrica $I(t)$ é representada como uma função do esforço mecânico aplicado, da velocidade angular do eixo e da tensão de alimentação.

$$I(t) = f(T_{\text{carga}}(t), \omega(t), V(t)) \quad (2.3)$$

Além das variações no valor médio da corrente, mudanças no regime de funcionamento do motor podem introduzir oscilações e irregularidades na forma de onda do sinal, associadas à dinâmica do sistema eletromecânico. Em motores com transmissão mecânica, como aqueles equipados com caixa de redução, efeitos como atrito, folgas mecânicas e irregularidades no contato das escovas tendem a produzir variações temporais na corrente elétrica consumida. Hughes e Drury (Hughes; Drury, 2019) ressaltam que essas flutuações carregam informações

relevantes sobre o comportamento dinâmico do motor, indo além de simples alterações de amplitude.

O monitoramento da corrente elétrica tem sido amplamente adotado em estratégias de diagnóstico e avaliação de condições de funcionamento de motores, sobretudo por permitir a análise do sistema sem a necessidade de sensores mecânicos adicionais. Trabalhos clássicos, como o de Thorsen e Dalva (Thorsen; Dalva, 1995), demonstram que a corrente elétrica pode ser utilizada como variável indireta para o monitoramento de motores, refletindo alterações no regime de operação e no esforço mecânico aplicado ao sistema.

Entretanto, é importante destacar que a corrente elétrica observada experimentalmente não representa uma grandeza ideal, mas sim um sinal influenciado pelas características do sistema de medição empregado. Em aplicações práticas, especialmente em sistemas experimentais e de baixo custo, fatores como ruído elétrico, *offset* de medição, limitações do sensor e do conversor analógico-digital podem afetar a forma de onda observada. Dessa forma, a análise da corrente elétrica deve considerar não apenas seus valores médios ou eficazes, mas também sua evolução temporal e suas características dinâmicas, motivando a adoção de abordagens capazes de extrair informações relevantes mesmo diante dessas limitações instrumentais.

2.3 Aquisição e processamento de sinais

A aquisição de sinais elétricos consiste no processo de obtenção de grandezas físicas contínuas, como a corrente elétrica, e sua conversão em sinais adequados para análise computacional. Em sistemas eletromecânicos, essa etapa é fundamental, pois estabelece a ligação entre o comportamento do sistema físico observado e os dados utilizados nas análises posteriores. Uma aquisição inadequada pode introduzir distorções ou perdas de informação que comprometem a interpretação do sinal e, consequentemente, as conclusões obtidas a partir dele. Conforme discutido por Oppenheim e Schafer (Oppenheim; Schafer, 2010), a qualidade do sinal adquirido influencia diretamente a confiabilidade das análises realizadas a partir de representações digitais.

O processo de aquisição envolve a amostragem do sinal contínuo e sua representação em formato discreto, resultando em uma série temporal que descreve a evolução da grandeza medida ao longo do tempo. Essa série temporal permite a observação de variações, padrões e irregularidades associadas ao comportamento do sistema monitorado. De acordo com Proakis e Manolakis (Proakis; Manolakis, 2007), a amostragem possibilita a análise de sinais físicos em

ambientes digitais, desde que realizada de forma adequada em relação às características do sinal e às limitações do sistema de aquisição.

Em aplicações práticas, especialmente em sistemas embarcados e de baixo custo, o processo de aquisição está sujeito a limitações inerentes ao hardware empregado. Conversores analógico-digital integrados a microcontroladores apresentam resolução finita, ruído eletrônico e variações temporais no intervalo de amostragem, o que pode resultar em pequenas flutuações na frequência de amostragem efetiva e na presença de offset no sinal digitalizado. Esses efeitos não inviabilizam a análise do sinal, mas exigem uma abordagem cuidadosa no tratamento dos dados adquiridos, considerando as características reais do sistema de medição.

Antes da aplicação de técnicas de análise, é comum realizar etapas de pré-processamento com o objetivo de tornar mais evidentes as informações relevantes contidas no sinal. Procedimentos como a remoção de offset, normalização e filtragem simples contribuem para reduzir efeitos indesejados e facilitar a interpretação do comportamento do sinal ao longo do tempo. Conforme destacado por Oppenheim e Schafer (Oppenheim; Schafer, 2010), essas etapas não têm como finalidade alterar a dinâmica do sistema observado, mas sim preparar o sinal para análises posteriores de forma mais consistente e confiável.

No contexto de análises baseadas em séries temporais e dinâmicas não lineares, o pré-processamento assume papel ainda mais relevante, uma vez que imperfeições na aquisição podem influenciar a geometria do espaço de fases reconstruído e, conseqüentemente, as métricas extraídas da série temporal. Dessa forma, a adoção de estratégias de normalização e segmentação temporal permite que a análise se concentre nas variações dinâmicas relativas do sinal, reduzindo a dependência de valores absolutos e tornando o método mais robusto frente às limitações instrumentais do sistema de aquisição.

Assim, a compreensão das etapas de aquisição, amostragem e pré-processamento é fundamental para garantir que os sinais analisados representem adequadamente o comportamento do sistema físico observado. Em sistemas eletromecânicos nos quais a dinâmica pode variar ao longo do tempo, uma abordagem cuidadosa no tratamento dos sinais estabelece a base necessária para a aplicação de métodos de análise no domínio da frequência e de técnicas voltadas ao estudo de comportamentos dinâmicos mais complexos, discutidos nas seções subsequentes.

2.4 Análise no domínio da frequência

A análise de sinais no domínio da frequência é amplamente utilizada no estudo de sistemas eletromecânicos, especialmente para a identificação de componentes periódicas e harmônicas associadas ao funcionamento de máquinas rotativas. Por meio de transformações matemáticas, como a Transformada de Fourier, é possível representar um sinal temporal como uma combinação de componentes senoidais de diferentes frequências, amplitudes e fases. Essa abordagem permite identificar padrões espectrais característicos e tem sido amplamente empregada em aplicações de monitoramento e diagnóstico de falhas (Bracewell, 2000).

Em particular, a Transformada Rápida de Fourier (*Fast Fourier Transform* / Transformada Rápida de Fourier (FFT)) destaca-se como uma ferramenta eficiente para a análise espectral de sinais discretos, possibilitando a identificação de frequências dominantes associadas a fenômenos físicos do sistema analisado. De acordo com Oppenheim e Schaffer (Oppenheim; Schaffer, 2010), a análise espectral é especialmente útil para a caracterização de sinais estacionários ou aproximadamente estacionários, nos quais as propriedades estatísticas do sinal não variam significativamente ao longo do tempo.

Apesar de sua ampla aplicação, a análise no domínio da frequência apresenta limitações quando aplicada a sistemas que exibem comportamento dinâmico não linear ou variações transitórias ao longo do tempo. Em tais situações, informações relevantes podem ser distribuídas de forma não uniforme no sinal temporal, dificultando sua representação adequada por meio de espectros globais de frequência. Além disso, a transformação para o domínio da frequência tende a suprimir a localização temporal de eventos, o que pode comprometer a análise de fenômenos de curta duração ou de caráter não estacionário (Proakis; Manolakis, 2007).

Dessa forma, embora a análise espectral constitua uma ferramenta consolidada e fundamental no processamento de sinais, suas limitações motivam a investigação de abordagens complementares capazes de considerar simultaneamente a evolução temporal e a dinâmica do sistema observado. Neste trabalho, a análise no domínio da frequência é apresentada como referencial conceitual, servindo de base para contextualizar a escolha de métodos alternativos voltados à análise dinâmica e não linear de sinais de corrente elétrica, discutidos nas seções subsequentes.

2.5 Teoria do Caos aplicada à análise de sinais

Sistemas físicos reais frequentemente apresentam comportamentos que não podem ser descritos de forma adequada por modelos lineares simplificados. Em sistemas eletromecânicos, como motores acoplados a mecanismos de transmissão, efeitos como atrito, folgas mecânicas, variações de carga e processos de comutação introduzem não linearidades que influenciam diretamente a dinâmica do sistema. Essas características fazem com que pequenas variações nas condições de operação possam resultar em respostas significativamente distintas ao longo do tempo. Conforme discutido por Strogatz (2015), a presença de não linearidades é um fator determinante para o surgimento de comportamentos dinâmicos complexos, limitando a aplicação de abordagens lineares tradicionais na descrição completa desses sistemas.

Do ponto de vista matemático, o comportamento de um sistema dinâmico pode ser representado por um conjunto de equações diferenciais que descrevem a evolução temporal de suas variáveis de estado. De forma geral, um sistema dinâmico contínuo pode ser expresso conforme a Equação (2.4), em que $\mathbf{x}(t)$ representa o vetor de estados do sistema e $f(\cdot)$ uma função não linear que governa sua dinâmica, conforme a formulação clássica apresentada por Strogatz (Strogatz, 2015) e Ott (Ott, 2002).

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) \quad (2.4)$$

Em muitos casos, especialmente quando se trabalha com dados amostrados experimentalmente, a dinâmica do sistema pode ser representada em forma discreta no tempo. Nesse contexto, um sistema dinâmico discreto pode ser descrito pela Equação (2.5), na qual o estado futuro do sistema depende de seu estado atual por meio de uma relação funcional determinística, também discutida na literatura de sistemas dinâmicos não lineares (Strogatz, 2015; Ott, 2002).

$$\mathbf{x}_{n+1} = f(\mathbf{x}_n) \quad (2.5)$$

A teoria do caos estuda sistemas dinâmicos determinísticos que apresentam elevada sensibilidade às condições iniciais, resultando em comportamentos aparentemente irregulares ao longo do tempo. Apesar dessa aparência imprevisível, sistemas caóticos não são aleatórios, mas governados por leis determinísticas bem definidas. Uma característica central desses sistemas é que padrões complexos podem emergir a partir de regras simples, tornando limitada a capacidade

de previsão de longo prazo, embora a dinâmica global do sistema possa ser analisada por meio de ferramentas específicas. De acordo com Ott (2002), a identificação de comportamentos caóticos requer métodos que considerem a evolução dinâmica do sistema, indo além de descrições puramente estatísticas ou espectrais do sinal.

Na análise de sinais provenientes de sistemas dinâmicos não lineares, a teoria do caos oferece ferramentas capazes de investigar a dinâmica interna do sistema a partir de séries temporais observadas experimentalmente. Uma das abordagens mais utilizadas consiste na reconstrução do espaço de fases, na qual a evolução temporal do sistema é representada em um espaço multidimensional construído a partir de atrasos temporais da própria série medida. Essa representação preserva informações essenciais sobre a dinâmica do sistema, permitindo analisar sua estrutura geométrica e identificar padrões associados ao regime de funcionamento.

A reconstrução do espaço de fases a partir de uma série temporal escalar $x(n)$ pode ser formalmente expressa por meio do método de *delay embedding*. Nesse método, vetores no espaço $\mathbb{R}^m$ são construídos a partir de amostras defasadas no tempo, conforme a Equação (2.6), em que τ representa o atraso temporal (em número de amostras) e m a dimensão de embedding.

$$\mathbf{y}(n) = [x(n), x(n + \tau), x(n + 2\tau), \dots, x(n + (m - 1)\tau)] \quad (2.6)$$

Em função das defasagens temporais adotadas, o número efetivo de vetores disponíveis para análise é reduzido, podendo ser expresso por:

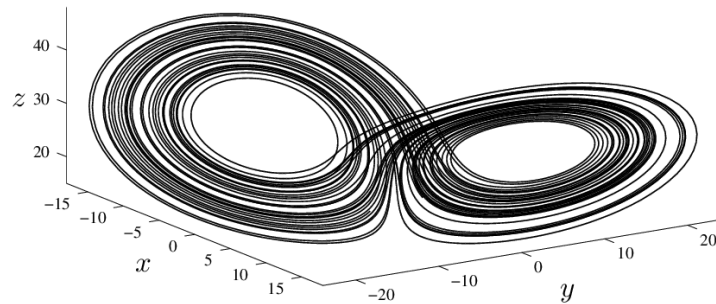
$$N_{\text{eff}} = N - (m - 1)\tau \quad (2.7)$$

Por meio da reconstrução do espaço de fases, sinais que aparentam comportamento irregular no domínio do tempo podem revelar estruturas organizadas, características de sistemas determinísticos complexos. Essa abordagem possibilita distinguir comportamentos periódicos, quase periódicos e caóticos, fornecendo uma descrição dinâmica mais rica do que análises baseadas apenas em estatísticas ou no domínio da frequência. Conforme apresentado por Kantz e Schreiber (2004), a análise no espaço de fases constitui uma ferramenta fundamental para o estudo de sistemas não lineares a partir de dados experimentais.

A Figura 2 apresenta um exemplo de atrator reconstruído no espaço de fases, ilustrando como a trajetória do sistema pode assumir uma estrutura geométrica característica quando

analisada sob a perspectiva da dinâmica caótica, evidenciando propriedades que não são diretamente observáveis na série temporal original.

Figura 2 – Representação conceitual da reconstrução do espaço de fases a partir de uma série temporal, evidenciando a estrutura dinâmica do sistema.



Fonte: Adaptado de (Almeida, 2018).

A partir dos fundamentos da teoria do caos, diversas técnicas têm sido desenvolvidas para a análise de séries temporais provenientes de sistemas dinâmicos complexos, com o objetivo de caracterizar alterações no comportamento do sistema sem a necessidade de modelos matemáticos detalhados. Essas abordagens baseiam-se na reconstrução do espaço de fases e na extração de métricas dinâmicas que refletem a estrutura do sistema ao longo do tempo, permitindo a comparação entre diferentes regimes de operação.

No contexto de sistemas eletromecânicos, técnicas derivadas da teoria do caos têm sido exploradas como alternativas complementares às análises tradicionais no domínio do tempo e da frequência, especialmente em situações nas quais efeitos não lineares desempenham papel relevante. A aplicação dessas abordagens a sinais elétricos, como a corrente consumida por motores, possibilita investigar alterações dinâmicas associadas a variações de carga, condições mecânicas ou regimes distintos de funcionamento, oferecendo uma perspectiva baseada na dinâmica do sistema, e não apenas em características estatísticas ou espectrais do sinal.

2.6 Sistema de aquisição embarcado

Em aplicações experimentais e educacionais, sistemas de aquisição embarcada têm sido amplamente utilizados como alternativa aos instrumentos tradicionais de laboratório, em função de seu baixo custo, flexibilidade e facilidade de integração com ambientes computacionais. Esses sistemas permitem a coleta de sinais elétricos diretamente no local de interesse, possibilitando o monitoramento contínuo de grandezas físicas relevantes ao comportamento de

sistemas eletromecânicos. No contexto do monitoramento de motores elétricos, a utilização de plataformas embarcadas torna viável a análise de sinais como a corrente elétrica, mesmo em cenários nos quais soluções de instrumentação dedicadas não são economicamente ou operacionalmente viáveis, reforçando sua aplicabilidade em estudos exploratórios e didáticos (Chapman, 2011; Hughes; Drury, 2019).

2.6.1 Sensor de corrente por efeito Hall

A medição de corrente elétrica por meio de sensores baseados no efeito Hall constitui uma abordagem amplamente adotada em sistemas de aquisição embarcada, especialmente quando se busca uma solução não invasiva e eletricamente isolada. O efeito Hall permite a detecção do campo magnético gerado pela passagem de corrente em um condutor, possibilitando a obtenção de um sinal proporcional à corrente elétrica sem a necessidade de inserção direta de elementos resistivos no circuito de potência. Essa característica é particularmente relevante em aplicações com motores elétricos, nas quais a corrente reflete diretamente o esforço mecânico imposto ao sistema. Conforme discutido na literatura clássica sobre acionamentos elétricos, a observação da corrente fornece informações relevantes sobre o regime de funcionamento do motor e suas variações dinâmicas (Krishnan, 2001; Boldea, 2016). A Figura 3 apresenta um exemplo de módulo sensor de corrente baseado no efeito Hall, comumente empregado em sistemas embarcados de aquisição.

2.6.2 Microcontrolador ESP32 como plataforma de aquisição

Microcontroladores têm sido amplamente empregados como plataformas de aquisição de dados em aplicações experimentais, educacionais e de pesquisa, atuando como soluções compactas e flexíveis para a coleta de sinais elétricos. Esses dispositivos integram recursos de processamento, memória e conversão analógico-digital em um único sistema, permitindo a aquisição e o armazenamento de séries temporais diretamente no ambiente embarcado. No contexto de sistemas eletromecânicos, o uso de microcontroladores possibilita a obtenção de sinais representativos do comportamento do sistema com custo reduzido e facilidade de implementação, tornando-se uma alternativa viável para estudos exploratórios e prototipagem rápida. A Figura 4 ilustra uma placa de desenvolvimento baseada no microcontrolador Microcontrolador de 32 bits da Espressif Systems (ESP32), amplamente utilizada em projetos de aquisição e processamento de sinais. A literatura clássica sobre máquinas elétricas e acionamentos destaca que, mesmo com

Figura 3 – Módulo sensor de corrente baseado no efeito Hall (ACS712).



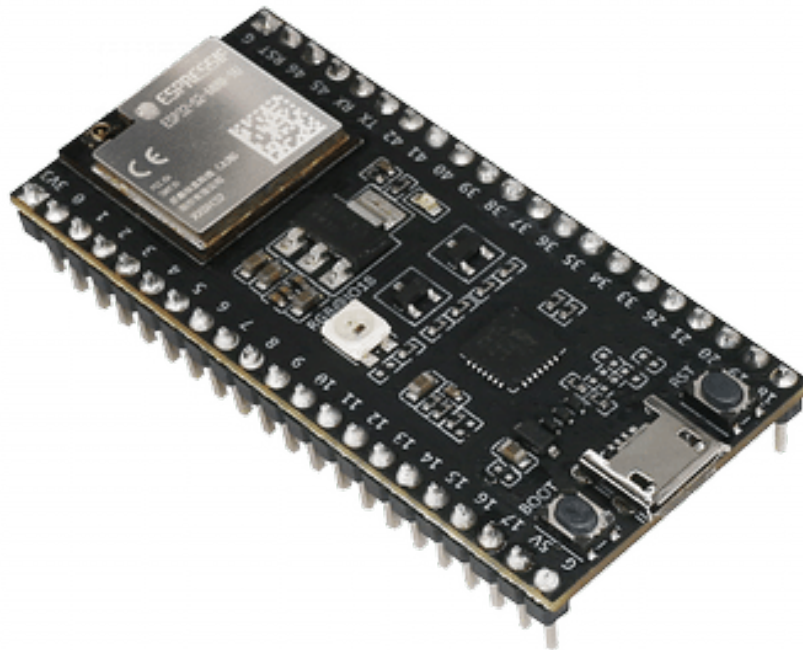
Fonte: Disponível em (Allegro Microsystems, 2024).

instrumentação simplificada, é possível extrair informações relevantes sobre o funcionamento do motor a partir de sinais elétricos adequadamente adquiridos e processados (Chapman, 2011; Hughes; Drury, 2019).

2.6.3 Conversão analógico-digital e limitações práticas

A conversão analógico-digital representa uma etapa fundamental no processo de aquisição de sinais, pois estabelece a transição entre o domínio físico contínuo e a representação discreta utilizada nas análises computacionais. Esse processo envolve a amostragem temporal do sinal e sua quantização em níveis discretos, definidos pela resolução do conversor analógico-digital. Conforme discutido por Oppenheim e Schafer (Oppenheim; Schafer, 2010) e Proakis e Manolakis (Proakis; Manolakis, 2007), a qualidade da conversão influencia diretamente a fidelidade do sinal digitalizado, estando sujeita a limitações como resolução finita, ruído e erros de quantização.

Figura 4 – Placa de desenvolvimento baseada no microcontrolador ESP32.



Fonte: Disponível em (Embarcados, 2024).

Em sistemas de aquisição embarcada, essas limitações tornam-se particularmente relevantes, uma vez que conversores integrados a microcontroladores apresentam restrições inerentes quando comparados a sistemas dedicados de instrumentação. A presença de offset, variações no intervalo de amostragem e ruído eletrônico pode afetar a forma de onda observada, impactando a análise direta de valores absolutos. No entanto, tais limitações não inviabilizam a análise do comportamento do sistema, desde que sejam consideradas adequadamente no tratamento dos dados. Nesse sentido, a adoção de estratégias de normalização e análise da dinâmica temporal do sinal permite concentrar a investigação nas variações relativas do comportamento do motor, tornando o método mais robusto frente às imperfeições do processo de aquisição (Oppenheim; Schafer, 2010; Proakis; Manolakis, 2007).

3 TRABALHOS RELACIONADOS

Este capítulo tem como objetivo apresentar trabalhos relacionados à proposta deste trabalho, destacando abordagens existentes na literatura que utilizam técnicas de análise dinâmica e não linear aplicadas a séries temporais de sistemas físicos. São apresentados resumos dos trabalhos selecionados, bem como uma análise comparativa que evidencia semelhanças e diferenças em relação à metodologia adotada neste estudo, permitindo contextualizar o escopo e as contribuições do presente trabalho.

3.1 Técnica baseada em teoria do caos para análise dinâmica de sistemas (Ramon, 2019)

O trabalho desenvolvido por Ramon propõe uma metodologia para a análise do comportamento dinâmico de sistemas físicos a partir de séries temporais experimentais, fundamentada em conceitos da teoria do caos. A abordagem baseia-se na reconstrução do espaço de fases por meio de atrasos temporais, possibilitando a investigação da dinâmica do sistema sem a necessidade de um modelo matemático explícito. A partir dessa reconstrução, são extraídas métricas não lineares, como medidas de complexidade, recorrência e instabilidade dinâmica, com o objetivo de identificar alterações no regime de funcionamento do sistema associadas à presença de falhas ou anomalias.

Os resultados apresentados por Ramon demonstram que a técnica é capaz de evidenciar mudanças dinâmicas sutis nos sinais analisados, as quais não são facilmente detectadas por métodos tradicionais baseados exclusivamente no domínio do tempo ou da frequência. Dessa forma, o trabalho estabelece uma base metodológica consistente para o uso de ferramentas da teoria do caos na análise de sistemas dinâmicos reais.

O presente trabalho inspira-se diretamente nessa abordagem metodológica, adotando os mesmos princípios fundamentais de análise não linear e reconstrução do espaço de fases. Assim como no estudo de Ramon, a proposta não depende de uma modelagem matemática detalhada do sistema físico, mas da análise direta da estrutura dinâmica do sinal medido. Em ambos os casos, assume-se que alterações no estado do sistema refletem-se na dinâmica da série temporal, podendo ser identificadas por meio de métricas caóticas.

Entretanto, o presente trabalho diferencia-se do estudo original quanto ao contexto de aplicação, ao tipo de sistema analisado e ao escopo da investigação. Enquanto Ramon desenvolve uma análise aprofundada da técnica em um cenário específico, com foco na validação

metodológica, este TCC possui caráter exploratório e aplicado, voltado à investigação da viabilidade da técnica em um motor de corrente contínua utilizado em ambiente educacional e experimental. Além disso, neste trabalho a corrente elétrica é empregada como variável observável indireta do comportamento dinâmico do sistema eletromecânico, caracterizando uma adaptação da metodologia para um novo domínio de aplicação.

3.2 Aplicação da teoria do caos no diagnóstico de falhas em máquinas rotativas (Li *et al.*, 2018)

O trabalho de Li *et al.* (2018) propõe uma abordagem para o diagnóstico de falhas em máquinas rotativas industriais baseada na teoria do caos. Os autores utilizam sinais experimentais obtidos durante a operação do sistema e aplicam técnicas de reconstrução do espaço de fases para analisar sua dinâmica não linear. A metodologia baseia-se na extração de características dinâmicas a partir das séries temporais, permitindo identificar alterações no regime de funcionamento associadas à ocorrência de falhas.

De forma semelhante ao presente trabalho, Li *et al.* utilizam conceitos da teoria do caos para analisar séries temporais provenientes de sistemas físicos reais, sem recorrer a modelos matemáticos detalhados. Em ambos os estudos, parte-se do princípio de que falhas e anomalias provocam alterações na dinâmica do sistema, passíveis de identificação por meio de métricas não lineares.

Entretanto, este trabalho diferencia-se da abordagem de Li *et al.* principalmente quanto ao contexto de aplicação e ao tipo de sinal analisado. Enquanto Li *et al.* concentram-se em máquinas industriais e utilizam predominantemente sinais de vibração, o presente estudo investiga um motor de corrente contínua em ambiente experimental, utilizando a corrente elétrica como variável observável, além de adotar um escopo exploratório voltado a aplicações de baixo custo.

3.3 Análise dinâmica não linear aplicada a sinais de vibração (Peng *et al.*, 2012)

Peng *et al.* (2012) investigam o uso de técnicas de análise dinâmica não linear para o diagnóstico de falhas em sistemas mecânicos a partir de sinais de vibração. Os autores exploram o comportamento caótico das séries temporais por meio da reconstrução do espaço de fases e da extração de métricas associadas à dinâmica não linear, demonstrando que mudanças estruturais

Quadro 1 – Análise comparativa entre os trabalhos relacionados e o presente estudo.

Trabalho	Sistema analisado	Sinal utilizado	Semelhanças	Diferenças
Ramon	Sistema dinâmico experimental	Sinal experimental indireto	Uso de reconstrução do espaço de fases e métricas caóticas	Escopo metodológico mais aprofundado e aplicação em contexto distinto
Li et al. (2018)	Máquinas rotativas industriais	Vibração	Diagnóstico de falhas via análise não linear	Foco industrial e uso de vibração em vez de corrente elétrica
Peng et al. (2012)	Sistemas mecânicos	Vibração	Análise dinâmica não linear para identificação de falhas	Escopo distinto e aplicação fora do contexto educacional
Presente trabalho	Motor de corrente contínua	Corrente elétrica	Aplicação de técnicas caóticas para análise dinâmica	Estudo exploratório, de baixo custo, sem sensores mecânicos

Fonte: elaborado pelo autor.

ou operacionais provocam alterações significativas na dinâmica do sinal.

Assim como no presente trabalho, Peng et al. partem do pressuposto de que o comportamento dinâmico de um sistema físico contém informações relevantes sobre seu estado de funcionamento. Ambos os trabalhos utilizam séries temporais experimentais e aplicam conceitos da teoria do caos como alternativa ou complemento às abordagens tradicionais de análise.

Por outro lado, o presente estudo diferencia-se quanto ao tipo de sinal analisado e ao contexto experimental. Enquanto Peng et al. utilizam sinais de vibração em sistemas mecânicos, este trabalho analisa a corrente elétrica de um motor de corrente contínua, explorando sua relação com o comportamento dinâmico do sistema eletromecânico em um cenário educacional.

3.4 Análise Comparativa

No Quadro 1 é apresentada uma análise comparativa entre os trabalhos relacionados e o presente estudo, com base em características técnicas e funcionais relevantes, evidenciando o posicionamento e as contribuições deste TCC no contexto da literatura.

4 METODOLOGIA

Este capítulo descreve os procedimentos adotados para a realização dos experimentos e para a análise dos sinais de corrente elétrica de um motor de corrente contínua. A metodologia proposta tem como objetivo investigar a viabilidade do uso da corrente elétrica como variável observável para a identificação de diferentes regimes de funcionamento do motor, por meio da aplicação de técnicas baseadas na teoria do caos.

A abordagem metodológica consiste na aquisição experimental dos sinais de corrente elétrica em diferentes condições de operação do motor. Em seguida, os sinais são condicionados, digitalizados e processados em ambiente computacional, onde são segmentados em janelas temporais e analisados por meio de métricas não lineares. Por fim, os resultados obtidos são comparados com um padrão de funcionamento considerado normal, permitindo a identificação de desvios associados a condições de carga, regimes transitórios ou estados de inoperância do sistema.

4.1 Materiais, fonte de alimentação e arquitetura do sistema

Para a realização dos experimentos, foram utilizados os seguintes materiais e equipamentos: um motor genérico de corrente contínua de pequeno porte, comumente empregado em aplicações educacionais e projetos de robótica; um sensor de corrente do tipo ACS712, versão de 5 A; um microcontrolador ESP32; uma fonte de alimentação contínua do tipo bateria, com tensão nominal de 5 V e corrente máxima de 1 A; uma chave liga/desliga instalada entre a bateria e o circuito de potência; resistores, capacitores, protoboard e cabos de conexão; além de um computador pessoal para aquisição e processamento dos dados.

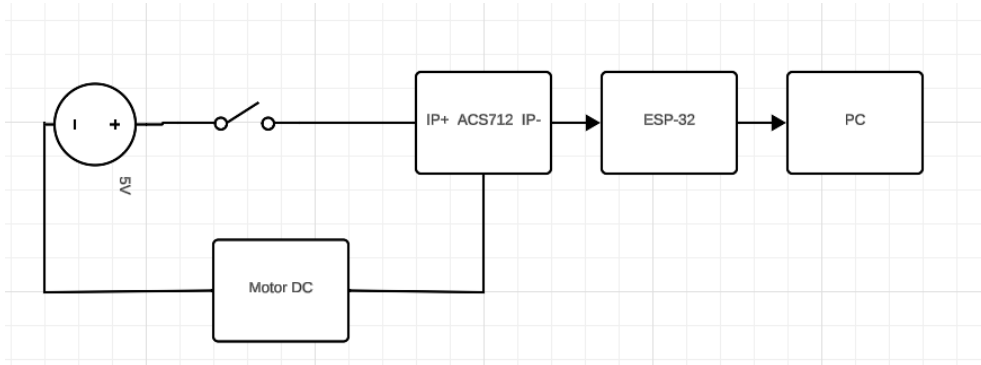
A arquitetura geral do sistema experimental é apresentada na Figura 5, evidenciando os blocos de potência, medição, aquisição digital e processamento computacional.

4.2 Montagem experimental e circuito elétrico

A montagem experimental foi concebida de modo a permitir a medição da corrente elétrica consumida pelo motor de corrente contínua, garantindo simplicidade, segurança operacional e reprodutibilidade dos ensaios. O sistema foi dividido em dois blocos principais: o circuito de potência e o circuito de medição.

No circuito de potência, toda a corrente fornecida pela bateria atravessa o sensor

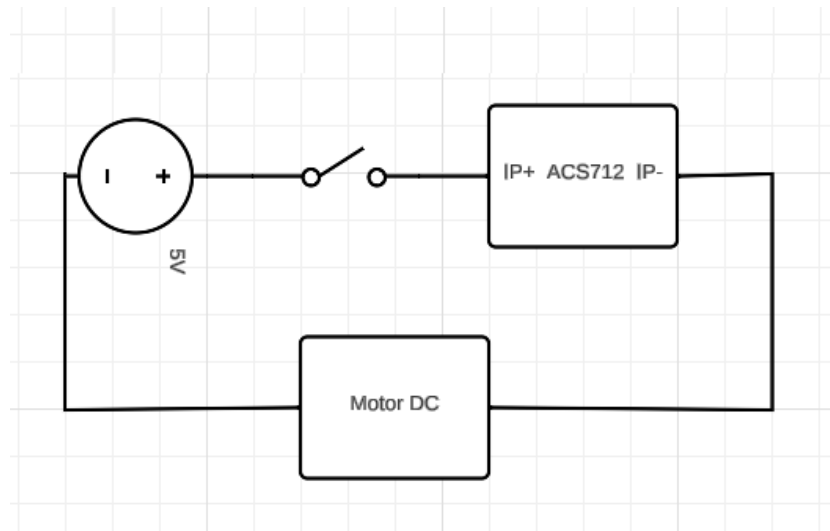
Figura 5 – Arquitetura geral do sistema experimental, incluindo circuito de potência, medição, aquisição via ESP32 e processamento em computador.



Fonte: Elaborado pelo autor.

ACS712 antes de alimentar o motor, conforme ilustrado na Figura 6. A ligação em série garante que a corrente medida corresponda integralmente à corrente consumida pelo motor durante seu funcionamento.

Figura 6 – Circuito de potência do sistema experimental, destacando a ligação em série entre bateria, sensor ACS712 e motor CC.



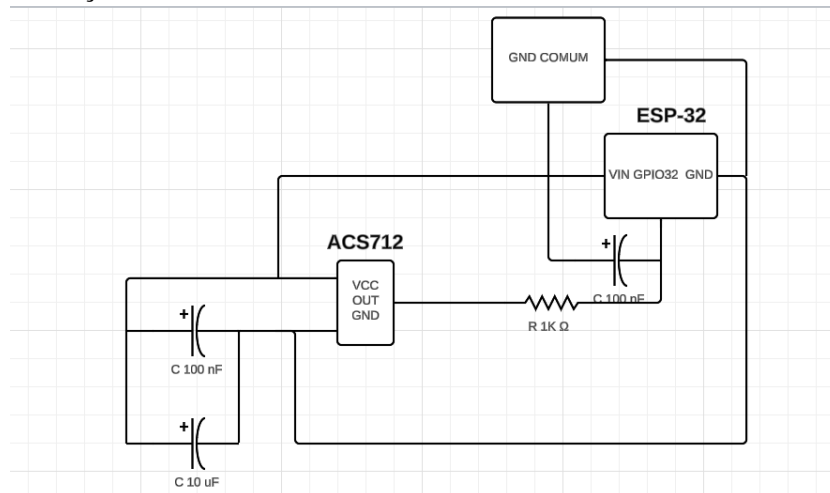
Fonte: Elaborado pelo autor.

No circuito de medição, a saída analógica do sensor Sensor de corrente baseado no efeito Hall (ACS712) foi conectada ao pino GPIO32 do ESP32. Um ponto fundamental identificado durante os testes foi a necessidade de um referencial elétrico comum. Assim, o terminal GND do sensor, do ESP32 e da bateria foram interligados, evitando leituras instáveis e flutuações indesejadas no sinal.

4.3 Condicionamento analógico do sinal

Para melhorar a qualidade do sinal analógico fornecido pelo sensor de corrente e reduzir a influência de ruídos elétricos, foram aplicadas técnicas simples de condicionamento analógico, conforme ilustrado na Figura 7.

Figura 7 – Filtro RC e desacoplamento da alimentação aplicados ao sensor ACS712 para redução de ruído e estabilização do sinal.



Fonte: Elaborado pelo autor.

Foi implementado um filtro passa-baixas RC na saída do sensor, composto por um resistor de 1 k Ω em série e um capacitor de 100 nF conectado ao terra. Adicionalmente, a alimentação do sensor foi desacoplada por meio de um capacitor cerâmico de 100 nF em paralelo com um capacitor eletrolítico de 10 μ F, posicionados próximos aos terminais VCC e GND do sensor.

4.4 Condições de operação do motor

Os experimentos foram realizados sob diferentes condições de operação. A condição normal foi definida como o motor operando livremente, sem carga mecânica adicional, sendo utilizada para a construção da baseline de normalidade.

A condição de carga mecânica contínua foi obtida por meio da aplicação de uma leve pressão manual, exercida pelos dedos do operador sobre o eixo do motor durante toda a aquisição. Essa abordagem introduz um aumento contínuo do torque requerido pelo motor, caracterizando um regime de carga externa de forma simples e funcionalmente repetível, embora sem controle quantitativo rigoroso sobre a intensidade da carga aplicada, em função do caráter

manual do procedimento.

Também foi analisado o estado inoperante, no qual o motor permaneceu desligado durante toda a aquisição, com o objetivo de avaliar a capacidade do método em distinguir a ausência de funcionamento.

Ensaio adicionais com carga intermitente foram realizados aplicando e removendo a pressão manual em ciclos aproximados de 5 s, permitindo a observação de transições dinâmicas no sinal de corrente.

4.5 Aquisição dos sinais de corrente elétrica

A aquisição dos sinais foi realizada pelo ESP32, que digitalizou o sinal analógico do sensor ACS712 utilizando seu conversor analógico-digital interno em modo de leitura pontual. Cada amostra foi associada a um carimbo de tempo obtido por temporizador interno, permitindo a reconstrução precisa da série temporal.

A comunicação entre o ESP32 e o computador foi realizada via interface serial USB. O fluxo completo de aquisição e processamento é ilustrado na Figura 8.

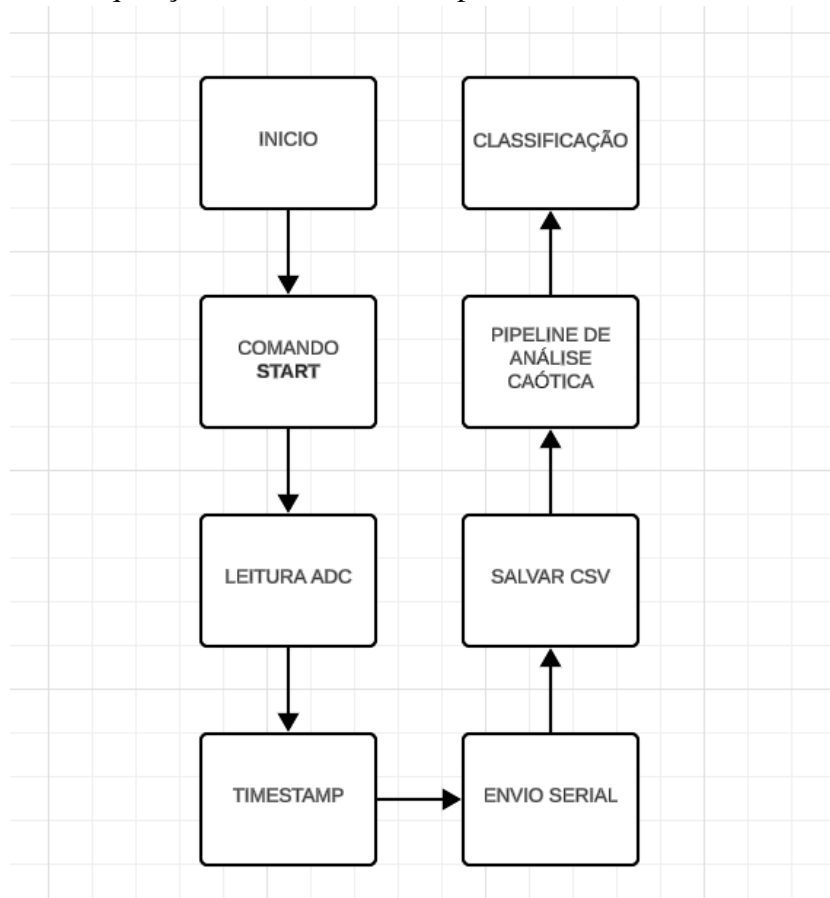
A frequência de amostragem foi definida por software e posteriormente estimada a partir dos dados adquiridos, resultando em uma frequência média entre aproximadamente 400 e 460 Hz.

4.6 Pipeline computacional e baseline

Os dados adquiridos foram processados por um pipeline computacional desenvolvido em linguagem Python, responsável pelas etapas de pré-processamento, segmentação temporal, reconstrução do espaço de fases, extração de métricas não lineares e avaliação do estado do motor. O pipeline opera em diferentes modos, incluindo calibração da baseline, avaliação de novas aquisições e execução integrada de coleta e análise.

A baseline de funcionamento normal foi construída a partir de três aquisições independentes do motor operando livremente. Para cada janela temporal, foram extraídas métricas baseadas na teoria do caos, e os valores médios e desvios padrão obtidos definiram o padrão estatístico de normalidade utilizado como referência.

Figura 8 – Fluxo de aquisição, armazenamento e processamento dos sinais de corrente elétrica.



Fonte: Elaborado pelo autor.

4.6.1 Pré-processamento do sinal e métricas no domínio do tempo

Seja $x[n]$ a série temporal discreta correspondente às amostras do sinal de corrente elétrica adquiridas pelo conversor analógico-digital. Inicialmente, foi estimado o valor médio do sinal, associado ao offset do sistema de aquisição, definido por:

$$\text{Offset} = \frac{1}{N} \sum_{n=1}^N x[n] \quad (4.1)$$

Esse valor é utilizado tanto para a correção do sinal quanto como uma métrica auxiliar, sendo apresentado nos resultados como indicador do nível DC do sistema de medição.

Após a remoção do offset, foram calculadas métricas estatísticas no domínio do tempo. O valor eficaz (RMS) do sinal foi obtido por:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N x^2[n]} \quad (4.2)$$

O valor pico-a-pico foi determinado por:

$$P2P = \max(x[n]) - \min(x[n]) \quad (4.3)$$

As métricas Offset, RMS e P2P são apresentadas nas tabelas de resultados como parâmetros descritivos do comportamento global do sinal de corrente em cada condição de operação.

4.6.2 *Reconstrução do espaço de fases*

Para a análise da dinâmica não linear do sistema, foi realizada a reconstrução do espaço de fases a partir da série temporal do sinal de corrente, conforme o teorema de imersão de Takens. A reconstrução é definida por:

$$\mathbf{x}(k) = [x(k), x(k + \tau), x(k + 2\tau), \dots, x(k + (m - 1)\tau)] \quad (4.4)$$

onde τ representa o atraso temporal e m a dimensão de embedding. Esses parâmetros foram definidos empiricamente de modo a garantir uma reconstrução adequada da dinâmica do sistema.

A reconstrução do espaço de fases constitui a base para todas as métricas não lineares apresentadas nos resultados, sendo cada valor calculado a partir das trajetórias reconstruídas.

4.6.3 *Extração de métricas não lineares*

A partir do espaço de fases reconstruído, foram extraídas métricas associadas à complexidade, recorrência e instabilidade dinâmica do sistema.

O expoente de Lyapunov λ foi estimado com o objetivo de quantificar a sensibilidade às condições iniciais, sendo valores positivos indicativos de comportamento caótico. Os valores médios de λ obtidos para cada condição de operação são apresentados nos resultados como medida de instabilidade dinâmica.

A análise de recorrência foi realizada por meio da construção da matriz de recorrência:

$$R_{i,j} = \Theta(\varepsilon - \|\mathbf{x}_i - \mathbf{x}_j\|) \quad (4.5)$$

onde $\Theta(\cdot)$ é a função degrau de Heaviside e ε o limiar de recorrência. A partir dessa matriz, foram calculadas métricas como a taxa de recorrência (RR) e o determinismo (DET), cujos valores médios por condição de operação são apresentados nas tabelas de resultados.

A entropia amostral (SampEn) foi utilizada como medida de complexidade do sinal, sendo calculada a partir da probabilidade de repetição de padrões no espaço reconstruído. Valores maiores de SampEn indicam maior irregularidade dinâmica.

4.6.4 *Escore global de desvio dinâmico*

Com o objetivo de sintetizar as variações observadas nas métricas não lineares, foi definido um escore global de desvio dinâmico, denotado por S . Esse escore é obtido a partir da normalização das métricas em relação à baseline de funcionamento normal e da agregação dos desvios observados.

Para cada aquisição, são calculados os valores médios (S_{mean}), medianos (S_{med}) e máximos (S_{max}), os quais são apresentados nos resultados como indicadores globais do afastamento do comportamento dinâmico de referência.

4.6.5 *Relação com os resultados experimentais*

Todas as métricas definidas nesta seção — Offset, RMS, P2P, expoente de Lyapunov, métricas de recorrência, entropia amostral e escore global S — são calculadas para cada janela temporal e posteriormente agregadas por condição de operação. Os valores numéricos obtidos são apresentados e discutidos no Capítulo de Resultados, permitindo a comparação direta entre os diferentes regimes de funcionamento do motor.

5 RESULTADOS

Este capítulo apresenta e discute os resultados obtidos a partir da aplicação do pipeline de análise caótica aos sinais de corrente elétrica adquiridos de um motor de corrente contínua. As avaliações foram realizadas considerando quatro condições distintas de operação: funcionamento normal (motor ligado e livre), funcionamento sob carga mecânica contínua (pressionado), estado desligado (inoperante) e funcionamento com carga mecânica intermitente (alternado).

Os resultados apresentados têm como objetivo avaliar se a dinâmica da corrente elétrica contém informações suficientes para diferenciar regimes de operação do motor, bem como investigar a capacidade do método proposto em identificar desvios dinâmicos associados a alterações mecânicas e estados de funcionamento distintos. A análise baseou-se na comparação entre métricas dinâmicas extraídas do espaço de fases reconstruído e uma referência estatística de funcionamento normal (*baseline*), construída a partir de três aquisições independentes do motor operando livremente.

5.1 Visão geral das aquisições

Todas as aquisições apresentaram duração aproximada de 30 s, com frequência de amostragem média estimada em aproximadamente 406,5 Hz. O offset do sinal, associado ao ponto médio do sensor ACS712 e às características do circuito de condicionamento, manteve-se relativamente estável dentro de cada condição de operação, apresentando apenas pequenas variações entre estados distintos.

Essas variações reduzidas de offset indicam que o sistema de aquisição apresenta comportamento consistente ao longo dos experimentos, sendo a variabilidade residual atribuída predominantemente ao ruído eletrônico e às limitações instrumentais do processo de medição. Ressalta-se que o offset é apresentado nos resultados apenas como um parâmetro descritivo do sistema de aquisição, não sendo utilizado como critério decisório para a classificação dos estados do motor.

O escore global de desvio dinâmico, denotado por S , foi calculado janela a janela a partir da normalização das métricas não lineares — entropia amostral (SampEn), expoente de Lyapunov aproximado (λ), taxa de recorrência (RR) e determinismo (DET) — em relação à média e ao desvio padrão da baseline. Para cada aquisição, foram reportados os valores da

mediana (S_{med}), média (S_{mean}) e máximo (S_{max}), permitindo avaliar tanto o comportamento dinâmico médio quanto a ocorrência de eventos transitórios localizados.

5.2 Parâmetros da reconstrução do espaço de fases

A Tabela 1 apresenta os parâmetros associados à reconstrução do espaço de fases do sinal de corrente elétrica, conforme o método de *delay embedding* definido pelas Equações (2.6) e (2.7), apresentadas na Fundamentação Teórica. São reportados o atraso temporal τ , a dimensão de embedding m e o número efetivo de pontos utilizados na reconstrução, N_{eff} .

Tabela 1 – Parâmetros da reconstrução do espaço de fases por condição de operação, calculados conforme as Equações (2.6) e (2.7).

Estado	τ (amostras)	m	N_{eff}
normal	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
pressionado	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
desligado	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
alternado	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$

Fonte: Elaborado pelo autor.

Observa-se que os valores de τ e m permaneceram consistentes entre as diferentes condições de operação, indicando que a estrutura dinâmica do sistema pode ser reconstruída de forma estável a partir do sinal de corrente elétrica. O número efetivo de pontos N_{eff} evidencia a redução do comprimento útil das séries temporais imposta pela reconstrução do espaço de fases, aspecto relevante para a confiabilidade das métricas dinâmicas extraídas.

5.3 Métricas dinâmicas por condição de operação

A Tabela 2 apresenta os valores médios das métricas dinâmicas extraídas a partir do espaço de fases reconstruído. Essas métricas refletem propriedades da dinâmica do sistema e constituem a base para o cálculo do escore global de desvio dinâmico S .

Tabela 2 – Valores médios das métricas dinâmicas por condição de operação.

Estado	SampEn	λ	RR	DET
normal	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
pressionado	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
desligado	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
alternado	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$

Fonte: Elaborado pelo autor.

De forma geral, observa-se que o estado desligado apresenta comportamento dinâmico claramente distinto, enquanto os estados normal, pressionado e alternado exibem diferenças mais sutis, coerentes com a natureza do experimento e com o caráter global das métricas utilizadas.

5.4 Resultados por condição de operação

5.4.1 Funcionamento normal

As três aquisições realizadas com o motor operando livremente foram classificadas pelo pipeline como **NORMAL**. Os valores de S_{med} variaram entre 1,12 e 1,71, indicando que, de forma global, a dinâmica da corrente elétrica manteve-se próxima ao padrão definido pela baseline.

Foram observados picos pontuais elevados em algumas aquisições, em especial na normal/coleta_3.csv, que apresentou $S_{\text{max}} = 8,89$ em uma janela localizada. Esse comportamento indica a presença de perturbações transitórias, possivelmente associadas a ruído elétrico, pequenas instabilidades mecânicas momentâneas ou interferências externas.

Apesar desses eventos localizados, a mediana do escore permaneceu em faixa compatível com o regime normal, evidenciando que o critério baseado em S_{med} é robusto a perturbações pontuais e adequado para representar o comportamento dinâmico médio do sistema.

5.4.2 Funcionamento sob carga mecânica contínua

No estado pressionado, os resultados apresentaram maior variabilidade entre as aquisições. A primeira coleta foi classificada como **NORMAL** ($S_{\text{med}} = 1,76$), enquanto as duas seguintes foram classificadas como **TRANSIÇÃO/CARGA**, com valores de $S_{\text{med}} = 2,73$ e $2,23$, respectivamente.

Esse comportamento é coerente com a aplicação manual da carga mecânica, na qual pequenas variações na intensidade e na estabilidade da pressão exercida sobre o eixo do motor resultam em alterações na dinâmica do sistema. Tais alterações refletem-se no padrão temporal da corrente elétrica e, consequentemente, nos valores das métricas dinâmicas e do escore S .

5.4.3 *Estado desligado*

As três aquisições realizadas com o motor desligado foram classificadas de forma consistente como **ANOMALIA/INOPERANTE**. Os valores de S_{med} situaram-se entre 6,48 e 6,79, significativamente superiores aos observados nos demais estados.

Nessa condição, o sinal de corrente é dominado predominantemente pelo offset do sistema de medição e pelo ruído eletrônico residual, uma vez que não há contribuição eletromecânica do motor. A baixa dispersão dos resultados evidencia elevada repetibilidade do comportamento dinâmico nesse estado.

5.4.4 *Funcionamento com carga mecânica intermitente*

No estado alternado, no qual a carga mecânica foi aplicada e removida em ciclos aproximados de 5 s, todas as aquisições foram classificadas como **NORMAL**, com valores de S_{med} entre 1,36 e 1,78.

Esse resultado indica que, considerando o critério global adotado, o comportamento dinâmico médio do sinal permaneceu próximo ao baseline. Ainda assim, os valores de S_{max} revelam a presença de janelas específicas com maior desvio dinâmico associadas aos instantes de aplicação da carga, sugerindo que análises localizadas no tempo podem fornecer informações adicionais sobre transições dinâmicas.

5.5 Resultados quantitativos

A Tabela 3 apresenta os resultados detalhados obtidos para cada aquisição individual, enquanto a Tabela 4 resume estatisticamente os escores por condição de operação.

Tabela 3 – Resultados da avaliação automática por aquisição.

Estado	Arquivo	F_s (Hz)	Offset	S_{med}	S_{max}	Classificação
normal	coleta_1.csv	406.5	2901	1.12	2.90	NORMAL
normal	coleta_2.csv	406.5	2901	1.55	3.74	NORMAL
normal	coleta_3.csv	406.5	2900	1.71	8.89	NORMAL (pico pontual)
pressionado	coleta_1.csv	406.5	2891	1.76	3.24	NORMAL
pressionado	coleta_2.csv	406.5	2893	2.73	4.77	TRANSIÇÃO/CARGA
pressionado	coleta_3.csv	406.5	2895	2.23	6.40	TRANSIÇÃO/CARGA (picos)
desligado	coleta_1.csv	406.5	2920	6.79	8.16	ANOMALIA/INOPERANTE
desligado	coleta_2.csv	406.5	2920	6.72	8.53	ANOMALIA/INOPERANTE
desligado	coleta_3.csv	406.5	2922	6.48	8.32	ANOMALIA/INOPERANTE
alternado	coleta_1.csv	406.5	2901	1.54	3.64	NORMAL
alternado	coleta_2.csv	406.5	2901	1.36	3.38	NORMAL
alternado	coleta_3.csv	406.5	2896	1.78	3.63	NORMAL

Fonte: Elaborado pelo autor.

Tabela 4 – Resumo estatístico dos escores por condição de operação.

Estado	Offset ($\mu \pm \sigma$)	S_{med} ($\mu \pm \sigma$)	S_{mean} ($\mu \pm \sigma$)	S_{max} ($\mu \pm \sigma$)
normal	2900.67 ± 0.58	1.46 ± 0.31	1.64 ± 0.44	5.18 ± 3.24
pressionado	2893.00 ± 2.00	2.24 ± 0.49	2.22 ± 0.40	4.80 ± 1.58
desligado	2920.67 ± 1.15	6.66 ± 0.16	6.83 ± 0.10	8.34 ± 0.19
alternado	2899.33 ± 2.89	1.56 ± 0.21	1.68 ± 0.22	3.55 ± 0.15

Fonte: Elaborado pelo autor.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho investigou o potencial do uso da corrente elétrica como variável observável para a análise de diferentes regimes de funcionamento de um motor de corrente contínua, por meio da aplicação de técnicas baseadas na teoria do caos. A abordagem adotada integrou a aquisição experimental de sinais reais, o condicionamento analógico do sinal, o processamento computacional estruturado e a análise não linear por janelas temporais, permitindo avaliar alterações dinâmicas no comportamento do sistema sem o uso de sensores mecânicos adicionais.

Os experimentos realizados indicaram que as métricas caóticas empregadas — entropia amostral, expoente de Lyapunov aproximado, taxa de recorrência e determinismo — apresentam sensibilidade a mudanças no regime de operação do motor. A construção de uma baseline estatística a partir do funcionamento considerado normal possibilitou a comparação quantitativa entre diferentes condições experimentais, permitindo avaliar o grau de desvio dinâmico associado a cada estado analisado.

Os resultados mostraram que, em condição de funcionamento normal, os sinais de corrente apresentam comportamento dinâmico relativamente estável, refletido em baixos valores do escore global de desvio. Em contrapartida, a aplicação de carga mecânica contínua resultou, em média, em um aumento desse escore, caracterizando um regime intermediário associado à elevação do esforço mecânico exigido pelo motor. Esse comportamento, entretanto, apresentou variabilidade entre as aquisições, refletindo a natureza manual e não controlada da aplicação de carga.

No estado inoperante, correspondente ao motor desligado, os sinais de corrente apresentaram valores significativamente elevados do escore de desvio dinâmico. Nessa condição, a dinâmica observada é predominantemente influenciada pelo offset do sistema de medição e pelo ruído eletrônico residual do processo de aquisição, uma vez que não há contribuição eletromecânica do motor. A repetibilidade dos resultados nesse estado evidencia que o método é capaz de distinguir de forma consistente sinais operantes de sinais dominados por ruído instrumental.

Os ensaios com carga mecânica intermitente evidenciaram a sensibilidade da abordagem a transições temporais no regime de funcionamento, ainda que, quando analisados de forma global, esses sinais tenham sido classificados predominantemente como normais. Esse resultado indica que critérios globais, como a mediana do escore por aquisição, podem não capturar plena-

mente comportamentos transitórios, reforçando a importância da análise por janelas temporais para a identificação localizada de desvios dinâmicos.

Dessa forma, os resultados obtidos indicam que a metodologia proposta é capaz de diferenciar regimes de funcionamento distintos do motor a partir exclusivamente da análise da corrente elétrica, utilizando métricas não lineares inspiradas na teoria do caos. No entanto, essa capacidade deve ser interpretada dentro do caráter exploratório do estudo, que teve como objetivo principal avaliar o potencial informativo da corrente elétrica, e não estabelecer um sistema definitivo de diagnóstico ou classificação.

É importante ressaltar que este trabalho apresenta limitações inerentes à sua configuração experimental. O motor analisado é de pequeno porte, a aplicação de carga mecânica foi realizada manualmente, sem controle quantitativo preciso do torque, e a aquisição dos sinais utilizou o conversor analógico-digital interno do microcontrolador, cujas limitações de resolução, linearidade e ruído influenciam diretamente o sinal observado. Essas restrições impactam a generalização dos resultados e limitam a precisão da correlação entre esforço mecânico e alterações dinâmicas da corrente.

Apesar dessas limitações, os resultados obtidos são coerentes com a fundamentação teórica adotada e demonstram que a corrente elétrica contém informações relevantes sobre a dinâmica do sistema eletromecânico. Nesse sentido, os objetivos do trabalho foram atendidos de forma parcial e investigativa, contribuindo para a compreensão do comportamento dinâmico do motor sob diferentes condições de operação e para a avaliação do uso de técnicas caóticas aplicadas a sinais elétricos reais.

Como continuidade natural deste estudo, trabalhos futuros podem explorar a utilização de conversores analógico-digital externos, com maior resolução e estabilidade, visando reduzir a influência do ruído e melhorar a qualidade da aquisição dos sinais. A realização de experimentos com aplicação de cargas mecânicas controladas, por meio de dispositivos instrumentados ou bancadas de ensaio, permitiria uma análise mais precisa da relação entre torque aplicado e alterações dinâmicas no sinal de corrente.

Outras extensões relevantes incluem a comparação entre diferentes plataformas de aquisição, como microcontroladores e osciloscópios digitais, bem como a investigação de outros tipos de perturbações, como variações na tensão de alimentação, defeitos simulados ou mudanças abruptas de carga. No âmbito do processamento, a integração das métricas caóticas com técnicas de aprendizado de máquina pode contribuir para a construção de classificadores automáticos e

para a avaliação mais sistemática da capacidade discriminativa do método.

Por fim, este trabalho estabelece uma base conceitual e experimental para estudos posteriores voltados à análise dinâmica e ao monitoramento de sistemas eletromecânicos, indicando que a corrente elétrica, quando analisada sob a perspectiva da teoria do caos, constitui uma fonte promissora de informação, ainda que sujeita a limitações que devem ser cuidadosamente consideradas em aplicações futuras.

REFERÊNCIAS

- Allegro Microsystems. **ACS712 Hall Effect-Based Linear Current Sensor**. 2024. Disponível em: <https://share.google/2yFCb1Pk2W0L7157u>. Acesso em: 1 dez. 2026.
- ALMEIDA, F. M. J. d. **Análise de sistemas dinâmicos não lineares aplicada a séries temporais**. Dissertação (Mestrado) – Universidade de São Paulo, 2018. Disponível em: https://bdta.abcd.usp.br/directbitstream/b89b489b-aaca-4d92-8412-bb7616d363f4/Almeida_Fernanda_Melo_jacques_de.pdf. Acesso em: 1 dez. 2026.
- BOLDEA, I. **Electric Drives**. Boca Raton: CRC Press, 2016.
- BRACEWELL, R. N. **The Fourier Transform and Its Applications**. 3. ed. New York: McGraw-Hill, 2000.
- CHAPMAN, S. J. **Electric Machinery Fundamentals**. 5. ed. New York: McGraw-Hill, 2011.
- Embarcados. **Placa ESP32-S2 DevKitM-1**. 2024. Disponível em: <https://embarcados.com.br/placa/esp32-s2-devkitm-1/>. Acesso em: 1 dez. 2026.
- HUGHES, A.; DRURY, B. **Electric Motors and Drives: Fundamentals, types and applications**. 5. ed. Oxford: Butterworth-Heinemann, 2019.
- JARDINE, A. K. S.; LIN, D.; BANJEVIC, D. **Maintenance, Replacement, and Reliability: Theory and applications**. 2. ed. Boca Raton: CRC Press, 2013.
- KRISHNAN, R. **Electric Motor Drives: Modeling, analysis, and control**. Upper Saddle River: Prentice Hall, 2001.
- LEI, Y.; HE, Z.; ZI, Y. Signal processing techniques for fault diagnosis of rotating machinery. **Mechanical Systems and Signal Processing**, v. 45, n. 1, p. 1–16, 2014.
- LI, X.; ZHANG, W.; DING, Q. Chaos theory based fault diagnosis of rotating machinery. **Mechanical Systems and Signal Processing**, v. 102, p. 1–16, 2018.
- OPPENHEIM, A. V.; SCHAFER, R. W. **Signals and Systems**. 2. ed. Upper Saddle River: Prentice Hall, 2010.
- OTT, E. **Chaos in Dynamical Systems**. 2. ed. Cambridge: Cambridge University Press, 2002.
- PENG, Z. K.; TSE, P. W.; CHU, F. L. A comparison study of improved hilbert–huang transform and wavelet transform: Application to fault diagnosis for rolling bearing. **Mechanical Systems and Signal Processing**, v. 24, n. 7, p. 2157–2174, 2012.
- PROAKIS, J. G.; MANOLAKIS, D. G. **Digital Signal Processing: Principles, algorithms, and applications**. 4. ed. Upper Saddle River: Pearson Prentice Hall, 2007.
- RAMON, F. S. Técnica baseada em teoria do caos para análise dinâmica de sistemas físicos. **Revista Brasileira de Engenharia**, v. 38, n. 2, p. 215–228, 2019.
- RoboMix. **Motor DC 3V a 6V com caixa de redução e eixo duplo**. 2024. Disponível em: <https://www.robomix.com.br/motor-dc-3v-a-6v-caixa-de-reducao-e-eixo-duplo>. Acesso em: 1 dez. 2026.

STROGATZ, S. H. **Nonlinear Dynamics and Chaos**: With applications to physics, biology, chemistry, and engineering. 2. ed. Boulder: Westview Press, 2015.

THORSEN, O. V.; DALVA, M. Condition monitoring of induction motors using current measurements. **IEEE Transactions on Industry Applications**, v. 31, n. 4, p. 923–929, 1995.

VERMA, A. K.; SARANGI, S.; KOLEKAR, M. H. A review of condition monitoring and fault diagnosis of induction motors. **Archives of Computational Methods in Engineering**, v. 28, n. 4, p. 2127–2150, 2021.

APÊNDICE A – CÓDIGO-FONTE DO PIPELINE DE ANÁLISE CAÓTICA

Este apêndice apresenta o código-fonte completo do pipeline computacional desenvolvido para a aquisição, processamento e análise dos sinais de corrente elétrica do motor de corrente contínua. O código foi implementado em linguagem Python e contempla as rotinas de leitura dos dados experimentais, pré-processamento do sinal, extração das métricas baseadas na teoria do caos, construção da baseline de funcionamento normal e avaliação automática dos estados do motor.

A inclusão do código no apêndice tem como objetivo garantir a reprodutibilidade dos experimentos, bem como fornecer subsídios para futuras extensões e validações da metodologia proposta.

Código-fonte 1 – Pipeline computacional para análise caótica da corrente elétrica

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  """
5  pipeline_analise_caotica.py (PT-BR)
6
7  Pipeline de análise não linear (caos) aplicado ao sinal
8  de corrente de um motor DC
9  (adquirido via ACS712 + ESP32), com baseline do estado
10  NORMAL (motor ligado livre).
11
12  Modos:
13  1) calibrar: cria baseline.json a partir de uma pasta com
14  CSVs do estado NORMAL.
15  2) avaliar: avalia 1 CSV, gera score e classifica (NORMAL
16  / TRANSICAO-CARGA / ANOMALIA-INOPERANTE).
17  3) coletar: (opcional) coleta via serial e salva CSV.
18  4) run: (opcional) coleta e j avalia usando baseline.
19  json.

```

```

16 Observa o importante:
17 - "ANOMALIA/INOPERANTE" inclui o caso de motor desligado,
    pois a dinamica colapsa e o desvio do normal cresce
    muito.
18 """
19
20 import argparse
21 import csv
22 import json
23 import time
24 from dataclasses import dataclass
25 from datetime import datetime
26 from pathlib import Path
27 from typing import Dict, List, Optional, Tuple
28
29 import numpy as np
30
31 try:
32     import serial # pyserial
33 except ImportError:
34     serial = None
35
36
37 # =====
38 # Estruturas / Config
39 # =====
40
41 @dataclass
42 class SerialConfig:
43     porta: str
44     baud: int
45     duracao_s: int

```

```

46     prefixo: str
47     espera_reset_s: float = 2.0
48
49
50 @dataclass
51 class ChaosFeatures:
52     sampen: float
53     lyap: float
54     rr: float
55     det: float
56
57
58 # =====
59 # I/O: Coleta via Serial
60 # =====
61
62 def coletar_csv(cfg: SerialConfig, out_dir: Path) -> Path:
63     """
64     Coleta dados via serial do ESP32.
65     Espera por marcadores #START / #END e salva CSV com
66         colunas: t, adc.
67     """
68     if serial is None:
69         raise RuntimeError("pyserial n o est instalado.
70             Rode: pip install pyserial")
71
72     out_dir.mkdir(parents=True, exist_ok=True)
73     nome_arquivo = f"{cfg.prefixo}_{datetime.now().strftime(
74         ('%Y%m%d_%H%M%S'))}.csv"
75     out_path = out_dir / nome_arquivo
76
77     print(f"[COLETA] Abrindo serial {cfg.porta} @ {cfg.baud

```

```
        } ...")
75     ser = serial.Serial(cfg.porta, cfg.baud, timeout=1)
76
77     # Ao abrir a serial, muitas placas resetam. Aguarde
        estabilizar.
78     time.sleep(cfg.espera_reset_s)
79     ser.reset_input_buffer()
80
81     cmd = f"START,{cfg.duracao_s}\n"
82     ser.write(cmd.encode())
83
84     started = False
85     n = 0
86
87     with out_path.open("w", newline="") as f:
88         w = csv.writer(f)
89         w.writerow(["t", "adc"])
90
91         while True:
92             line = ser.readline().decode(errors="ignore").
                strip()
93             if not line:
94                 continue
95
96             if line == "#START":
97                 started = True
98                 continue
99
100             if line == "#END":
101                 break
102
103             if not started:
```

```

104         continue
105
106         if line.lower() == "t,adc":
107             continue
108
109         parts = line.split(",")
110         if len(parts) != 2:
111             continue
112
113         t_s, adc = parts[0].strip(), parts[1].strip()
114         try:
115             float(t_s)
116             int(adc)
117         except ValueError:
118             continue
119
120         w.writerow([t_s, adc])
121         n += 1
122
123     ser.close()
124     print(f"[COLETA] OK: {out_path.name} | amostras={n}")
125     return out_path
126
127
128 # =====
129 # Leitura CSV / Pr -processamento
130 # =====
131
132 def ler_csv(path: Path) -> Tuple[np.ndarray, np.ndarray]:
133     """L     CSV com cabe alho t,adc e retorna arrays t (s)
134         e x (adc)."""
135     t_list, x_list = [], []

```



```

135     with path.open("r", newline="") as f:
136         reader = csv.DictReader(f)
137         for row in reader:
138             try:
139                 t_list.append(float(row["t"]))
140                 x_list.append(int(row["adc"]))
141             except Exception:
142                 continue
143     t = np.asarray(t_list, dtype=np.float64)
144     x = np.asarray(x_list, dtype=np.float64)
145     return t, x
146
147
148 def estimar_fs(t: np.ndarray) -> float:
149     """Fs estimada pela mediana de dt."""
150     if len(t) < 3:
151         return 0.0
152     dt = np.diff(t)
153     dt = dt[np.isfinite(dt) & (dt > 0)]
154     if len(dt) == 0:
155         return 0.0
156     return float(1.0 / np.median(dt))
157
158
159 def remover_offset_por_mediana(x: np.ndarray, t: np.ndarray
160     , base_s: float = 2.0) -> Tuple[np.ndarray, float]:
161     """
162     Remove offset DC estimando a mediana no inicio (
163         primeiros base_s segundos).
164     Retorna d = x - offset e o offset.
165     """
166     if len(t) == 0:

```

```

165         return x.copy(), 0.0
166
167     t0 = t[0]
168     mask = (t >= t0) & (t < (t0 + base_s))
169     if np.sum(mask) < 20:
170         n = min(len(x), 200)
171         mask = np.zeros(len(x), dtype=bool)
172         mask[:n] = True
173
174     offset = float(np.median(x[mask]))
175     d = x - offset
176     return d, offset
177
178
179 def normalizar_zscore(d: np.ndarray) -> np.ndarray:
180     """Normaliza (z-score) para reduzir efeito de ganho e
181         facilitar compara o entre janelas."""
182     mu = float(np.mean(d))
183     sd = float(np.std(d))
184     if sd <= 0:
185         return np.zeros_like(d)
186     return (d - mu) / sd
187
188 # =====
189 # Sele o de e Embedding
190 # =====
191
192 def escolher_tau_autocorr(x: np.ndarray, max_tau: int =
193     200) -> int:
194     """
195     : primeiro atraso em que autocorrela o cai abaixo

```

```

        de 1/e (heur stica simples e defens vel).
195     """
196     x0 = x - np.mean(x)
197     sd = np.std(x0)
198     if sd <= 0:
199         return 1
200
201     ac0 = np.dot(x0, x0)
202     if ac0 <= 0:
203         return 1
204
205     target = 1.0 / np.e
206     for tau in range(1, max_tau + 1):
207         a = np.dot(x0[:-tau], x0[tau:]) / ac0
208         if a < target:
209             return tau
210     return min(50, max_tau) if max_tau >= 50 else max_tau
211
212
213 def embed(x: np.ndarray, m: int, tau: int) -> np.ndarray:
214     """Embedding temporal (Takens) com dimens o m e atraso
        tau."""
215     n = len(x) - (m - 1) * tau
216     if n <= 10:
217         return np.empty((0, m), dtype=np.float64)
218     Y = np.zeros((n, m), dtype=np.float64)
219     for j in range(m):
220         Y[:, j] = x[j * tau : j * tau + n]
221     return Y
222
223
224 # =====

```

```

225 # SampEn (enxuta)
226 # =====
227
228 def sample_entropy(x: np.ndarray, m: int = 2, r: Optional[
    float] = None) -> float:
229     """
230     Sample Entropy (SampEn) simples ( $O(N^2)$ ) - suficiente
        para janelas ~400 pts.
231     """
232     x = np.asarray(x, dtype=np.float64)
233     n = len(x)
234     if n < m + 6:
235         return float("nan")
236
237     sd = float(np.std(x))
238     if sd <= 0:
239         return 0.0
240     if r is None:
241         r = 0.2 * sd
242
243     def _phi(mm: int) -> float:
244         N = n - mm + 1
245         if N <= 1:
246             return 0.0
247         count = 0
248         total = 0
249         for i in range(N):
250             xi = x[i : i + mm]
251             for j in range(i + 1, N):
252                 xj = x[j : j + mm]
253                 total += 1
254                 if np.max(np.abs(xi - xj)) <= r:

```

```

255         count += 1
256     return count / total if total > 0 else 0.0
257
258     A = _phi(m + 1)
259     B = _phi(m)
260
261     if B <= 0 or A <= 0:
262         return float("inf")
263     return float(-np.log(A / B))
264
265
266 # =====
267 # Lyapunov (Rosenstein - aprox)
268 # =====
269
270 def lyapunov_rosenstein(x: np.ndarray, m: int, tau: int,
271     k_max: int = 25, theiler: int = 20) -> float:
272     """
273     Estimativa aproximada do expoente de Lyapunov (
274         Rosenstein).
275     """
276     x = np.asarray(x, dtype=np.float64)
277     Y = embed(x, m=m, tau=tau)
278     if Y.shape[0] < (k_max + 15):
279         return float("nan")
280
281     n = Y.shape[0]
282
283     # vizinho mais pr ximo (naive) com janela de Theiler
284     nn = np.full(n, -1, dtype=int)
285     for i in range(n):
286         best_j = -1

```

```

285         best_d = np.inf
286         for j in range(n):
287             if abs(i - j) <= theiler:
288                 continue
289             d = np.linalg.norm(Y[i] - Y[j])
290             if 0 < d < best_d:
291                 best_d = d
292                 best_j = j
293         nn[i] = best_j
294
295     div = []
296     for k in range(1, k_max + 1):
297         vals = []
298         for i in range(n - k):
299             j = nn[i]
300             if j < 0 or (j + k) >= n:
301                 continue
302             dk = np.linalg.norm(Y[i + k] - Y[j + k])
303             if dk > 0:
304                 vals.append(np.log(dk))
305         div.append(np.nanmean(vals) if len(vals) >= 10 else
306                    np.nan)
307
308     div = np.asarray(div, dtype=np.float64)
309     ks = np.arange(1, k_max + 1, dtype=np.float64)
310
311     mask = np.isfinite(div)
312     if np.sum(mask) < 8:
313         return float("nan")
314
315     ks2 = ks[mask]
316     div2 = div[mask]

```

```

316     cut = max(8, int(0.6 * len(ks2)))
317     ks2, div2 = ks2[:cut], div2[:cut]
318
319     A = np.vstack([ks2, np.ones_like(ks2)]).T
320     slope, _ = np.linalg.lstsq(A, div2, rcond=None)[0]
321     return float(slope)
322
323
324 # =====
325 # RQA (RR e DET) - enxutos
326 # =====
327
328 def rqa_rr_det(x: np.ndarray, m: int, tau: int,
329               eps_quantile: float = 0.10, lmin: int = 2) -> Tuple[
330               float, float]:
331     """
332     RR: Recurrence Rate
333     DET: Determinism (propor o de pontos recorrentes em
334           linhas diagonais >= lmin)
335     """
336     Y = embed(x, m=m, tau=tau)
337     if Y.shape[0] < 40:
338         return float("nan"), float("nan")
339
340     n = Y.shape[0]
341     D = np.sqrt(((Y[:, None, :] - Y[None, :, :]) ** 2).sum(
342         axis=2))
343
344     dvals = D[np.triu_indices(n, k=1)]
345     if len(dvals) == 0:
346         return float("nan"), float("nan")

```

```

344     eps = float(np.quantile(dvals, eps_quantile))
345     R = (D <= eps).astype(np.uint8)
346     np.fill_diagonal(R, 0)
347
348     RR = float(R.sum() / (n * n))
349     rec_points = int(R.sum())
350     if rec_points == 0:
351         return RR, 0.0
352
353     det_points = 0
354     for k in range(-(n - 1), n):
355         diag = np.diagonal(R, offset=k)
356         if diag.size < lmin:
357             continue
358         run = 0
359         for v in diag:
360             if v:
361                 run += 1
362             else:
363                 if run >= lmin:
364                     det_points += run
365                 run = 0
366         if run >= lmin:
367             det_points += run
368
369     DET = float(det_points / rec_points) if rec_points > 0
370         else 0.0
371     return RR, DET
372
373 # =====
374 # Features de caos por janela

```



```

375 # =====
376
377 def extrair_features_caos_por_janela(
378     t: np.ndarray,
379     x: np.ndarray,
380     base_s: float = 2.0,
381     win_s: float = 1.0,
382     hop_s: float = 1.0,
383     m: int = 3,
384     tau_max: int = 200
385 ) -> Tuple[Dict[str, float], List[ChaosFeatures], List[
    Tuple[float, float]]]:
386     """
387     Retorna:
388         - resumo global (fs, dur, offset)
389         - lista de features por janela
390         - lista de spans (t0,t1) por janela (para reportar
          janelas cr ticas)
391     """
392     fs = estimar_fs(t)
393     if fs <= 0:
394         raise ValueError("Fs inv lida (verifique
          timestamps no CSV).")
395
396     dur_s = float(t[-1] - t[0]) if len(t) > 1 else 0.0
397
398     d, offset = remover_offset_por_mediana(x, t, base_s=
          base_s)
399
400     win_n = max(30, int(win_s * fs))
401     hop_n = max(1, int(hop_s * fs))
402

```

```

403     feats: List[ChaosFeatures] = []
404     spans: List[Tuple[float, float]] = []
405
406     for i0 in range(0, len(d) - win_n + 1, hop_n):
407         i1 = i0 + win_n
408         seg_t = t[i0:i1]
409         seg_d = d[i0:i1]
410
411         xn = normalizar_zscore(seg_d)
412         tau = escolher_tau_autocorr(xn, max_tau=tau_max)
413
414         sampen = sample_entropy(xn, m=2)
415         lyap = lyapunov_rosenstein(xn, m=m, tau=tau, k_max
416                                     =25, theiler=max(10, tau))
417
418         rr, det = rqa_rr_det(xn, m=m, tau=tau, eps_quantile
419                             =0.10, lmin=2)
420
421         feats.append(ChaosFeatures(sampen=sampen, lyap=lyap
422                                   , rr=rr, det=det))
423         spans.append((float(seg_t[0]), float(seg_t[-1])))
424
425     resumo = {"fs": fs, "dur_s": dur_s, "offset": offset}
426     return resumo, feats, spans
427
428
429 def feats_to_matrix(feats: List[ChaosFeatures]) -> np.
430     ndarray:
431         return np.array([[f.sampen, f.lyap, f.rr, f.det] for f
432                           in feats], dtype=np.float64)
433
434 # =====

```

```

430 # Baseline (NORMAL): m dia e desvio
431 # =====
432
433 def calibrar_baseline(
434     pasta_csvs: Path,
435     out_baseline: Path,
436     base_s: float,
437     win_s: float,
438     hop_s: float,
439     m: int
440 ) -> None:
441     csvs = sorted([p for p in pasta_csvs.glob("*.csv") if p
442                     .is_file()])
443     if not csvs:
444         raise RuntimeError(f"Nenhum CSV encontrado em: {
445                             pasta_csvs}")
446
447     X_all = []
448     meta = {
449         "normal_definicao": "MOTOR_LIGADO_LIVRE",
450         "parametros": {"base_s": base_s, "win_s": win_s, "
451                        hop_s": hop_s, "m": m},
452         "arquivos": [p.name for p in csvs],
453     }
454
455     for p in csvs:
456         t, x = ler_csv(p)
457         _, feats, _ = extrair_features_caos_por_janela(
458             t, x, base_s=base_s, win_s=win_s, hop_s=hop_s,
459             m=m
460         )
461         X = feats_to_matrix(feats)

```

```

458
459     mask = np.all(np.isfinite(X), axis=1)
460     X = X[mask]
461     if X.shape[0] < 5:
462         continue
463     X_all.append(X)
464
465 if not X_all:
466     raise RuntimeError("Falha ao gerar baseline: poucas
467                         janelas v lidas nos CSVs normais.")
468
469 Xn = np.vstack(X_all)
470 mu = np.mean(Xn, axis=0)
471 sd = np.std(Xn, axis=0)
472 sd[sd < 1e-9] = 1e-9
473
474 baseline = {
475     "meta": meta,
476     "features": ["sampen", "lyap", "rr", "det"],
477     "mean": {"sampen": float(mu[0]), "lyap": float(mu
478             [1]), "rr": float(mu[2]), "det": float(mu[3])},
479     "std": {"sampen": float(sd[0]), "lyap": float(sd
480             [1]), "rr": float(sd[2]), "det": float(sd[3])},
481 }
482
483 out_baseline.parent.mkdir(parents=True, exist_ok=True)
484 out_baseline.write_text(json.dumps(baseline, indent=2,
485     ensure_ascii=False), encoding="utf-8")
486
487 print(f"[CALIBRAR] Baseline salvo em: {out_baseline}")
488 print(f"[CALIBRAR] janelas_normais_total={Xn.shape
489     [0]}")

```

```

485     print(f"[CALIBRAR] mean: sampen={mu[0]:.4f}, lyap={mu
        [1]:.4f}, rr={mu[2]:.4f}, det={mu[3]:.4f}")
486     print(f"[CALIBRAR] std : sampen={sd[0]:.4f}, lyap={sd
        [1]:.4f}, rr={sd[2]:.4f}, det={sd[3]:.4f}")
487
488
489 def carregar_baseline(path: Path) -> Dict[str, Dict[str,
    float]]:
490     data = json.loads(path.read_text(encoding="utf-8"))
491     mu = data["mean"]
492     sd = data["std"]
493     return {"mean": mu, "std": sd, "meta": data.get("meta",
        {}), "features": data.get("features", [])}
494
495
496 # =====
497 # Score S (desvio do normal) e decis o (AJUSTE FINO)
498 # =====
499
500 def score_por_janela(X: np.ndarray, mu: np.ndarray, sd: np.
    ndarray) -> np.ndarray:
501     """
502     S = norma L2 do z-score das features.
503     """
504     Z = (X - mu) / sd
505     return np.sqrt(np.sum(Z**2, axis=1))
506
507
508 def classificar_estado(S_med: float, S_max: float,
    lim_normal: float, lim_anom: float, lim_pico: float) ->
    str:
509     """

```

```

510     Ajuste fino baseado nos seus resultados reais:
511         - S_med captura o regime "t pico"
512         - S_max captura picos/transientes (ex.: in cio ,
           varia o de press o , etc.)
513
514     Regras:
515         - S_med < lim_normal -> NORMAL
516         - lim_normal <= S_med < lim_anom -> TRANSICAO/CARGA (
           com indicador de pico forte se S_max>=lim_pico)
517         - S_med >= lim_anom -> ANOMALIA/INOOPERANTE
518     """
519     if S_med < lim_normal:
520         return "NORMAL"
521
522     if S_med < lim_anom:
523         if S_max >= lim_pico:
524             return "TRANSICAO/CARGA (PICOS FORTES)"
525         return "TRANSICAO/CARGA"
526
527     return "ANOMALIA/INOOPERANTE"
528
529
530 def avaliar_csv(
531     csv_path: Path,
532     baseline_path: Path,
533     base_s: float,
534     win_s: float,
535     hop_s: float,
536     m: int,
537     lim_normal: float,
538     lim_anom: float,
539     lim_pico: float,

```

```

540     topk: int = 3
541 ) -> None:
542     base = carregar_baseline(baseline_path)
543
544     mu = np.array([base["mean"]["sampen"], base["mean"]["lyap"], base["mean"]["rr"], base["mean"]["det"]],
545                   dtype=np.float64)
546     sd = np.array([base["std"]["sampen"], base["std"]["lyap"], base["std"]["rr"], base["std"]["det"]],
547                   dtype=np.float64)
548     sd[sd < 1e-9] = 1e-9
549
550     t, x = ler_csv(csv_path)
551     resumo, feats, spans = extrair_features_caos_por_janela(
552         t, x, base_s=base_s, win_s=win_s, hop_s=hop_s, m=m
553     )
554     X = feats_to_matrix(feats)
555
556     mask = np.all(np.isfinite(X), axis=1)
557     Xv = X[mask]
558     spans_v = [spans[i] for i in range(len(spans)) if mask[i]]
559
560     if Xv.shape[0] < 5:
561         print(f"[AVALIAR] {csv_path.name} | ERRO: poucas janelas v lidas.")
562         return
563
564     S = score_por_janela(Xv, mu, sd)
565     S_med = float(np.median(S))
566     S_mean = float(np.mean(S))

```

```

565     S_max = float(np.max(S))
566
567     estado = classificar_estado(S_med, S_max, lim_normal,
568                                  lim_anom, lim_pico)
569
570     # Top-K janelas cr ticas
571     idx = np.argsort(-S)[:topk]
572     tops = []
573     for k in idx:
574         a, b = spans_v[int(k)]
575         tops.append(f"{a:.0f}-{b:.0f}s(S={S[int(k)]:.2f})")
576
577     # Sa da ENXUTA (como voc pediu)
578     print(
579         f"[AVALIAR] {csv_path.name} | Fs={resumo['fs']:.1f}
580         Hz | dur={resumo['dur_s']:.1f}s | off={resumo['
581         offset']:.0f} | "
582         f"S_med={S_med:.2f} | S_mean={S_mean:.2f} | S_max={
583         S_max:.2f} | Estado={estado} | Top={''; '.join(
584         tops)}"
585     )
586
587     # =====
588     # CLI / Main
589     # =====
590
591     def parse_args():
592         p = argparse.ArgumentParser(description="Pipeline de
593             an lise ca tica (PT-BR) para motor DC via corrente
594             (ACS712 + ESP32).")
595         sub = p.add_subparsers(dest="cmd", required=True)

```



```

590
591 c = sub.add_parser("coletar", help="Coleta CSV via
      serial (ESP32).")
592 c.add_argument("--porta", default="/dev/ttyUSB0")
593 c.add_argument("--baud", type=int, default=115200)
594 c.add_argument("--duracao", type=int, default=30)
595 c.add_argument("--prefixo", default="normal")
596 c.add_argument("--outdir", default="coletas")
597
598 cal = sub.add_parser("calibrar", help="Cria baseline.
      json a partir de CSVs NORMAIS (motor ligado livre)
      .")
599 cal.add_argument("pasta_normais", help="Pasta contendo
      CSVs do estado NORMAL.")
600 cal.add_argument("--baseline_out", default="baseline/
      baseline.json")
601 cal.add_argument("--base_s", type=float, default=2.0)
602 cal.add_argument("--win_s", type=float, default=1.0)
603 cal.add_argument("--hop_s", type=float, default=1.0)
604 cal.add_argument("--m", type=int, default=3)
605
606 a = sub.add_parser("avaliar", help="Avalia um CSV
      comparando com baseline.json.")
607 a.add_argument("csv", help="CSV a avaliar.")
608 a.add_argument("--baseline", default="baseline/baseline
      .json")
609 a.add_argument("--base_s", type=float, default=2.0)
610 a.add_argument("--win_s", type=float, default=1.0)
611 a.add_argument("--hop_s", type=float, default=1.0)
612 a.add_argument("--m", type=int, default=3)
613
614 # Ajuste fino via CLI (valores padr o baseados no que

```

```

        voc observou)
615 a.add_argument("--lim_normal", type=float, default=2.0,
        help="Abaixo disso -> NORMAL (mediana do S).")
616 a.add_argument("--lim_anom", type=float, default=4.0,
        help="Acima disso -> ANOMALIA/INOOPERANTE (mediana do
        S).")
617 a.add_argument("--lim_pico", type=float, default=6.0,
        help="Acima disso -> marca 'PICOS FORTES' (m ximo
        do S).")
618 a.add_argument("--topk", type=int, default=3)
619
620 r = sub.add_parser("run", help="Coleta e j avalia (
        precisa de baseline.json).")
621 r.add_argument("--porta", default="/dev/ttyUSB0")
622 r.add_argument("--baud", type=int, default=115200)
623 r.add_argument("--duracao", type=int, default=30)
624 r.add_argument("--prefixo", default="teste")
625 r.add_argument("--outdir", default="coletas")
626 r.add_argument("--baseline", default="baseline/baseline
        .json")
627 r.add_argument("--base_s", type=float, default=2.0)
628 r.add_argument("--win_s", type=float, default=1.0)
629 r.add_argument("--hop_s", type=float, default=1.0)
630 r.add_argument("--m", type=int, default=3)
631 r.add_argument("--lim_normal", type=float, default=2.0)
632 r.add_argument("--lim_anom", type=float, default=4.0)
633 r.add_argument("--lim_pico", type=float, default=6.0)
634 r.add_argument("--topk", type=int, default=3)
635
636 return p.parse_args()
637
638

```

```
639 def main():
640     args = parse_args()
641
642     if args.cmd == "coletar":
643         cfg = SerialConfig(
644             porta=args.porta,
645             baud=args.baud,
646             duracao_s=args.duracao,
647             prefixo=args.prefixo,
648         )
649         coletar_csv(cfg, Path(args.outdir))
650         return
651
652     if args.cmd == "calibrar":
653         calibrar_baseline(
654             pasta_csvs=Path(args.pasta_normais),
655             out_baseline=Path(args.baseline_out),
656             base_s=args.base_s,
657             win_s=args.win_s,
658             hop_s=args.hop_s,
659             m=args.m,
660         )
661         return
662
663     if args.cmd == "avaliar":
664         avaliar_csv(
665             csv_path=Path(args.csv),
666             baseline_path=Path(args.baseline),
667             base_s=args.base_s,
668             win_s=args.win_s,
669             hop_s=args.hop_s,
670             m=args.m,
```

```
671         lim_normal=args.lim_normal,
672         lim_anom=args.lim_anom,
673         lim_pico=args.lim_pico,
674         topk=args.topk,
675     )
676     return
677
678 if args.cmd == "run":
679     cfg = SerialConfig(
680         porta=args.porta,
681         baud=args.baud,
682         duracao_s=args.duracao,
683         prefixo=args.prefixo,
684     )
685     csv_path = coletar_csv(cfg, Path(args.outdir))
686     avaliar_csv(
687         csv_path=csv_path,
688         baseline_path=Path(args.baseline),
689         base_s=args.base_s,
690         win_s=args.win_s,
691         hop_s=args.hop_s,
692         m=args.m,
693         lim_normal=args.lim_normal,
694         lim_anom=args.lim_anom,
695         lim_pico=args.lim_pico,
696         topk=args.topk,
697     )
698     return
699
700
701 if __name__ == "__main__":
702     main()
```

APÊNDICE B – CÓDIGO-FONTE DO FIRMWARE DE AQUISIÇÃO (ESP32)

Este apêndice apresenta o firmware desenvolvido em C para o microcontrolador ESP32, responsável pela aquisição das amostras do ADC (sinal analógico do ACS712) e pela transmissão serial dos dados no formato t,adc. O protocolo simples utilizado inclui os marcadores #READY, #START e #END, além do comando START, <segundos> enviado pelo computador.

Código-fonte 2 – Firmware de aquisição no ESP32 (arquivo coleta_acs712.c).

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <ctype.h>
5
6  #include "freertos/FreeRTOS.h"
7  #include "freertos/task.h"
8
9  #include "esp_rom_sys.h"
10 #include "esp_log.h"
11 #include "esp_timer.h"
12
13 #include "driver/uart.h"
14 #include "esp_err.h"
15
16 // ADC (driver novo)
17 #include "esp_adc/adc_oneshot.h"
18
19 static const char *TAG = "COLETA";
20
21 // UART0 (USB Serial)
22 #define UART_PORT          UART_NUM_0
23 #define UART_BUF_SIZE      256
24
25 // GPIO32 no ESP32 = ADC1_CHANNEL_4

```

```

26 #define ADC_UNIT_ID      ADC_UNIT_1
27 #define ADC_CHANNEL      ADC_CHANNEL_4
28
29 // Amostragem (alvo; na pr tica o printf pode limitar)
30 #define FS_HZ             1000
31 #define DT_US             (1000000 / FS_HZ)
32
33 // ----- M DIA / OVERSAMPLING -----
34 #define AVG_N             16          // 8, 16, 32... (quanto
    maior, mais liso)
35 #define INNER_DELAY_US 40          // atraso entre leituras
    internas (us). 0..100 costuma ok
36 // -----
37
38 // l  uma linha da UART (at  '\n'), retorna tamanho;
    coloca '\0' no final
39 static int uart_readline(char *out, int max_len, int
    timeout_ms) {
40     int idx = 0;
41     uint8_t ch;
42
43     int tleft = timeout_ms;
44     while (idx < max_len - 1) {
45         int n = uart_read_bytes(UART_PORT, &ch, 1,
            pdMS_TO_TICKS(10));
46         tleft -= 10;
47         if (n <= 0) {
48             if (tleft <= 0) break;
49             continue;
50         }
51         if (ch == '\r') continue;
52         if (ch == '\n') break;

```

```

53         out[idx++] = (char)ch;
54     }
55     out[idx] = '\0';
56     return idx;
57 }
58
59 static void str_upper(char *s) {
60     for (; *s; s++) *s = (char)toupper((unsigned char)*s);
61 }
62
63 // ----- NOVO: leitura com m dia -----
64 static int adc_read_avg(adc_one_shot_unit_handle_t
65     adc_handle) {
66     int acc = 0;
67
68     for (int i = 0; i < AVG_N; i++) {
69         int raw = 0;
70         esp_err_t err = adc_one_shot_read(adc_handle,
71             ADC_CHANNEL, &raw);
72         if (err != ESP_OK) return -1;
73
74         acc += raw;
75
76         if (INNER_DELAY_US > 0) {
77             esp_rom_delay_us(INNER_DELAY_US);
78         }
79     }
80
81     return acc / AVG_N;
82 }
83
84 // -----

```

```

83 // aquisi o: envia #START, dados e #END
84 static void do_acquire(adc_oneshot_unit_handle_t adc_handle
    , int seconds) {
85     printf("#START\n");
86     printf("t,adc\n");
87
88     int64_t t0 = esp_timer_get_time();
89     int64_t total_us = (int64_t)seconds * 1000000LL;
90
91     while ((esp_timer_get_time() - t0) < total_us) {
92         int raw = adc_read_avg(adc_handle);
93         if (raw < 0) {
94             printf("#ERR,ADC_READ\n");
95             break;
96         }
97
98         float t_s = (float)(esp_timer_get_time() - t0) /
            1000000.0f;
99         printf("%.6f,%d\n", t_s, raw);
100
101         // tentativa de 1 kHz (mas o printf pode limitar o
            throughput real)
102         esp_rom_delay_us(DT_US);
103     }
104
105     printf("#END\n");
106 }
107
108 void app_main(void) {
109     // 1) UART (USB Serial)
110     const uart_config_t uart_cfg = {
111         .baud_rate = 115200,

```



```

112     .data_bits = UART_DATA_8_BITS,
113     .parity    = UART_PARITY_DISABLE,
114     .stop_bits = UART_STOP_BITS_1,
115     .flow_ctrl = UART_HW_FLOWCTRL_DISABLE
116 };
117 uart_param_config(UART_PORT, &uart_cfg);
118 uart_driver_install(UART_PORT, 2048, 0, 0, NULL, 0);
119
120 // 2) ADC oneshot
121 adc_oneshot_unit_init_cfg_t unit_cfg = {
122     .unit_id = ADC_UNIT_ID,
123     .ulp_mode = ADC_ULP_MODE_DISABLE
124 };
125 adc_oneshot_unit_handle_t adc_handle;
126 ESP_ERROR_CHECK(adc_oneshot_new_unit(&unit_cfg, &
    adc_handle));
127
128 adc_oneshot_chan_cfg_t chan_cfg = {
129     .atten = ADC_ATTEN_DB_12,      // DB_11 est
    deprecated (DB_12      equivalente)
130     .bitwidth = ADC_BITWIDTH_12
131 };
132 ESP_ERROR_CHECK(adc_oneshot_config_channel(adc_handle,
    ADC_CHANNEL, &chan_cfg));
133
134 printf("#READY\n");
135 printf("#INFO,Send START,<seconds>\n");
136 printf("#INFO,AVG_N=%d,INNER_DELAY_US=%d,DT_US=%d\n",
    AVG_N, INNER_DELAY_US, DT_US);
137
138 char line[UART_BUF_SIZE];
139

```

```

140 while (1) {
141     int n = uart_readline(line, sizeof(line), 1000);
142     if (n <= 0) {
143         vTaskDelay(pdMS_TO_TICKS(10));
144         continue;
145     }
146
147     char cmd[UART_BUF_SIZE];
148     strncpy(cmd, line, sizeof(cmd));
149     cmd[sizeof(cmd) - 1] = '\0';
150     str_upper(cmd);
151
152     if (strncmp(cmd, "START", 5) == 0) {
153         int seconds = 30;
154         char *comma = strchr(cmd, ',');
155         if (comma) {
156             int s = atoi(comma + 1);
157             if (s > 0) seconds = s;
158         }
159         ESP_LOGI(TAG, "Starting acquisition for %d
160                     seconds", seconds);
161         do_acquire(adc_handle, seconds);
162     } else {
163         printf("#ERR, UNKNOWN_CMD\n");
164     }
165 }

```