

Processamento de Sinais para Arrays de Antenas Utilizando Rádio Definido por Software e Multithread

Renan D. Selli, Candice Müller e Fernando C. C. de Castro

Resumo—O uso de múltiplos rádios definidos por software (SDRs – *Software Defined Radios*) de baixo custo em *arrays* de antenas com mais de dois elementos impõe desafios significativos de sincronização entre dispositivos SDRs, essenciais para garantir a coerência de fase. Quando múltiplos dispositivos são operados de forma independente, diferenças de temporização e aquisição podem comprometer o alinhamento temporal dos sinais, inviabilizando o processamento espacial coerente. Neste contexto, este artigo propõe uma abordagem *multithread* para digitalização e processamento de sinais em *arrays* de antenas utilizando SDRs de baixo custo. A solução emprega *threads* coordenadas por primitivas POSIX para realizar o processamento paralelo das amostras IQ dos múltiplos canais, seguida por uma cadeia de processos responsável pelo alinhamento temporal entre SDRs por meio de correlação cruzada. A abordagem proposta foi implementada e validada experimentalmente em duas etapas. Na primeira etapa, a abordagem e os algoritmos foram avaliados e validados. Na segunda etapa, foi estabelecido um enlace de comunicação no qual o sinal transmitido é recebido por um *array* composto por quatro dipolos, cujas saídas são conectadas a dois SDRs RSPduo. Os resultados experimentais demonstram a eficácia da solução proposta, permitindo a sincronização temporal dos quatro sinais demodulados na saída da cadeia de processamento, com MER (*Modulation Error Ratio*) superior a 24 dB e SER (*Symbol Error Rate*) nula.

Palavras-chave: Rádio Definido por *Software*, *Multithread*, *Array* de Antenas, Sincronismo.

Abstract—The use of multiple low-cost software-defined radios (SDRs) in antenna arrays with more than two elements imposes significant synchronization challenges among SDR devices, which are essential to ensure phase coherence. When multiple devices operate independently, timing and acquisition differences can compromise the temporal alignment of the signals, preventing coherent spatial processing. In this context, this paper proposes a multithread approach for signal digitization and processing in antenna arrays using low-cost SDRs. The solution employs threads coordinated by POSIX primitives to perform the parallel processing of IQ samples from the multiple channels, followed by a processing chain responsible for temporal alignment among SDRs via cross-correlation. The proposed approach was implemented and experimentally validated in two stages. In the first stage, the approach and the algorithms were evaluated and validated. In the second stage, a communication link was established where the transmitted signal is received by an array composed of four dipoles, whose outputs are connected to two RSPduo SDRs. Experimental results demonstrate the effectiveness of the

proposed solution, enabling the temporal synchronization of the four demodulated signals at the output of the processing chain, with a MER (*Modulation Error Ratio*) higher than 24 dB and a zero SER (*Symbol Error Rate*).

Keywords—Software-Defined Radio, Multithreading, Antenna Arrays, Synchronization.

I. INTRODUÇÃO

A crescente demanda por altas taxas de dados, eficiência espectral e comunicação sem fio confiável tem impulsionado avanços significativos nas técnicas de processamento de sinais e arquiteturas de hardware. Nesse contexto, os *array* de antenas emergiram como uma tecnologia fundamental, permitindo filtragem espacial, formação de feixes (*beamforming*) e estimativa de direção de chegada, que são essenciais para os modernos sistemas de comunicação e sensoriamento [1]. Essas capacidades são imprescindíveis em uma ampla gama de aplicações, incluindo comunicações sem fio, sistemas de radar e paradigmas emergentes, como as redes de quinta geração (5G) e além [2].

O processamento de sinais para *arrays* de antenas, no entanto, impõe requisitos computacionais consideráveis, particularmente quando o processamento em tempo real é exigido. A necessidade de lidar com múltiplos fluxos de dados simultaneamente, combinada com algoritmos cada vez mais complexos, exige soluções de processamento eficientes e escaláveis. As implementações tradicionais de *thread* única frequentemente falham em explorar ao máximo as capacidades dos modernos processadores *multi-core*, levando a gargalos de desempenho e limitando a escalabilidade do sistema [3], [4].

Paralelamente, o rádio definido por *software* (SDR) tem ganhado destaque como uma plataforma flexível e de bom custo-benefício para a prototipagem e implementação de sistemas avançados de comunicação. Ao transferir as tarefas de processamento de sinais de hardwares dedicados para software, o SDR possibilita o desenvolvimento rápido, a reconfigurabilidade e a experimentação [3]. No entanto, alcançar um processamento de alto desempenho em sistemas baseados em SDR continua sendo um desafio, especialmente em aplicações que envolvem *arrays* de antenas, nas quais múltiplos sinais devem ser processados simultaneamente [5].

Em sistemas de *arrays* de antenas, a sincronização precisa dos sinais é um requisito fundamental; caso contrário, o próprio princípio de operação em fase falha. A formação de feixes e a estimativa de direção de chegada dependem de

relações de fase precisas entre os sinais recebidos ou transmitidos por cada elemento da antena [5]. Em implementações práticas, isso significa que todos os canais (ou sinais) devem estar sincronizados. Sem essa sincronização, o desempenho do *array* é comprometido.

Nesse contexto, este artigo propõe uma abordagem *multithread* para implementação de um receptor digital no âmbito do processamento de sinais necessário para *arrays* de antenas. A abordagem utiliza processamento paralelo para executar simultaneamente as rotinas de recepção e processamento associadas aos diferentes SDRs, reduzindo o risco de perda de amostras durante a aquisição. A abordagem é implementada e validada experimentalmente utilizando um *array* de antenas de quatro dipolos, demonstrando sua eficácia e aplicabilidade prática.

II. Array DE ANTENAS

Um *array* de antenas é uma estrutura formada pela união de N elementos irradiantes dispostos geometricamente de modo a produzir um padrão de irradiação desejado. O padrão de radiação desejado é obtido pela interferência construtiva e destrutiva entre os sinais nos elementos, cujo resultado depende da geometria do *array*, do espaçamento entre elementos, da amplitude e fase dos sinais em cada elemento [7]. Essas propriedades habilitam aplicações como estimação de direção de chegada (*Direction of Arrival*, DOA), direcionamento dinâmico de feixe (*beamforming*), múltiplo acesso espacial (SDMA) e, em sistemas 5G, o emprego de *arrays* MIMO massivos (*Massive MIMO*) para atendimento simultâneo de múltiplos usuários [1], [2].

A capacidade de direcionamento de feixe de um *array* de antenas é alcançada por meio de técnicas de *beamforming*. As três arquiteturas clássicas de *beamforming* existentes na literatura podem ser classificadas como analógica, digital e híbrida. No *beamforming* analógico, os ajustes de fase e amplitude dos sinais são realizados diretamente em RF utilizando deslocadores de fase e atenuadores, antes da combinação dos sinais. O *beamforming* digital realiza o processamento em tempo real do sinal em banda base após a recepção individual de cada elemento. Essa abordagem permite a implementação de diferentes técnicas; no entanto, exige uma cadeia de RF para cada elemento do *array*, o que torna o sistema mais custoso. Na arquitetura híbrida, um estágio analógico, que agrupa subconjuntos de antenas em subarranjos, é combinado com um estágio digital em banda base, reduzindo o número de cadeias de RF necessárias [2]. Essa abordagem é amplamente empregada em sistemas *Massive MIMO* e comunicações de ondas milimétricas (mmWave).

III. ABORDAGEM Multithread UTILIZANDO RADIO DEFINIDO POR Software

Um SDR é uma arquitetura de rádio na qual funções tradicionalmente implementadas em *hardware* dedicado são realizadas em *software* [3]. Essa abordagem garante flexibilidade ao sistema, permitindo a reconfiguração de parâmetros como frequência de operação, largura de banda, esquema de modulação, dentre outros, sem alterações no *hardware* físico.

Dispositivos SDR de baixo custo, contudo, impõem uma limitação estrutural para aplicações em *arrays*: cada dispositivo oferece, tipicamente, no máximo dois canais de recepção simultânea. Assim, para qualquer *array* com $M > 2$ elementos, essa restrição implica na necessidade de $\lceil M/2 \rceil$ SDRs independentes operando de maneira sincronizada. Os dados brutos recebidos por cada SDR são acessados através de uma interface de programação de aplicações (API, do inglês *Application Programming Interface*). A API é um conjunto padronizado de funções e regras que permite a um programa acessar e controlar recursos disponibilizados por outro software ou dispositivo. Usualmente, a API é fornecida pelo fabricante do hardware e atua como interface entre o programa rodando no computador e o SDR. Desta forma, permite configurar o dispositivo, iniciar a aquisição e receber dados brutos por meio de *callbacks*, *buffers* ou mecanismos equivalentes, dentre outras funcionalidades, sem que o programa precise lidar diretamente com os detalhes de baixo nível do *hardware*.

O problema principal, além da necessidade de múltiplos dispositivos, é garantir coerência entre os sinais recebidos pelos M canais dos SDRs. Na abordagem direta, a utilização de processos separados para controlar cada SDR introduz atrasos na ativação dos dispositivos e no início das leituras dos *streams* de dados de cada canal. Esta abordagem, na qual cada SDR é controlado por um programa diferente, gera processos independentes, comprometendo a coerência intercanal, comprometendo o funcionamento do sistema. Nessa condição, os instantes de início da captura podem ocorrer em momentos distintos, e as relações de fase entre os canais tornam-se indefinidas, inviabilizando qualquer processamento coerente [6], [8]. Em abordagens *single-thread*, a execução sequencial do processamento digital de sinais dentro do próprio fluxo de captura introduz latências que frequentemente superam o tempo de preenchimento dos *buffers* do rádio, causando o descarte de amostras (*buffer overflow*). A perda não compensada de amostras, além de acarretar em perda de sincronismo, pode introduzir um deslocamento temporal entre os canais, comprometendo a coerência temporal e de fase necessária ao processamento espacial. Logo, o paralelismo via *multithread* não é apenas uma otimização de desempenho, mas uma condição estritamente necessária para a viabilidade do sistema com múltiplos SDRs.

Neste contexto, este trabalho propõe o uso de *multithreading* para a recepção dos sinais provenientes dos canais dos $\lceil M/2 \rceil$ SDRs conectados aos elementos do *array*. Cada *thread* configura, inicializa, gerencia e captura continuamente as amostras IQ (*In-Phase and Quadrature*) recebidas pelo respectivo SDR do conjunto de SDRs que compõem o sistema.

A coordenação entre as *threads* é implementada com primitivas POSIX de exclusão mútua (*mutex*) e variáveis de condição. Na prática, para um *array* com M elementos, o sistema utiliza $N = \lceil M/2 \rceil$ *threads*, cada uma responsável por alocar um dos N dispositivos SDR e gerenciar os respectivos *callbacks* de recepção, correspondentes a até dois canais por dispositivo. A inicialização ocorre de forma coordenada: cada *thread* configura seu SDR e, por meio das variáveis de condição protegidas por *mutex*, sinaliza a conclusão dessa

etapa. As *threads* permanecem bloqueadas até que todos os N dispositivos estejam configurados; então, o conjunto é liberado para iniciar a recepção de forma sincronizada, evitando conflitos na alocação de recursos da API. Estruturas globais, como os *buffers* de correção e *flags* de estado (*buffer_ready*), são compartilhadas. Quando os *callbacks* preenchem os *buffers* com a quantidade necessária de amostras demoduladas, as *flags* sinalizam aos laços principais de cada *thread* para executar o cálculo de alinhamento temporal (correlação cruzada) fora das rotinas de interrupção de *hardware*, prevenindo perda de amostras [4].

Uma vez recebidas, as amostras IQ de cada canal são submetidas a uma cadeia de processamento de sinais idêntica. A Fig. 1 apresenta a cadeia de processamento, implementada em cada *thread*. Os dados brutos adquiridos via *thread* são inicialmente filtrados por um filtro de cosseno elevado de raiz quadrada (RRC) [9] e, na sequência, o número de amostras por símbolo IQ é reduzido para duas amostras. Após esta etapa do processamento, os símbolos encontram-se dispersos e rotacionados devido à falta de sincronismo tanto de tempo quanto de portadora. Para tratar o sincronismo de tempo, o algoritmo de Gardner-Farrow [10] é implementado, seguido pelo recuperador de portadora, sendo adotado neste trabalho o algoritmo Costas-Loop [11].

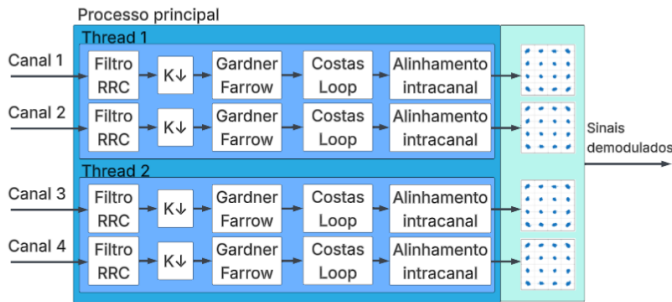


Fig. 1. Cadeia de processamento dos dados implementada em cada *thread*.

Na saída do Costas-Loop, os sinais demodulados ainda podem conter deslocamentos entre canais. Tal problema ocorre mesmo com a inicialização coordenada pelas *threads* e decorre da diferença nos instantes efetivos da captura dos símbolos IQs em cada canal. Esta diferença nos instantes de amostragem dos múltiplos canais é suficiente para destruir a coerência de fase entre os elementos do *array* [6].

Visando possibilitar o processamento espacial do sinal, é necessário realizar o alinhamento intercanal, i.e., entre SDRs. Tal ajuste é feito através da correlação cruzada entre uma sequência de referência conhecida e os sinais recebidos em cada canal. O deslocamento temporal δ_k do k -ésimo canal é dado por:

$$\delta_k = \arg \max_{\tau} \left| \sum_{n=0}^{N-1} x_{\text{ref}}(n) x_k^*(n - \tau) \right| \quad (1)$$

em que $x_{\text{ref}}(n)$ representa a sequência de referência conhecida, $x_k(n)$ é o sinal demodulado do k -ésimo canal, sendo $k = 1, 2, \dots, M$, n é o índice de amostra, τ é o atraso e $(\cdot)^*$

denota o conjugado complexo. O operador $\arg \max$ retorna o valor de τ que maximiza o módulo da correlação cruzada.

Uma vez obtido o valor de deslocamento δ_k dos M canais, os *buffers* que armazenam os sinais são alinhados, garantindo coerência temporal entre os M fluxos de símbolos IQs. Na prática, o demodulador de cada canal descarta as amostras iniciais e começa a gravar a partir do ponto onde o sinal recebido coincide com a sequência de referência[1], [12].

IV. RESULTADOS EXPERIMENTAIS

A abordagem *multithread* SDR proposta foi implementada e validada em um *array* de quatro elementos. Para a recepção dos sinais do sistema *quad-channel* foram utilizados dois SDRs RSPduo da SDRplay [13], cada um operando em modo *Dual Tuner*, totalizando quatro canais independentes. O sistema opera na frequência de $f = 1.0$ GHz, adota modulação 16-QAM, taxa de símbolo de 250 kbps, taxa de amostragem de 2 MHz e fator de superamostragem $k = 8$. Os demoduladores são implementados em linguagem C, executados em sistema UNIX, e utilizam a API da SDRplay com POSIX Threads para a captura paralela dos quatro canais, conforme descrito na seção anterior.

O sistema foi validado em duas etapas experimentais distintas. A Etapa 1 teve como objetivo validar a abordagem. Para tal, o sinal 16-QAM em 1 GHz foi gerado pelo gerador vetorial de sinais e aplicado à entrada de um divisor de potência de Wilkinson 1:4. As saídas do divisor foram conectadas às entradas de dois SDRs, conforme ilustra a Fig. 2. Esta etapa permitiu validar a abordagem *multithread* SDR proposta e os algoritmos de demodulação e sincronização.

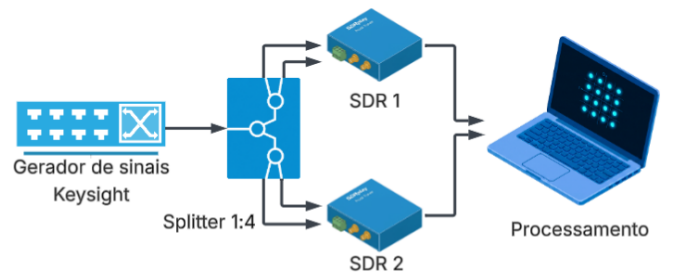


Fig. 2. Setup da Etapa 1: validação da abordagem *multithread* SDR proposta e dos algoritmos de demodulação e sincronização.

A Fig. 3 apresenta as constelações obtidas nos quatro canais, denotados por Canal A, Canal B, Canal C e Canal D, após a cadeia de processamento (ver Fig. 1). Após a convergência dos algoritmos de sincronismo, todos os canais apresentam constelações com MER superior a 26 dB, SER nula e deslocamento nulo entre os sinais intercanal, validando a implementação da abordagem *multithread* e dos algoritmos da cadeia de processamento dos dados recebidos.

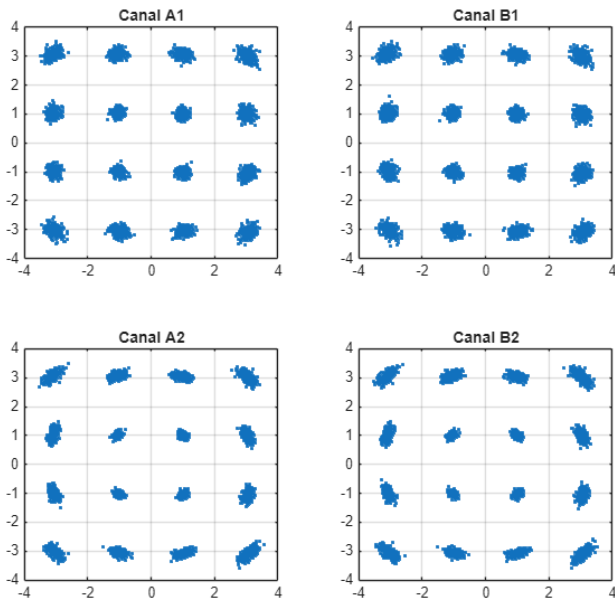


Fig. 3. Etapa 1: Constelações resultantes nos quatro canais utilizando a abordagem *multithread* e os algoritmos da cadeia de processamento após a convergência dos algoritmos de sincronismo de tempo e de frequência.

A Etapa 2 visa avaliar a solução proposta em situação real de operação. Nesta etapa, o sinal gerado pelo gerador vetorial de sinais alimenta uma antena refletor de canto, conforme mostra a Fig. 4. O sinal transmitido pelo refletor de canto se propaga até o *array* de antenas de quatro dipolos linearmente espaçadas, que possui as suas saídas conectadas à entrada de cada um dos canais dos SDRs. O gerador vetorial de sinais foi configurado com potência de -5 dBm, e o refletor de canto foi posicionado a aproximadamente 3m do *array*. A Fig. 5 mostra a *setup* experimental utilizado nas medidas.

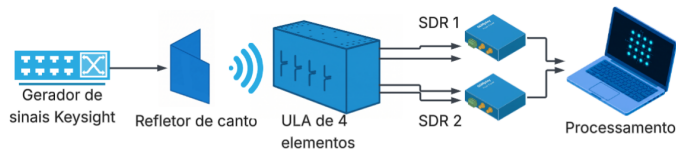


Fig. 4. *Setup* da Etapa 2: validação da abordagem *multithread* proposta em situação real, utilizando um *array* de quatro elementos.



Fig. 5. *Setup* experimental utilizado nas medidas da Etapa 2

A Figura 6 apresenta a constelação dos sinais recebidos pelos quatro elementos do *array*. Em todos os quatro canais,

a SER foi igual a zero, enquanto a MER permaneceu superior a 24 dB. Ressalta-se que tais resultados indicam o sucesso da recuperação individual da portadora em cada canal. No entanto, a comprovação do alinhamento temporal e espacial é atestada pela correlação cruzada, que resultou em um deslocamento relativo intercanal nulo ($\delta_k = 0$) após a compensação, validando a abordagem proposta em um sistema real.

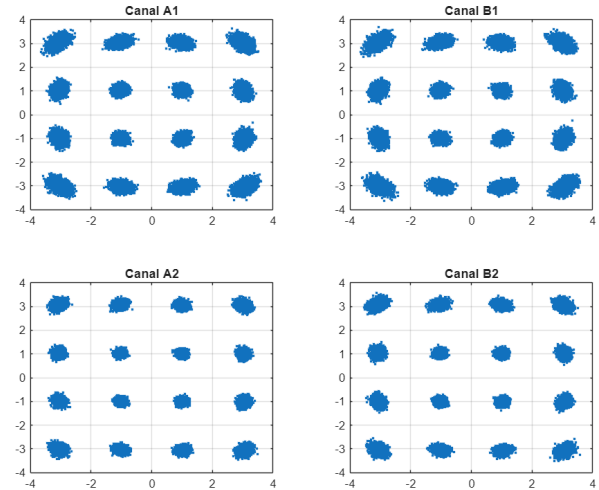


Fig. 6. Etapa 2: Constelações resultantes nos quatro canais utilizando a abordagem *multithread* e os algoritmos da cadeia de processamento obtidas na medida prática.

Os resultados obtidos mostram que a abordagem *multithreaded* proposta fornece, na saída da cadeia de processamento de sinais, os sinais dos quatro canais do *array* sincronizados e sem o problema de perda de informação, presente quando múltiplos processos são utilizados para controlar os múltiplos SDRs. Os sinais demodulados possuem coerência temporal, permitindo a aplicação de técnicas de pós-processamento, como técnicas de *beamforming* e de *direction of arrival* (DOA).

V. CONCLUSÕES

Este trabalho propôs o uso da arquitetura *multithreaded* na recepção de sinais provenientes de *arrays* de antenas, utilizando rádio definido por *software*. A abordagem proposta permite compensar o desalinhamento temporal intercanal, i.e., entre SDRs, presente na abordagem direta, onde cada SDR é controlado por seu respectivo processo.

A inicialização coordenada dos SDRs usando *multithread* e o sincronismo temporal intercanal, com base no deslocamento temporal obtido pela correlação cruzada, mantém a coerência temporal entre os múltiplos sinais recebidos, possibilitando a implementação de técnicas de pós-processamento, tais como formação de feixe e estimativa de direção de chegada.

A abordagem proposta foi validada através de medidas experimentais, em duas etapas. Na primeira etapa, foi validada a abordagem proposta e os algoritmos desenvolvidos. Na segunda etapa, a solução proposta foi validada em situação real, no qual um enlace de comunicação 16-QAM foi estabelecido entre o transmissor e o receptor, este composto por um *array* de quatro antenas dipolo linearmente espaçadas e dois SDRs. Os resultados experimentais confirmaram o sincronismo e a

integridade dos sinais recebidos. Os sinais demodulados dos quatro canais, corretamente sincronizados, alcançaram MER superior a 24 dB e SER nula.

Como continuidade deste trabalho, pretende-se expandir a cadeia de processamento para incluir algoritmos de *beamforming* e estimativa de direção de chegada (DOA) em cenários reais, avaliando a robustez do receptor desenvolvido na presença de sinais interferentes e em condições de baixa relação sinal-ruído (SNR).

AGRADECIMENTOS

Este trabalho foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico, Chamada CNPq/MCTI/FNDCT N° 22/2022 - Programa MCTI de Inovação em Grafeno, InovaGrafeno-MCTI, processo 408423/2022-06.

REFERÊNCIAS

- [1] L. C. Godara, "Application of antenna arrays to mobile communications, Part II: Beam-forming and direction-of-arrival considerations," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195–1245, Ago. 1997.
- [2] R. M. Dreifuerst e R. W. Heath Jr., "Massive MIMO in 5G: How beamforming, codebooks, and feedback enable larger arrays," *arXiv preprint arXiv:2301.13390*, 2023.
- [3] R. Akeela e B. Dezfouli, "Software-defined radios: Architecture, state-of-the-art, and challenges," *Computer Communications*, vol. 128, pp. 106–125, 2018.
- [4] D. R. Butenhof, *Programming with POSIX threads*. Boston, MA, EUA: Addison-Wesley Professional, 1997.
- [5] C. Müller e F. C. C. de Castro, "Adaptive Blind Antenna Array Beam Shaping," em *Anais do XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrt 2025)*, Natal, RN, 2025.
- [6] M. Krueckemeier, F. Schwartz, C. Monka-Ewe e J. Schoebel, "Synchronization of multiple USRP SDRs for coherent receiver applications," em *2019 Sixth International Conference on Software Defined Systems (SDS)*, 2019, pp. 58–63.
- [7] C. A. Balanis, *Antenna Theory: Analysis and Design*, 4ª ed. Hoboken, NJ, EUA: John Wiley & Sons, 2016.
- [8] M. Rashid e J. A. Nanzer, "Frequency and Phase Synchronization in Distributed Antenna Arrays Based on Consensus Averaging and Kalman Filtering," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 9, pp. 8103–8116, Set. 2022.
- [9] J. G. Proakis e M. Salehi, *Digital Communications*, 5ª ed. Nova Iorque, NY, EUA: McGraw-Hill, 2008.
- [10] F. M. Gardner, "A BPSK/QPSK Timing-Error Detector for Sampled Receivers," *IEEE Transactions on Communications*, vol. 34, no. 5, pp. 423–429, Maio 1986.
- [11] M. Rice, *Digital Communications: A Discrete-Time Approach*. Upper Saddle River, NJ, EUA: Pearson/Prentice Hall, 2008.
- [12] C. H. Knapp e G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Ago. 1976.
- [13] SDRplay, *RSPduo Dual Tuner 14-bit SDR*. Datasheet, SDRplay, 2017.
- [14] Keysight Technologies, *EXG X-Series RF Vector Signal Generator N5172B*. Keysight Technologies, 2026.