

CONTROLE PID EM HARDWARE RECONFIGURÁVEL APLICADO À ESTABILIZAÇÃO DE UM PÊNDULO INVERTIDO

Tharick Pontes Ferreira¹, Guilherme Miyata Meirelles², Giovanna Rabello³, Alessandro de Lima⁴, Anderson Kenji Hirata⁵ e Valdeci Donizete Gonçalves⁶

Resumo

O pêndulo invertido constitui um sistema dinâmico clássico caracterizado por instabilidade, não linearidade e alta sensibilidade a atrasos na ação de controle, sendo amplamente utilizado para a validação de estratégias de controle em tempo real. Embora plataformas baseadas em microcontroladores sejam frequentemente empregadas nesse tipo de aplicação devido ao baixo custo e à facilidade de implementação, essas soluções apresentam limitações relacionadas ao determinismo temporal, à latência e ao *jitter*. Neste trabalho, é proposta a implementação de uma arquitetura híbrida de controle em tempo real para a estabilização de um pêndulo invertido sobre carrinho, utilizando uma *Field Programmable Gate Array* (FPGA) como núcleo de processamento do controle. Nessa arquitetura, um microcontrolador Arduino é responsável pela aquisição e pelo pré-processamento dos sinais dos sensores, enquanto a FPGA executa o controlador Proporcional-Integral-Derivativo (PID), a geração do sinal PWM e a lógica de acionamento do atuador, assegurando comportamento determinístico da malha de controle. A implementação foi realizada de forma modular em linguagem VHDL, utilizando aritmética inteira escalonada. O sistema desenvolvido foi avaliado quanto ao seu comportamento temporal e à previsibilidade da execução do controle em tempo real. Os resultados experimentais indicam que a utilização de hardware reconfigurável permite maior controle sobre a temporização da malha, evidenciando o potencial de arquiteturas híbridas FPGA – microcontrolador em aplicações de controle de sistemas rápidos e instáveis.

Palavras-chave: FPGA; Sistemas em tempo real; Hardware reconfigurável; Sistemas embarcados; Controle digital.

¹ Graduando em Engenharia de Controle e Automação, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus de São José dos Campos, Estudante, tharick.pontes@aluno.ifsp.edu.br.

² Graduando em Engenharia de Controle e Automação, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus de São José dos Campos, Estudante, g.miyata@aluno.ifsp.edu.br.

³ Graduando em Engenharia de Controle e Automação, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus de São José dos Campos, Estudante, giovanna.rabello@aluno.ifsp.edu.br.

⁴ Graduando em Engenharia de Controle e Automação, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus de São José dos Campos, Estudante, l.alessandro@aluno.ifsp.edu.br.

⁵ Mestre em Ciências e Tecnologias Espaciais, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus de São José dos Campos, Professor EBTT, anderson.hirata@ifsp.edu.br.

⁶ Doutor em Engenharia, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus de São José dos Campos, Professor EBTT, valdecidgoncalves@ifsp.edu.br.

PID CONTROL ON RECONFIGURABLE HARDWARE APPLIED TO INVERTED PENDULUM STABILIZATION

Abstract

The inverted pendulum is a classical dynamic system characterized by instability, nonlinearity, and high sensitivity to delays in control action, being widely used for the validation of real-time control strategies. Although microcontroller-based platforms are commonly employed in such applications due to their low cost and ease of implementation, these solutions present limitations related to temporal determinism, latency, and jitter. In this work, a hybrid real-time control architecture is proposed for the stabilization of an inverted pendulum on a cart, using a Field Programmable Gate Array (FPGA) as the core processing unit for control tasks. In the proposed architecture, an Arduino microcontroller is responsible for sensor data acquisition and signal pre-processing, while the FPGA executes the Proportional-Integral-Derivative (PID) controller, PWM signal generation, and actuator control logic, ensuring deterministic execution of the control loop. The implementation was carried out in a modular manner using VHDL and scaled integer arithmetic. The developed system was evaluated with respect to its temporal behavior and execution predictability in real-time operation. Experimental results indicate that the use of reconfigurable hardware provides greater control over loop timing, highlighting the potential of hybrid FPGA - microcontroller architectures for control applications involving fast and unstable systems.

Keywords: FPGA; Real-time systems; Reconfigurable hardware; Embedded systems; Digital control.

CONTROL PID EN HARDWARE RECONFIGURABLE APLICADO A LA ESTABILIZACIÓN DE UN PÉNDULO INVERTIDO

Resumen

El péndulo invertido constituye un sistema dinámico clásico caracterizado por inestabilidad, no linealidad y alta sensibilidad a retardos en la acción de control, siendo ampliamente utilizado para la validación de estrategias de control en tiempo real. Aunque las plataformas basadas en microcontroladores son comúnmente empleadas en este tipo de aplicaciones debido a su bajo costo y facilidad de implementación, dichas soluciones presentan limitaciones relacionadas con el determinismo temporal, la latencia y el *jitter*.

En este trabajo se propone una arquitectura híbrida de control en tiempo real para la estabilización de un péndulo invertido sobre carro, utilizando una *Field Programmable Gate Array* (FPGA) como núcleo de procesamiento del control. En la arquitectura propuesta, un microcontrolador Arduino se encarga de la adquisición y el preprocesamiento de las señales de los sensores, mientras que la FPGA ejecuta el controlador Proporcional-Integral-Derivativo (PID), la generación de la señal PWM y la lógica de accionamiento del actuador, garantizando una ejecución determinista de la malha de control.

La implementación se realizó de manera modular en lenguaje VHDL, utilizando aritmética entera escalonada. El sistema desarrollado fue evaluado en cuanto a su comportamiento temporal y a la previsibilidad de la ejecución en tiempo real. Los resultados experimentales indican que el uso de hardware reconfigurable permite un mayor control sobre la temporización

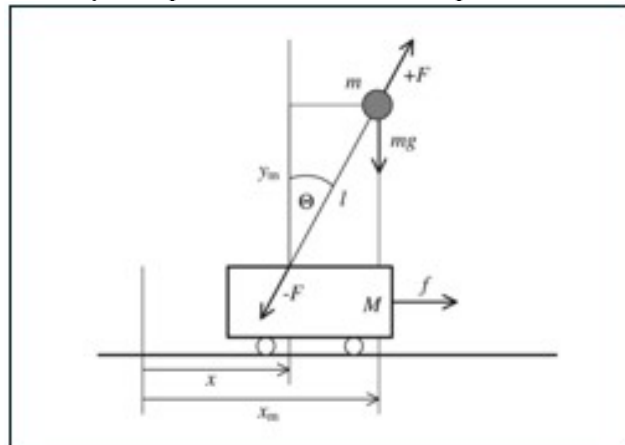
del lazo, evidenciando el potencial de arquitecturas híbridas FPGA–microcontrolador en aplicaciones de control de sistemas rápidos e inestables.

Palabras-clave: FPGA; Sistemas en tiempo real; Hardware reconfigurable; Sistemas embebidos; Control digital.

Introdução

O pêndulo invertido (Figura 1) é amplamente reconhecido na literatura como um sistema dinâmico instável, não linear e subatuado (HERNÁNDEZ-GUZMÁN; SILVA-ORTIGOZA, 2019; OGATA, 2011). Em razão dessas características, é frequentemente utilizado como sistema de referência para o estudo e a validação de estratégias de controle, uma vez que pequenas perturbações ou atrasos na ação de controle podem conduzir rapidamente à perda de estabilidade, tornando-o um problema clássico na avaliação de técnicas de controle em tempo real (HERNÁNDEZ-GUZMÁN; SILVA-ORTIGOZA, 2019).

Figura 1: Representação esquemática do sistema de pêndulo invertido sobre carro.



Fonte: Keviczky, et al. (2019)

Tradicionalmente, plataformas baseadas em microcontroladores (MCUs) são amplamente empregadas na implementação de controladores aplicados ao pêndulo invertido, em função do baixo custo, ampla disponibilidade e facilidade de programação em linguagens de alto nível (RAO, 2025). Entretanto, essas plataformas apresentam limitações relevantes em aplicações que demandam determinismo temporal, baixa latência e execução simultânea de múltiplas tarefas. A arquitetura sequencial dos microcontroladores, aliada à dependência de interrupções e ciclos de processamento em software, pode introduzir atrasos variáveis e *jitter*, comprometendo o desempenho do controle, especialmente em sistemas rápidos e intrinsecamente instáveis (RAO, 2025).

Nesse contexto, as *Field Programmable Gate Arrays* (FPGAs) surgem como uma alternativa promissora para aplicações de controle em tempo real. Diferentemente dos

microcontroladores convencionais, as FPGAs permitem a implementação de algoritmos diretamente em hardware, explorando paralelismo, alto grau de determinismo e latência previsível (RAO, 2025; KRASŇANSKÝ; DVORŠČÁK; KOZÁK, 2014). Essas características tornam essa tecnologia especialmente adequada para sistemas de controle que exigem respostas rápidas e precisas, como o pêndulo invertido (ADHIKARI *et al.*, 2011), além de possibilitar a customização da arquitetura de controle em nível lógico, sem a limitação da execução sequencial de instruções.

Dentre as diversas estratégias de controle existentes, o controlador proporcional–integral–derivativo (PID) permanece como uma das técnicas mais empregadas na indústria e no meio acadêmico, em função de sua simplicidade de implementação, interpretabilidade e eficácia em uma ampla gama de sistemas (ÅSTRÖM; HÄGGLUND, 1995). A implementação de controladores PID em FPGA possibilita a exploração das vantagens do hardware reconfigurável, mantendo a estrutura clássica dessa técnica de controle amplamente consolidada na literatura (KRASŇANSKÝ; DVORŠČÁK; KOZÁK, 2014).

Diante desse cenário, este trabalho propõe o desenvolvimento e a validação de uma arquitetura híbrida de controle em tempo real para a estabilização de um pêndulo invertido sobre carrinho. Nessa arquitetura, um microcontrolador é empregado para a aquisição e o pré-processamento dos sinais dos sensores, enquanto uma FPGA executa as rotinas críticas do controle, incluindo o controlador Proporcional–Integral–Derivativo (PID), a geração do sinal PWM e a lógica de acionamento do atuador. A proposta explora as vantagens complementares de cada plataforma, com o objetivo de assegurar execução determinística da malha de controle e evidenciar a aplicabilidade do hardware reconfigurável em sistemas de controle clássicos de alta dinâmica.

Trabalhos Relacionados

No problema de controle do pêndulo invertido, o objetivo principal consiste em conduzir o sistema a partir de uma condição inicial afastada do equilíbrio até a posição vertical instável (*swing-up*) e, posteriormente, manter essa posição de forma estável frente a perturbações externas e incertezas do modelo. Em função de sua dinâmica altamente instável e sensível a atrasos na ação de controle, o pêndulo invertido é amplamente utilizado como sistema de referência para a validação de arquiteturas e estratégias de controle em tempo real.

Trabalhos recentes também exploram abordagens avançadas de controle aplicadas a sistemas de pêndulo invertido, incluindo técnicas baseadas em aprendizado por reforço (HERNÁNDEZ; GARCIA-HERNANDEZ; JURADO, 2024).

Historicamente, os controladores Proporcional–Integral–Derivativo (PID) evoluíram de dispositivos mecânicos para implementações digitais em tempo discreto (KULISZ; JOKIEL, 2024). Atualmente, o PID é amplamente empregado em aplicações industriais e acadêmicas, sendo comumente implementado em software sobre Controladores Lógicos Programáveis (CLPs) ou microcontroladores (MCUs). Nessas plataformas, o período de amostragem assume papel central no desempenho do sistema, uma vez que limita diretamente a velocidade de resposta do processo controlado (KULISZ; JOKIEL, 2024).

Em aplicações de controle associadas a sistemas de alta dinâmica, a literatura aponta que a organização temporal da execução do controle torna-se um fator crítico. A execução sequencial de rotinas de aquisição, processamento e atuação em plataformas baseadas exclusivamente em software pode introduzir atrasos variáveis, especialmente à medida que novas funcionalidades são incorporadas ao sistema, como filtragem digital, comunicação serial ou múltiplas malhas de controle (KULISZ; JOKIEL, 2024). Esse aspecto tem motivado a investigação de arquiteturas alternativas capazes de oferecer maior previsibilidade temporal.

Nesse contexto, as *Field Programmable Gate Arrays* (FPGAs) têm sido amplamente exploradas em aplicações de controle em tempo real, devido à possibilidade de implementação de algoritmos diretamente em hardware, com execução concorrente e temporização determinística (KULISZ; JOKIEL, 2024; WANG et al., 2022). Trabalhos recentes demonstram a aplicação de FPGAs em sistemas de controle de alta precisão, incluindo o uso de estruturas paralelas para controladores PID e a integração de técnicas avançadas de controle e inteligência artificial, evidenciando o potencial dessa tecnologia para sistemas rápidos e não lineares (WANG et al., 2022).

Entretanto, a literatura também aponta que soluções baseadas exclusivamente em FPGA podem apresentar desafios relacionados à complexidade de desenvolvimento, à implementação de protocolos de comunicação e ao gerenciamento de sinais provenientes de sensores digitais. Em razão disso, arquiteturas híbridas, que combinam microcontroladores para tarefas de aquisição, comunicação e pré-processamento com FPGAs dedicadas às rotinas críticas do controle, têm se mostrado uma alternativa eficaz para explorar as vantagens complementares dessas plataformas (KULISZ; JOKIEL, 2024).

No contexto específico do pêndulo invertido, estudos clássicos destacam a natureza altamente dinâmica do sistema, frequentemente ilustrada pela analogia do equilíbrio de uma haste vertical instável, na qual pequenas variações exigem ações corretivas rápidas e contínuas (KEVICZKY et al., 2019). Essa característica reforça a relevância de arquiteturas de controle capazes de garantir previsibilidade temporal e resposta rápida da malha de controle.

Em trabalho anterior, De Lima et al. (2025) apresentaram o desenvolvimento de uma bancada didática de pêndulo invertido, com ênfase nos desafios de integração de sensores e na prototipagem mecânica do sistema. O presente trabalho dá continuidade a essa linha de pesquisa, deslocando o foco para a arquitetura de controle digital e para a implementação do controlador em hardware reconfigurável, explorando uma abordagem híbrida voltada à execução determinística da malha de controle, aspecto não abordado em profundidade no estudo anterior.

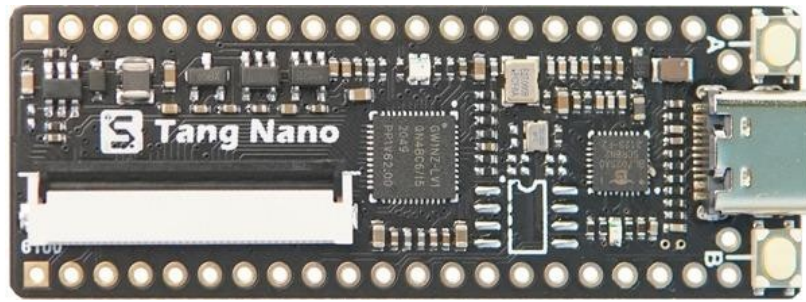
Dessa forma, observa-se uma lacuna na literatura quanto à análise do comportamento temporal de arquiteturas híbridas FPGA–microcontrolador aplicadas especificamente à estabilização do pêndulo invertido, aspecto que este trabalho se propõe a investigar.

Sistema Proposto

O sistema proposto para a estabilização do pêndulo invertido sobre carrinho é baseado em uma arquitetura de controle em malha fechada, integrando sensores, atuadores e algoritmos de controle digital. A estrutura do sistema foi organizada a partir de blocos funcionais bem definidos, com o objetivo de garantir modularidade, determinismo temporal e facilidade de análise do comportamento do controle em tempo real.

A plataforma de hardware reconfigurável utilizada neste trabalho é a placa de desenvolvimento Tang Nano 1K (Figura 2), baseada no dispositivo FPGA Gowin GW1NZ-LV1. Trata-se de uma FPGA de pequeno porte, adequada a aplicações de controle em tempo real, disponibilizando recursos suficientes para a implementação de lógica combinacional e sequencial com comportamento determinístico. A escolha dessa plataforma está alinhada ao objetivo do trabalho de implementar controladores digitais diretamente em hardware descritivo, sem o uso de processadores embarcados ou unidades nativas de ponto flutuante.

Figura 2: Placa de desenvolvimento baseada em chip FPGA Tang Nano 1K



Fonte: Sipeed (2025)

Em alguns dispositivos FPGA disponíveis comercialmente, é possível encontrar arquiteturas que integram núcleos de microcontroladores ou processadores embarcados, possibilitando a execução de tarefas como tratamento de protocolos de comunicação, filtragem digital e operações em ponto flutuante no próprio dispositivo. No entanto, tais soluções apresentam maior complexidade de configuração e custo elevado, não sendo adotadas neste trabalho. O foco foi direcionado à avaliação do comportamento determinístico da lógica implementada em FPGA discreta, associada a um microcontrolador externo para tarefas auxiliares.

Dessa forma, a FPGA utilizada não possui suporte nativo a operações em ponto flutuante, o que impõe limitações quanto à representação direta de grandezas reais. Assim, os dados processados internamente são representados em formato inteiro, utilizando fatores de escala apropriados para preservar a resolução necessária ao controle. Essa abordagem contribui para a redução do consumo de recursos lógicos e para a simplicidade da arquitetura implementada, mantendo compatibilidade com os requisitos dinâmicos do sistema.

Por esse motivo, e considerando também a complexidade de implementação de protocolos de comunicação de alto nível, como o I²C, em FPGA discreta dentro do escopo do trabalho, o sistema adota uma arquitetura híbrida. Nessa arquitetura, a FPGA é responsável pelas rotinas críticas do controle em tempo real, enquanto um microcontrolador atua como unidade auxiliar para tarefas de comunicação e pré-processamento de sinais.

Para a medição do deslocamento do carrinho, é utilizado um sensor de distância a laser VL53L0X, cuja comunicação via protocolo I²C é realizada por um microcontrolador Arduino. O Arduino é responsável pela aquisição dos dados do sensor, pela aplicação de filtragem digital por média móvel e pelo envio das informações processadas à FPGA por meio de comunicação serial UART. A taxa de atualização dos dados foi definida em 50 Hz, valor compatível com a dinâmica do sistema mecânico.

A realimentação da posição angular da haste do pêndulo é realizada por meio de um *encoder* incremental rotativo em quadratura, que fornece dois sinais defasados em 90° elétricos. Esses sinais são processados diretamente pela FPGA, permitindo a determinação da posição relativa e do sentido de movimento, sendo utilizados pelo controlador para a estabilização do sistema.

A implementação do sistema em FPGA foi conduzida de forma modular, sendo subdividida em sete entidades descritas em linguagem VHDL, cada uma responsável por uma função específica do sistema de controle. A entidade principal, denominada *main_inverted.vhd*, é responsável pelo instanciamento e pela interconexão das demais entidades do sistema.

A leitura dos sinais do *encoder* incremental é realizada pelo módulo *read_encoder.vhd*, responsável pelo processamento dos sinais em quadratura. A recepção dos dados enviados pelo microcontrolador é implementada pelo módulo *uart_rx.vhd*, que converte a sequência serial recebida em palavras digitais. Em seguida, o módulo *range_frame_parser.vhd* interpreta essas palavras e converte os valores recebidos para a posição do carrinho em milímetros.

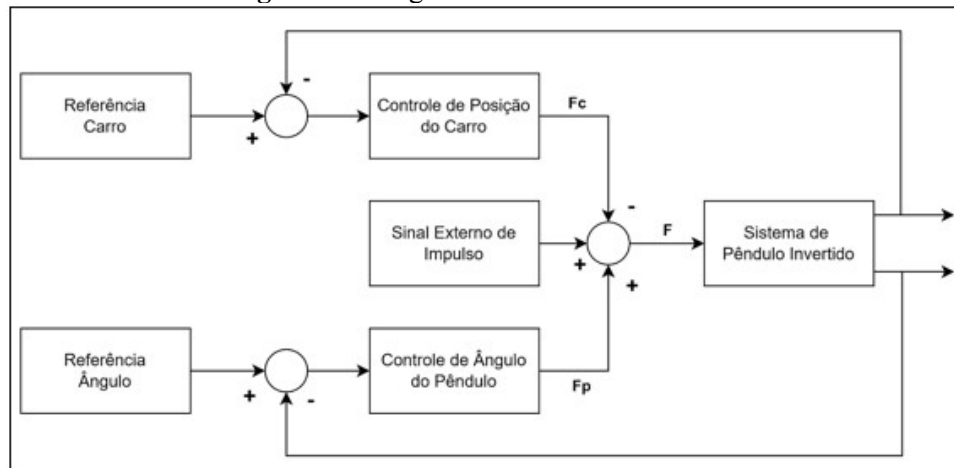
O controle é executado pelo módulo *PID.vhd*, que recebe a referência, o sinal de realimentação, os parâmetros do controlador e a frequência de amostragem, gerando o esforço de controle. Esse módulo é instanciado duas vezes na entidade principal, sendo um controlador dedicado ao deslocamento do carrinho e outro à estabilização da posição angular da haste do pêndulo.

A etapa de atuação é composta pelos módulos *PWM.vhd* e *H_Bridge.vhd*. O primeiro é responsável pela geração do sinal PWM a partir do valor de controle calculado, enquanto o segundo converte o sinal de controle em comandos de direção e *duty cycle* para o acionamento da ponte H. O estágio de potência é implementado por meio de uma ponte H baseada no circuito integrado L298N, responsável pelo acionamento do motor que promove o deslocamento horizontal do carrinho.

Em virtude das diferenças de níveis lógicos entre a FPGA, operando em 3,3 V, e os dispositivos externos que operam em 5 V, foi empregado um módulo conversor de nível lógico bidirecional TXS0108E, garantindo a compatibilidade elétrica dos sinais digitais. Para fins de instrumentação, monitoramento e depuração, a comunicação entre a FPGA e o computador é realizada por meio de uma interface UART, utilizando um conversor USB-serial baseado no circuito CH340.

A organização funcional do sistema é apresentada no diagrama de blocos da Figura 3, evidenciando a interligação entre os módulos de controle.

Figura 3 – Diagrama de Blocos do Sistema

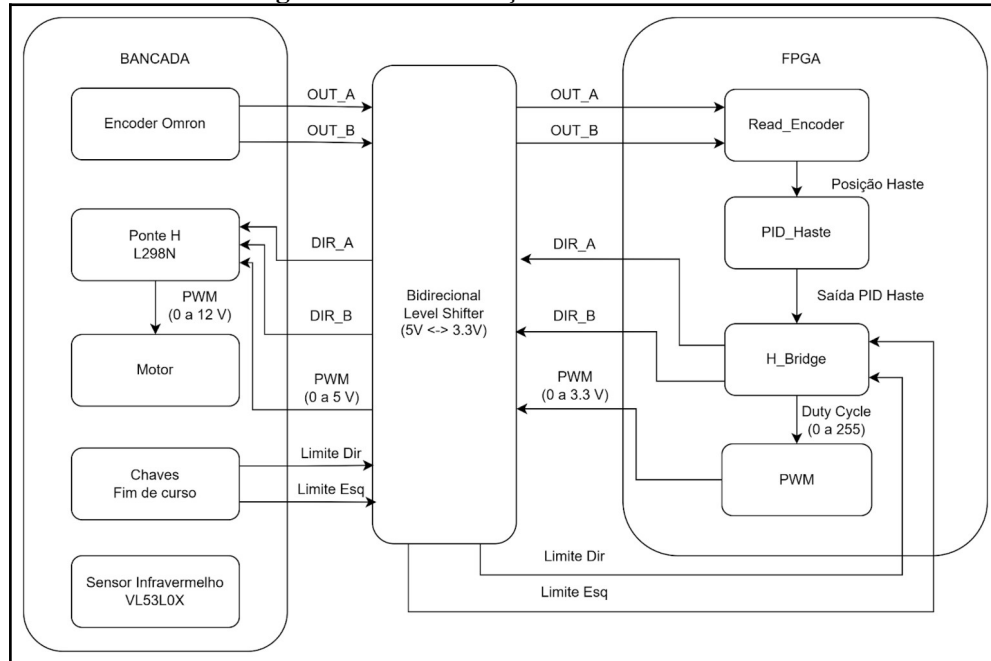


Fonte: Traduzido de Akole e Tyagi (2008)

A Figura 3 apresenta o diagrama de blocos funcional do sistema, mostrando como os módulos de controle se conectam ao processo do pêndulo invertido. O controle é organizado em duas malhas principais: uma de posição do carro e outra de ângulo do pêndulo. Em cada malha, a referência desejada (posição do carro e ângulo) é comparada com a variável medida, gerando um erro que alimenta, respectivamente, o bloco “Controle de Posição do Carro” e o bloco “Controle de Ângulo do Pêndulo”. As saídas desses controladores, indicadas por F_c e F_p , são combinadas em um somador para formar a força total F aplicada ao sistema. Além disso, o diagrama inclui a possibilidade de um sinal externo de impulso, que também entra na soma para representar perturbações ou excitações aplicadas durante os testes. Por fim, as saídas do “Sistema de Pêndulo Invertido” são realimentadas para fechar as malhas, permitindo que o controle corrija continuamente o comportamento do carro e do pêndulo.

A Figura 4 ilustra a comunicação entre os sensores e a FPGA.

Figura 4 – Comunicação Sensores x FPGA

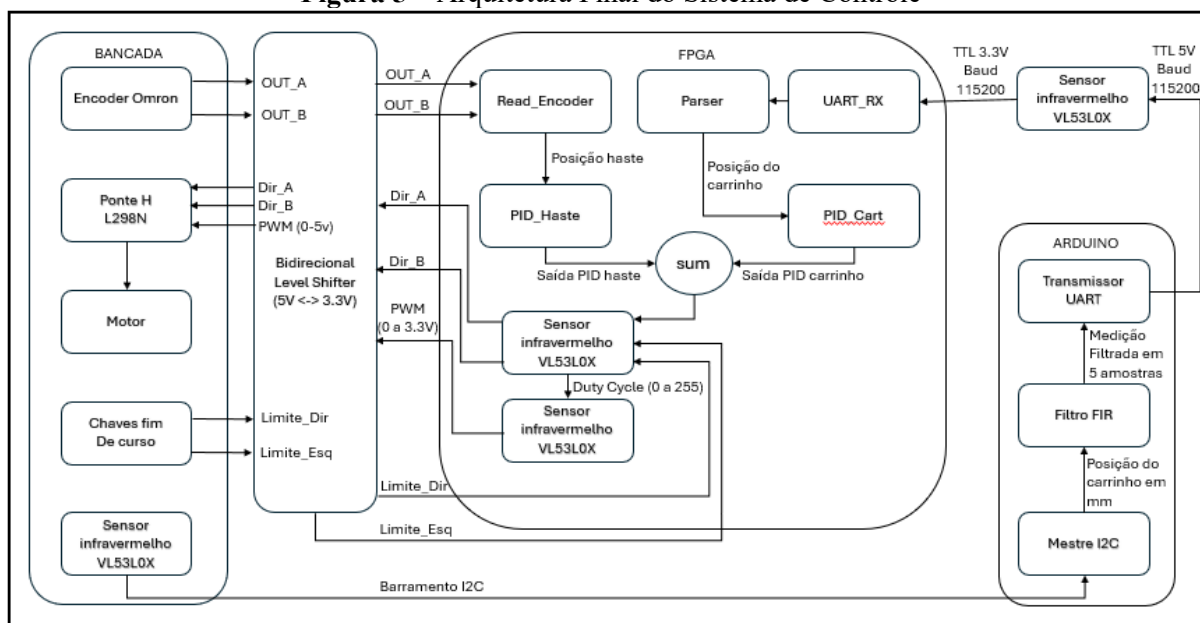


Fonte: os autores

A Figura 4 apresenta o fluxo de sinais entre a bancada e a FPGA. Os pulsos do encoder incremental (OUT_A e OUT_B), que saem em nível TTL de 5 V, passam por um conversor bidirecional de nível lógico (5 V ↔ 3,3 V) antes de entrarem na FPGA, onde são lidos pelo bloco Read_Encoder para estimar a posição da haste. A posição calculada alimenta o controlador PID_Haste, que gera um comando de controle. Esse comando é aplicado ao bloco H_Bridge, responsável por definir o sentido (DIR_A e DIR_B) e o ciclo de trabalho do PWM, que determinam a tensão média aplicada ao motor via ponte H L298N. Além disso, as chaves de fim de curso (Limite Dir e Limite Esq) também são encaminhadas à FPGA, permitindo implementar intertravamentos de segurança e impedir movimento além dos limites mecânicos do sistema.

Já a Figura 5 apresenta a arquitetura final do sistema de controle implementado.

Figura 5 – Arquitetura Final do Sistema de Controle



Fonte: os autores

A Figura 5 apresenta a arquitetura final do sistema, integrando aquisição, processamento e atuação. Na bancada, o encoder incremental fornece os sinais `OUT_A` e `OUT_B`, que passam pelo conversor bidirecional de nível lógico ($5\text{ V} \leftrightarrow 3,3\text{ V}$) e são lidos na FPGA pelo bloco `Read_Encoder`, responsável por estimar a posição da haste. Essa posição alimenta o controlador `PID_Haste`, que gera uma parcela do comando de controle. Em paralelo, a posição do carrinho é obtida pelo sensor infravermelho VL53L0X, cuja leitura é feita no Arduino (como mestre I2C), filtrada (filtro FIR e média em 5 amostras) e enviada por UART a 115200 bps; após adequação de nível (TTL 5 V para TTL 3,3 V), a FPGA recebe os dados via `UART_RX`, interpreta no `Parser` e aplica o controlador `PID_Carr`. As duas saídas de controle (da haste e do carrinho) são combinadas no bloco `sum`, resultando no comando final aplicado ao estágio de potência: a FPGA define direção (`Dir_A` e `Dir_B`) e duty cycle (`PWM`), que acionam a ponte H L298N e, consequentemente, o motor. As chaves de fim de curso (`Limite_Dir` e `Limite_Esq`) também entram na FPGA para intertravamento e proteção do movimento.

Metodologia Experimental

A metodologia experimental adotada foi estruturada de forma cronológica e orientada por requisitos, tendo como foco a implementação e a validação de uma arquitetura híbrida de controle em tempo real para a estabilização de um pêndulo invertido sobre carrinho. O desenvolvimento foi conduzido de maneira incremental, permitindo a verificação progressiva do funcionamento da bancada experimental, dos subsistemas de aquisição, atuação e controle, bem como do comportamento temporal da malha de controle.

Inicialmente, foi realizada a validação funcional da bancada experimental, abrangendo sensores, atuador, comunicação e instrumentação. Nessa etapa, utilizou-se um microcontrolador Arduino como plataforma de apoio, com o objetivo de assegurar o correto

funcionamento dos componentes físicos do sistema e estabelecer uma base operacional estável para o desenvolvimento da arquitetura definitiva. Essa fase permitiu verificar o comportamento do atuador, a confiabilidade dos sensores e a integridade da comunicação, sem foco em desempenho temporal.

Com a bancada validada, procedeu-se à implementação da arquitetura híbrida descrita na seção de Sistema Proposto, na qual as funções do sistema foram distribuídas de acordo com as características de cada plataforma. O microcontrolador Arduino UNO passou a atuar como unidade auxiliar, sendo responsável pela leitura do sensor de distância VL53L0X via protocolo I²C, pela aplicação de filtragem digital por média móvel e pela transmissão dos dados processados à **FPGA** conforme descrito na seção Sistema Proposto. A FPGA concentrou as rotinas críticas do controle em tempo real, assegurando execução síncrona e determinística da malha de controle.

A implementação do controle na FPGA foi realizada em linguagem de descrição de hardware (**VHDL**), adotando uma abordagem modular. Cada bloco funcional foi desenvolvido e validado individualmente, incluindo os módulos de leitura do *encoder* incremental, recepção e decodificação serial, controlador PID, geração do sinal PWM e lógica de direção. Antes da integração completa, os módulos foram submetidos a simulações funcionais, com o objetivo de verificar o correto comportamento lógico e temporal da implementação.

O controlador PID foi implementado em forma discreta, considerando a natureza digital da execução em hardware. A temporização da malha de controle foi definida a partir do clock de referência da FPGA, configurado em 27 MHz, garantindo execução síncrona das etapas de processamento. A taxa de amostragem do controlador foi estabelecida em 1 kHz, valor selecionado de modo a atender aos requisitos dinâmicos do sistema mecânico e do atuador, evitando efeitos adversos associados à discretização.

Durante os ensaios experimentais, foram incorporadas estratégias de proteção e robustez ao sistema, incluindo saturação do sinal de controle, bloqueio de movimento por meio das chaves de fim de curso e sincronização de sinais assíncronos por meio de *flip-flops* em cascata. Essas medidas visaram garantir operação segura e confiável da bancada durante os testes em malha fechada.

No subsistema de realimentação, os sinais do *encoder* incremental foram processados por meio de um algoritmo de decodificação em quadratura dupla, permitindo a determinação precisa da posição e do sentido de movimento da haste. Para o sinal de deslocamento do carrinho proveniente do sensor infravermelho, foi aplicada filtragem digital do tipo FIR, com o

objetivo de atenuar componentes de alta frequência sem comprometer as dinâmicas relevantes do sistema.

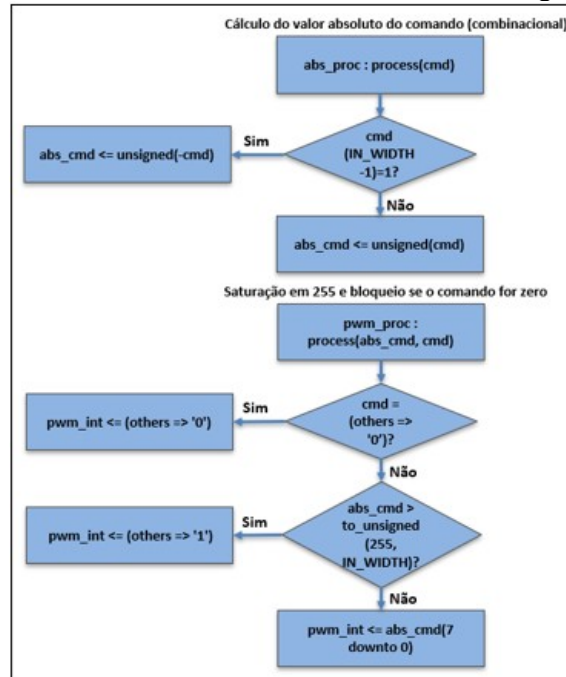
Após a integração completa do sistema em malha fechada, foram realizados os ensaios experimentais para análise do comportamento do controle em tempo real. A avaliação do desempenho foi conduzida com base em critérios relacionados à previsibilidade temporal da execução da malha de controle, à estabilidade operacional e ao comportamento dinâmico do sistema, em consonância com o objetivo do trabalho de analisar a aplicação de arquiteturas híbridas FPGA–microcontrolador em sistemas rápidos e intrinsecamente instáveis.

Resultados

Os resultados obtidos evidenciam os impactos da adoção de uma arquitetura híbrida de controle, na qual as rotinas críticas em tempo real são executadas em FPGA, enquanto tarefas auxiliares permanecem no microcontrolador. Observou-se que a implementação em hardware reconfigurável permitiu a construção de uma arquitetura lógica determinística, composta por módulos independentes e executados de forma concorrente, refletindo diretamente no comportamento temporal da malha de controle.

A Figura 6 apresenta a arquitetura lógica implementada na FPGA para o processamento do comando de controle. O fluxograma ilustra a separação explícita entre a magnitude e o sentido do comando, bem como a geração do sinal PWM a partir do valor absoluto do comando, incorporando mecanismos de saturação e bloqueio para condições de comando nulo.

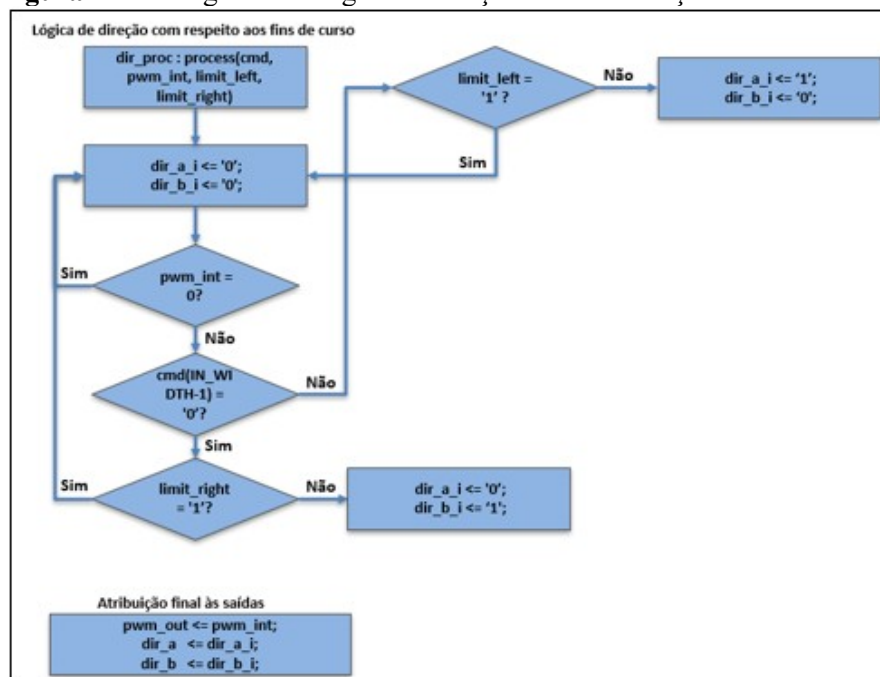
Figura 6 – Fluxograma do cálculo do valor absoluto do comando e geração do sinal PWM na FPGA



Fonte: os autores

Essa abordagem assegura compatibilidade direta com o módulo de geração de PWM e reduz a ocorrência de sobrecomando, prevenindo esforços excessivos sobre o atuador e contribuindo para a estabilidade da malha de controle. Complementarmente, a Figura 7 apresenta a lógica de definição do sentido de rotação do atuador, considerando o sinal do comando, o valor do PWM calculado e os estados das chaves de fim de curso.

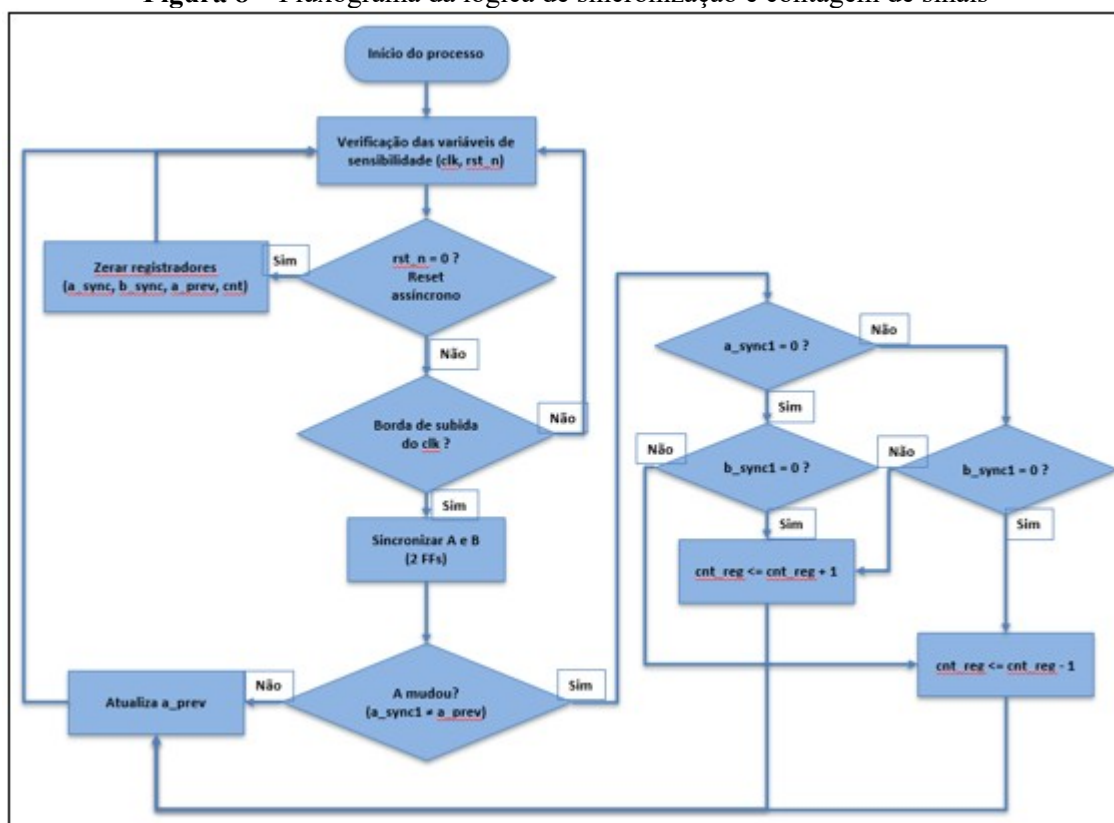
Figura 7 – Fluxograma da lógica de direção com verificação de fins de curso



Fonte: os autores

Observa-se que o **movimento é automaticamente bloqueado** quando o **fim de curso correspondente ao sentido solicitado encontra-se ativo**, assegurando a proteção mecânica do sistema durante os ensaios experimentais. A Figura 8 apresenta a lógica implementada para a **sincronização de sinais assíncronos** e a contagem de eventos no domínio de clock da FPGA.

Figura 8 – Fluxograma da lógica de sincronização e contagem de sinais



Fonte: os autores

A utilização de *flip-flops* em cascata reduziu a probabilidade de ocorrência de metaestabilidade, enquanto a detecção controlada de transições assegurou a atualização confiável do contador interno. Esse aspecto mostrou-se fundamental para a robustez do sistema, considerando a presença de sinais externos não sincronizados provenientes dos sensores.

No subsistema de realimentação por *encoder*, verificou-se experimentalmente que a contagem baseada exclusivamente em interrupções no microcontrolador apresentava suscetibilidade a erros durante reversões rápidas de movimento e sob vibração mecânica. A implementação do algoritmo de decodificação em quadratura dupla na FPGA eliminou o erro acumulado de contagem, permitindo a preservação da referência de equilíbrio mesmo após ciclos repetidos de movimento, aumentando a confiabilidade da variável de estado utilizada pelo controlador.

Em relação à medição do deslocamento do carrinho por meio do sensor VL53L0X, observou-se a presença de ruído no sinal bruto, o que impactava o cálculo do erro de controle. A aplicação de filtragem digital do tipo FIR atenuou significativamente as componentes de alta frequência, resultando em uma realimentação mais estável, sem comprometer de forma relevante a dinâmica do sistema.

A execução das rotinas críticas do controle em FPGA evidenciou comportamento temporal previsível, com frequência de amostragem constante e latência definida pelo clock do hardware. Em contraste, a execução sequencial das tarefas no microcontrolador mostrou-se adequada para prototipagem e validação funcional, porém mais sensível a variações temporais decorrentes da concorrência entre tarefas, reforçando a adequação da arquitetura híbrida adotada.

De forma geral, os resultados demonstram que a utilização de hardware reconfigurável para a execução da malha de controle permite maior controle sobre a temporização do sistema, contribuindo para a estabilidade e previsibilidade do comportamento dinâmico do pêndulo invertido, especialmente em um sistema intrinsecamente instável.

Conclusão

Este trabalho apresentou o desenvolvimento e a validação de uma arquitetura híbrida de controle em tempo real aplicada à estabilização de um pêndulo invertido sobre carrinho, explorando as vantagens complementares de um microcontrolador e de uma FPGA. A solução proposta alocou ao microcontrolador as tarefas de aquisição e pré-processamento dos sinais dos sensores, enquanto a FPGA concentrou a execução das rotinas críticas do controle, incluindo o controlador Proporcional–Integral–Derivativo (PID), a geração do sinal PWM e a lógica de acionamento do atuador.

Os resultados obtidos evidenciaram que a execução do controle diretamente em hardware reconfigurável proporciona maior previsibilidade temporal, com frequência de amostragem constante e latência definida pelo *clock* do sistema, características fundamentais para a estabilidade de sistemas intrinsecamente instáveis. A abordagem modular adotada na implementação em FPGA permitiu a separação clara das funções do sistema, facilitando a validação individual dos blocos e contribuindo para a robustez da malha de controle.

A utilização do microcontrolador como unidade auxiliar mostrou-se adequada para o tratamento de protocolos de comunicação, filtragem digital e integração com sensores, reduzindo a complexidade da implementação na FPGA e evidenciando a viabilidade prática de

arquiteturas híbridas em aplicações de controle em tempo real. Dessa forma, o trabalho demonstra que a combinação entre FPGA e microcontrolador constitui uma solução eficiente e flexível, conciliando determinismo temporal e facilidade de desenvolvimento.

Por outro lado, verificou-se que o desenvolvimento em FPGA impõe maior esforço de projeto, uma vez que a implementação ocorre em nível de hardware descritivo, demandando cuidados adicionais quanto à temporização, representação numérica e consumo de recursos lógicos. A adoção de aritmética inteira escalonada, embora adequada aos objetivos do trabalho, impõe limitações de precisão que devem ser consideradas em aplicações que exigem maior resolução numérica.

Como trabalhos futuros, propõe-se a realização de uma análise comparativa sistemática entre diferentes arquiteturas de controle aplicadas ao pêndulo invertido, incluindo uma implementação integral baseada em microcontrolador e uma implementação dedicada em FPGA, sob condições experimentais controladas e métricas bem definidas. Essa comparação permitiria quantificar de forma objetiva os impactos do determinismo temporal, da latência e do *jitter* no desempenho dinâmico do sistema, complementando a análise qualitativa apresentada neste trabalho.

Adicionalmente, sugere-se a investigação do uso de plataformas FPGA de maior capacidade, que disponham de suporte nativo a operações em ponto flutuante ou recursos dedicados para processamento numérico, de modo a viabilizar a implementação de controladores com maior resolução e precisão. A adoção dessas plataformas permitiria avaliar os ganhos decorrentes do uso de aritmética em ponto flutuante na implementação do controlador PID, bem como a aplicação de estratégias de controle mais avançadas, ampliando o escopo do sistema proposto.

Referências

ADHIKARI, Bikram; GURUNG, Deepak; KUNWAR, Giresh Singh; GYAWALI, Prashanta. FPGA control of a mobile inverted pendulum robot. **Journal of the Institute of the Engineering**, Nepal, v. 8, n. 1, p. 188–196, 2011.

AKOLE, Mohan; TYAGI, Barjeev. Design of fuzzy logic controller for nonlinear model of inverted pendulum-cart system. In: **NATIONAL SYSTEMS CONFERENCE (NSC)**, 32., 2008, Roorkee, India. Proceedings [...]. Roorkee: Indian Institute of Technology Roorkee, 2008.

ÅSTRÖM, Karl Johan; HÄGGLUND, Tore. **PID controllers**. 2. ed. Research Triangle Park: Instrument Society of America, 1995.

HERNANDEZ, Ruben; GARCIA-HERNANDEZ, Ramon; JURADO, Francisco. Modeling, Simulation, and Control of a Rotary Inverted Pendulum: A Reinforcement Learning-Based Control Approach. **Modelling**, v. 5, p. 1824–1852, 2024. DOI: 10.3390/modelling5040095.

HERNÁNDEZ-GUZMÁN, V. M.; SILVA-ORTIGOZA, R. **Automatic Control with Experiments**. Cham: Springer, 2019.

KEVICZKY, László; BARS, Ruth; HETTHÉSSY, Jenő; BÁNYÁSZ, Csilla. **Control engineering**. 1. ed. Cham: Springer, 2019.

KRASŇANSKÝ, Róbert; DVORŠČÁK, Branislav; KOZÁK, Štefan. Hardware realization of embedded control algorithm on FPGA. In: **COMPUTATION TOOLS 2014 – The Fifth International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking**. Lisboa: IARIA, 2014.

KULISZ, Józef; JOKIEL, Filip. A Hardware Implementation of the PID Algorithm Using Floating-Point Arithmetic. **Electronics**, v. 13, art. 1598, 19 p., 2024. DOI: 10.3390/electronics13081598.

OGATA, Katsuhiko. **Engenharia de controle moderno**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

DE LIMA, Alessandro; ALMEIDA, Bruno da Silva; RABELLO, Giovanna; MEIRELLES, Guilherme Miyata; FERREIRA, Tharick Pontes; HIRATA, Anderson Kenji; GONÇALVES, Valdeci Donizete. Aplicação de uma bancada didática de pêndulo invertido: desafios na integração de sensores e prototipagem. In: **CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA (COBENGE)**, 2025, Campinas, SP. Anais [...]. Brasília: ABENGE, 2025. DOI: <https://doi.org/10.37702/2175-957x.cobenge.2025.6361>.

RAO, Ravi. FPGA vs Microcontroller: Differences and Applications. **Wevolver**, 2025. Disponível em: < <https://www.wevolver.com/article/fpga-vs-microcontroller> >. Acesso em: 3 jan. 2026.

SIPEED. **Tang Nano 1K**. [S. l.], 2025. Disponível em: < <https://wiki.sipeed.com/hardware/en/tang/Tang-Nano-1K/Nano-1k.html> >. Acesso em: 12 jan. 2026.

WANG, Jun; LI, Moudao; JIANG, Weibin; HUANG, Yanwei; LIN, Ruiquan. A Design of FPGA-Based Neural Network PID Controller for Motion Control System. **Sensors**, v. 22, art. 889, 18 p., 2022. DOI: 10.3390/s22030889.