



ISSN 2447-6102



Artigo

V-FES-Cycling: Desenvolvimento de uma interface de comunicação entre um sistema FES-Cycling e um ambiente de realidade virtual para reabilitação de pacientes com lesão medular

Sousa, B.L.^{1,*}, Silva Levy, V.², Bestard, G.A.³, Cormane, J.⁴, Ochoa-Diaz, C.⁵

¹ Programa de Pós-Graduação em Engenharia Biomédica (PPGEB), Faculdade de Ciências e Tecnologia em Engenharia (FCTE), Universidade de Brasília; breno.linhares@aluno.unb.br

² Departamento de Design, Universidade de Brasília; vi.silva.levy@proton.me

³ Faculdade de Ciências e Tecnologia em Engenharia (FCTE), Universidade de Brasília; guilhermo@unb.br

⁴ Faculdade de Ciências e Tecnologia em Engenharia (FCTE), Universidade de Brasília; jcormane@unb.br

⁵ Faculdade de Ciências e Tecnologia em Engenharia (FCTE), Universidade de Brasília; claudiaochoa@unb.br

* Correspondência: breno.linhares@aluno.unb.br; Tel.: +55 (61)996868223

Received: 30/05/2025; Accepted: 03/06/2025; Published: 10/06/2025

Resumo: A reabilitação de indivíduos com Lesão Medular (LME) enfrenta desafios relacionados à adesão dos pacientes devido à natureza repetitiva dos exercícios de reabilitação. A Realidade Virtual (RV) e os jogos sérios têm sido aplicados para aumentar a motivação, permitindo um ambiente interativo e imersivo. A Estimulação Elétrica Funcional (FES) é uma abordagem eficaz para ativar grupos musculares e promover o ciclismo assistido, facilitando a reabilitação. Este estudo apresenta a plataforma *V-FES-Cycling*, que integra um triciclo acionado por FES a um ambiente virtual para reabilitação. A plataforma inclui um triciclo instrumentado com um sensor encoder para monitorar a pedalada, transmitindo os dados via *Bluetooth* para um software desenvolvido na Unreal Engine (UE). O sistema permite a sincronização do movimento real com um modelo 3D na RV e ajustes em tempo real dos parâmetros da estimulação elétrica. Os resultados demonstraram que a interface possibilita uma sincronização precisa dos dados, com latência de 0,97ms, garantindo resposta rápida e imersão do usuário. Futuras melhorias incluem aprimoramento do protocolo de comunicação, implementação de um freio eletromagnético e desenvolvimento de um exergame para aumentar a adesão dos pacientes. Ensaios clínicos futuros serão conduzidos para validar a eficácia da plataforma na reabilitação motora.

Palavras-chave: Reabilitação; Ciclismo-FES; Realidade Virtual; Lesão medular; Integração.

V-FES-Cycling: Development of a communication interface between a FES-Cycling system and a virtual reality environment for rehabilitation of spinal cord injury patients

Abstract: The rehabilitation of individuals with Spinal Cord Injury (SCI) faces challenges related to patient compliance due to the repetitive nature of rehabilitation exercises. Virtual Reality (VR) and serious games have been applied to increase motivation, allowing for an interactive and immersive environment. Functional Electrical Stimulation (FES) is an effective approach to activate muscle groups and promote assisted cycling, facilitating rehabilitation. This study presents the *V-FES-Cycling* platform, which integrates an FES-driven tricycle with a virtual environment for



rehabilitation. The platform includes an instrumented tricycle with an encoder sensor to monitor pedaling, transmitting the data via Bluetooth to software developed in Unreal Engine (UE). The system allows synchronization of the real movement with a 3D model in VR and real-time adjustments of the electrical stimulation parameters. The results showed that the interface enables precise data synchronization, with a latency of 0.97ms, guaranteeing a quick response and user immersion. Future improvements include improving the communication protocol, implementing an electromagnetic brake and developing an exergame to increase patient compliance. Future clinical trials will be conducted to validate the platform's effectiveness in motor rehabilitation.

Keywords: Rehabilitation; FES-Cycling; Virtual Reality; Spinal Cord Injury; Integration.

1. Introdução

A reabilitação de indivíduos com Lesão Medular (LME) permanece um desafio em saúde pública, principalmente por necessitar que pacientes com limitações motoras realizem exercícios repetitivos e intensivos que podem comprometer a adesão devido à falta de motivação e engajamento (ESFAHLANI et al., 2018). Nesse contexto, a Realidade Virtual (RV) e os jogos sérios têm sido amplamente empregados para proporcionar uma experiência mais imersiva e estimulante, favorecendo a adesão de pacientes e melhorando resultados terapêuticos em comparação com abordagens convencionais (SOUSA; BALBINO, 2018), (CYRINO, 2019).

Um sistema de RV é uma tecnologia de interface que recria ambientes simulados, proporcionando ao usuário uma experiência imersiva e interativa. Em reabilitação, os jogos sérios vão além do entretenimento, sendo aplicados no aprendizado e no treinamento motor. Dentro desse contexto, os *exergames*, que combinam exercícios físicos com elementos de jogos, são amplamente utilizados para incentivar a prática de atividades físicas, especialmente na neurorreabilitação. Estudos indicam que sistemas interativos baseados em *exergames* são eficazes na motivação de pessoas com deficiência para a realização de exercícios (ROJO et al., 2023), (MOLINA et al., 2014).

A reabilitação baseada em tecnologias imersivas aumenta a motivação dos pacientes ao possibilitar a realização de atividades físicas em ambientes virtuais seguros. Além disso, fornece feedback visual, auditivo e tátil, tornando os objetivos mais claros e facilitando a execução de tarefas repetitivas e trabalhosas (ROJO et al., 2023), (CUESTA-GOMEZ et al., 2020).

A LME é uma condição neurológica que compromete o sistema motor e somatossensorial, causando paralisia parcial ou total dos membros inferiores e desencadeando complicações metabólicas e sistêmicas que afetam a qualidade de vida desse público (MONGEOT et al., 2023), (SANTANA et al., 2023). A Estimulação Elétrica Funcional (FES) é uma abordagem eficaz na reabilitação, promovendo contrações musculares por meio de estímulos elétricos, viabilizando movimentos como caminhar e pedalar. Seus benefícios incluem prevenção da atrofia, aumento da amplitude de movimento e estímulo à neuroplasticidade (MAGALHÃES, 2022). Dentre as aplicações da FES, o ciclismo assistido por FES também chamado de (*FES-cycling*) destaca-se na reabilitação ao ativar grupos musculares de forma sincronizada, permitindo a execução de exercícios de pedalada controlados em bicicletas estacionárias ou *Trike* reduzindo riscos de quedas e fraturas (FATTAL et al., 2020).

Estudos anteriores sobre tecnologias de reabilitação em indivíduos com LME utilizaram principalmente dispositivos comerciais prontos, como eletroestimuladores, cicloergômetros e jogos de ciclismo, além de RV semi-imersiva. No estudo realizado por (DUFFELL et al., 2019) foi analisado os efeitos da plataforma *iCycle*, que integra um cicloergômetro acionado por FES com o *biofeedback* de um jogo comercial de corrida em RV semi-imersiva, sem detalhar a integração entre hardware e software, focando mais na análise dos escores motores de pacientes. Já Rojo (ROJO et al., 2023) desenvolveu a plataforma *PedaleoVR*, composta por um cicloergômetro não acionado por FES sincronizado a um exergame com diferentes cenários virtuais, avaliando usabilidade, credibilidade, motivação e efeitos adversos de desconforto chamado *cybersickness* causados pela plataforma em RV.

Dentre os estudos reportados, poucos dos trabalhos integraram hardware e software de RV e *FES-Cycling* para reabilitação. A combinação dessas tecnologias ainda é pouco explorada como estratégia para tornar o treinamento mais imersivo, interativo e confortável, utilizando *biofeedback* sensorial para reduzir possíveis sintomas de desconforto causado pela RV.

Nesse contexto, este artigo apresenta a primeira etapa do desenvolvimento de uma interface de comunicação entre o *firmware* de uma placa de estimulação elétrica de um triciclo acionado por FES, desenvolvido em um trabalho anterior por (MAGALHÃES, 2022), e um software para interface em RV. A interface captura e transmite dados de um sensor que monitora o ângulo da pedalada, sincronizando-os com o movimento de um objeto tridimensional no

ambiente virtual. Além disso, permite ajustar os parâmetros de eletroestimulação em tempo real. Os resultados estabelecem uma base para futuras pesquisas sobre interação de usuários com LME em RV, tornando o treinamento mais realista e eficaz.

2. Componentes da Plataforma V-FES-Cycling

A integração da tecnologia de reabilitação por *FES-Cycling* com a RV requer o desenvolvimento de uma interface de comunicação capaz de transmitir e receber dados entre o *firmware* da placa de estimulação FES e o software de ambiente virtual. Essa interface deve processar os dados em tempo real, permitindo ao usuário interagir com elementos gráficos, como a sincronização da pedalada real com a virtual.

A plataforma *V-FES-Cycling* foi desenvolvida com o sistema *Power Trike*, um triciclo adaptado para pessoas com LME e equipado com um sensor encoder incremental para medir ângulo e cadência da pedalada. Um circuito FES de 8 canais gera a pedalada por estimulação de músculos específicos da perna, conforme detalhado em (MAGALHÃES, 2022). A imersão do usuário ocorre por meio do *HTC Vive Pro 2*, que rastreia os movimentos das mãos e da cabeça (HTC Corporation, 2025).

A comunicação entre os sistemas foi implementada por *Bluetooth*, com um módulo HC-05 no circuito FES e uma antena integrada ao computador. A interface foi programada no Unreal Engine 5.4.3 (UE), da Epic Games, permitindo a criação do ambiente virtual via C++ e *Blueprint*. A Figura 1 apresenta uma visão geral do sistema.



Figura 1. Representação geral do sistema: A plataforma *V-FES-Cycling* é composta por um triciclo adaptado para pacientes com lesão medular, onde a parte traseira do triciclo possui um circuito de estimulação FES que são conectados os eletrodos colocados nos músculos específicos dos membros inferiores. A parte frontal do triciclo é instrumentada com um sensor encoder incremental que mede o ângulo de pedalada. E o usuário do triciclo visualiza o ambiente em RV desenvolvido na UE por meio dos óculos (*Headset do HTC Vive pro 2*) e consegue realizar algumas interações com as mãos por meios dos controladores.

2.1 Componentes para o desenvolvimento do ambiente virtual

O *HTC Vive Pro 2* é um sistema de RV composto por um *Head-Mounted Display* (HMD), dois controladores e duas estações bases. O HMD exibe o ambiente virtual com uma taxa de atualização de até 120 Hz. As estações bases utilizam infravermelho para rastrear a posição e orientação dos controladores e do HMD no espaço tridimensional calibrado.

A Unreal Engine (UE) foi escolhida por suas ferramentas avançadas para criação e animação de elementos 3D, simulação física, interfaces gráficas e efeitos visuais realistas, além de permitir integração com o *HTC Vive Pro 2*. Neste trabalho, a UE foi utilizada com programação em C++ e *Blueprints* a programação em C++ permitiu manipulação de

dados e comunicação serial via *Bluetooth*, enquanto os *Blueprints* facilitaram a interação com elementos visuais e objetos 3D.

O circuito de estimulação FES é baseado em um microcontrolador *ARM Cortex-M7 (STM32)*, responsável pelo gerenciamento e processamento do sistema. A placa de estimulação gera pulsos bifásicos simétricos com frequência de até *100 Hz*, largura de pulso e amplitude de corrente ajustáveis. A comunicação ocorre via *Bluetooth* utilizando o módulo HC-05.

3.Interface de Comunicação Entre os Sistemas

A Figura 2 apresenta o diagrama de blocos da interface de comunicação entre o sistema *FES-Cycling* e o sistema de RV, composto por três módulos principais: circuito de estimulação FES, módulo de RV e interface virtual desenvolvida na UE. A comunicação entre os módulos ocorre de forma bidirecional, conforme ilustrado pelas setas TX e RX, com uma taxa de *10 Hz*. O circuito de estimulação FES utiliza um *firmware* previamente desenvolvido por Magalhães (MAGALHÃES, 2022), responsável por receber os parâmetros de eletroestimulação via *Bluetooth*, gerar pulsos elétricos bifásicos para os canais habilitados e enviá-los aos eletrodos de superfície. Além disso, realiza a aquisição e o processamento dos dados do sensor encoder, transmitindo-os por *Bluetooth*. O módulo de RV recebe esses dados por meio da antena *Bluetooth 5.2* do computador e os encaminha ao módulo de interface virtual da UE via porta serial.

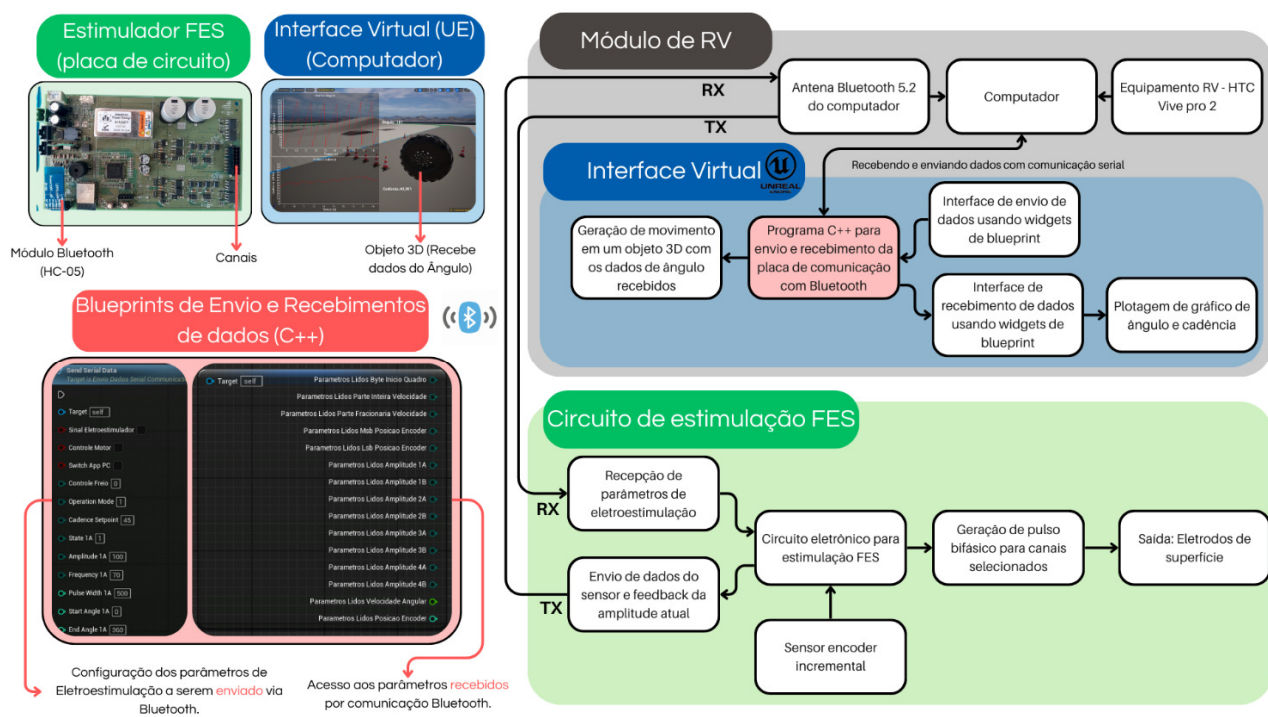


Figura 2. Diagrama de blocos de integração do sistema: A comunicação utilizada para a integração do sistema é realizada via *Bluetooth*, permitindo o envio e recebimento de dados entre o módulo de Realidade Virtual (RV) e o circuito de estimulação FES. O módulo de RV contém uma interface interativa desenvolvida na UE, enquanto o circuito de estimulação FES é responsável por gerar pulsos bifásicos nos eletrodos, além de realizar a aquisição e o processamento dos dados obtidos pelo sensor encoder. Essa comunicação na interface Virtual é feita por meio das *blueprint* de envio e recebimento de dados programadas em C++.

3.1 Interface virtual

O módulo de interface virtual é responsável pelo tratamento e interpretação dos dados do quadro de comunicação. A UE permite a criação de classes em C++ que podem ser acessadas em *Blueprints* por meio das macros *UPROPERTY* e *UFUNCTION*, possibilitando a interação de elementos visuais e 3D com os dados recebidos. Conforme ilustrado na Figura 2, a interface virtual implementa dois blocos *Blueprints* em C++. A *Blueprint* chamada *SendSerialData*

permite ao usuário configurar os parâmetros de eletroestimulação, antes do envio dos dados. Já a *Blueprint* de recepção possibilita acessar os dados recebidos, e utilizá-los para interagir com elementos 3D no ambiente virtual. As principais funções do módulo de interface virtual são:

- **Recepção de dados do FES-Cycling:** Exibição em tempo real dos gráficos de ângulo e cadência da pedalada do usuário na interface em RV, a partir dos dados do sensor encoder.
- **Configuração e envio de parâmetros de estimulação:** A *Blueprint SendSerialData* permite definir parâmetros como frequência, amplitude, largura de pulso e modo de operação (manual, automático ou controle), que são enviados à placa de estimulação FES para gerar pulsos bifásicos nos canais selecionados.
- **Sincronização do movimento em um objeto 3D:** O nó *posição encoder* da *Blueprint* de recepção obtém os valores de ângulo e os transfere a um objeto 3D de uma roda, gerando a animação de rotação de 0 a 360°, sincronizada com o movimento real de pedalada do usuário.

3.2 Quadro de comunicação

A Tabela 1 apresenta uma versão reduzida do quadro de comunicação para envio e recebimento de dados. A transmissão é representada por (Tx-PC), indicando os dados enviados pela placa de estimulação FES ao computador, e (Rx-Placa), que corresponde aos dados enviados do computador para a placa. No total, 13 bytes são utilizados para a transmissão ao computador e 78 bytes para a transmissão à placa, sendo que alguns bytes são reservados e não utilizados. A tabela também detalha especificações como faixa de valores, unidades, resolução e quantidade de bytes de cada variável.

A *Blueprint* de recepção de dados foi programada para comunicação serial utilizando a biblioteca *Windows.h*, recebendo os parâmetros: início do quadro, ângulo de pedal, cadência e amplitudes de corrente dos canais 1A até 4B. O byte de início do quadro é utilizado para identificar o primeiro byte e garantir a sincronização correta dos dados. A *Blueprint SendSerialData* envia os parâmetros modo de operação, estado do canal, corrente, frequência, largura de pulso, ângulo inicial e final para os oito canais da placa de estimulação FES, sendo quatro destinados à perna direita e quatro à esquerda. A variável *estado canal* permite habilitar ou desabilitar canais individualmente, enquanto modo de operação assume valores (0,1,2,3,4), representando os modos desligado, manual, automático, controle e avançado, respectivamente. Nos testes da interface, apenas o modo manual foi utilizado para configurar as variáveis de forma independente.

Tabela 1. Quadro de comunicação do sistema.

Variáveis	Faixa	Resolução	Bytes	Tx/Rx
Início do quadro	2	-	1	Tx-PC
Ângulo pedal	0° – 360°	1°	2	Tx-PC
Cadência	0 – 60 RPM	0.1 RPM	2	Tx-PC
Canal	0–100 mA	1 mA	1	Tx-PC
1A,1B,2A,2B,3A,3B,4A,4B	Reservado	-	-	Tx-PC
Byte 13				
Parâmetros para 8 canais				
Modo de operação	0,1,2,3,4	-	1	Rx-Placa
Estado canal	0(OFF),1(ON)	-	1	Rx-Placa
Corrente	0–100 mA	1 mA	1	Rx-Placa
Frequência	0–35 Hz	1 Hz	1	Rx-Placa
Largura de pulso	0–500 µs	1 µs	2	Rx-Placa
Ângulo inicial	0°–360°	1°	2	Rx-Placa
Ângulo final	0°–360°	1°	2	Rx-Placa
Bytes 0,2,76 e 77	Reservados	-	-	Rx-Placa

4. Resultados

Os principais resultados incluem o desenvolvimento de um programa em C++ para a interface de comunicação entre o módulo de RV e o sistema de reabilitação por *FES-Cycling*. A Figura 3 apresenta o fluxo lógico da leitura dos dados recebidos via porta serial. Ao inicializar o ambiente em RV, a variável *serialHandle* é definida como inválida para indicar que a porta serial está fechada. Em seguida, verifica-se se a porta já está aberta e, se necessário, ela é fechada antes de uma nova tentativa de abertura, prevenindo conflitos. Após a abertura da porta no modo leitura, a função *LerDados()* cria um *buffer* de 13 bytes para armazenar os dados recebidos. A leitura é validada por duas verificações: (i) se o tempo de leitura excedeu um limite (*Timeout*) ou a quantidade de *bytes* lidos foi inferior a 13, e (ii) se o quadro de comunicação contém valores válidos. Caso ocorra erro na leitura, a função *FlushSerial()* limpa o *buffer* de entrada para evitar interferências nas próximas leituras. Se os dados forem válidos, eles são copiados para a estrutura *ParametrosLidos* e processados para calcular o ângulo (graus) e a cadência (RPM). Esses valores são acessíveis em *Blueprints* devido à macro *USTRUCT*, permitindo integração direta com elementos 3D na UE. A função *AbrirPortaSerial()* configura a comunicação serial, definindo a porta COM, o modo de leitura e o baud rate de 115200 bps. Já a função *FecharPortaSerial()* é acionada ao encerrar o ambiente RV, limpando o *buffer* e cancelando operações pendentes antes do fechamento da porta.

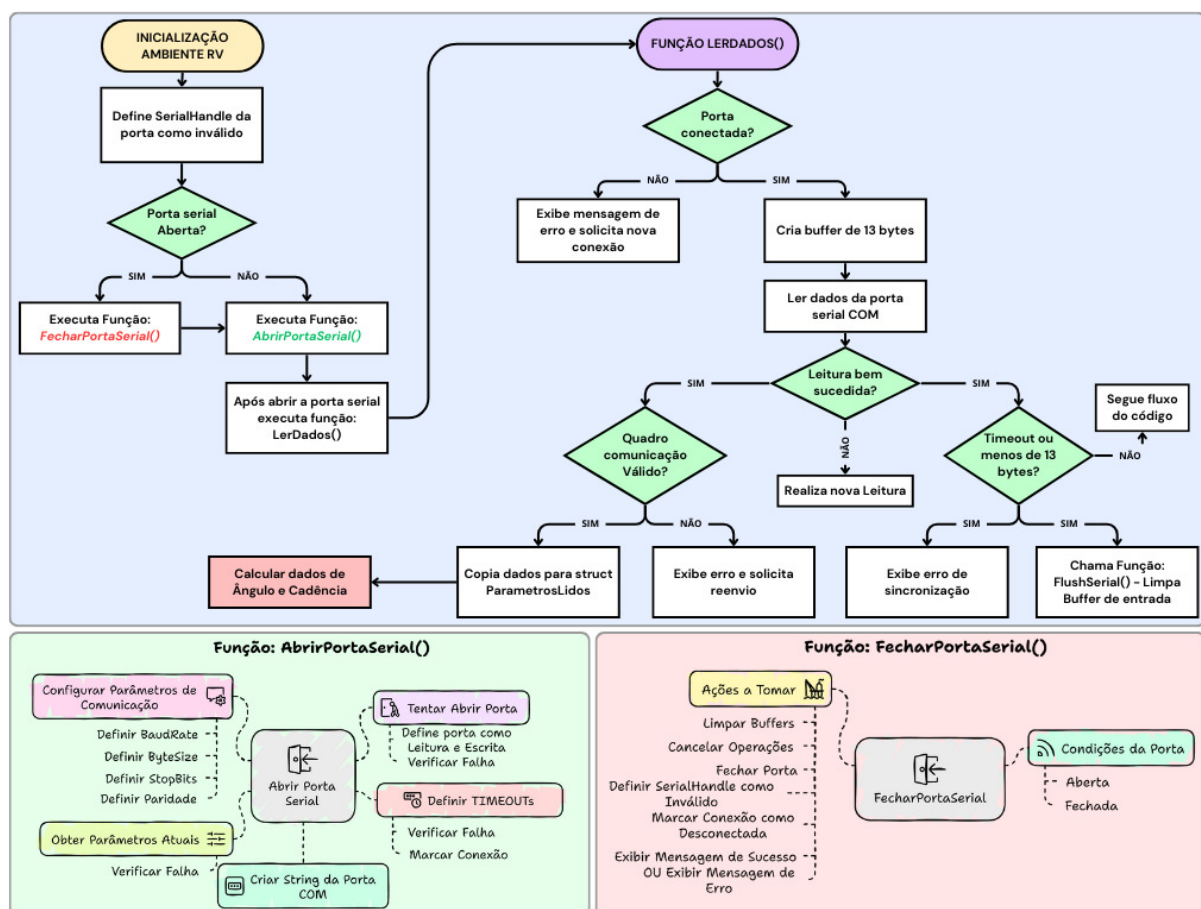


Figura 3. Fluxograma do código de recepção de dados bluetooth: O programa em C++ para a recepção de dados é formado por três funções principais que são *AbrirPortaSerial()*, *FecharPortaSerial()* e função *LerDados()*. As funções *AbrirPortaSerial()* e *FecharPortaSerial()* são responsáveis pela configuração e acesso aos dados recebidos na porta serial, assim como garantir que a porta seja acessada e encerrada corretamente evitando possíveis conflitos se manter a porta aberta ao se encerrar a execução do ambiente em RV. A função *LerDados()* realiza leitura dos 13 bytes e já posiciona cada byte em sua respectiva posição para ficar de acordo com o quadro de comunicação projetado.

Já no fluxo lógico do programa de envio de dados chamado *SendSerialData*. Inicialmente, o usuário define os parâmetros de eletroestimulação a serem transmitidos ao circuito de estimulação FES. Quando a interface em RV é

inicializada, a função *AbrirPortaSerial()* é executada no modo de escrita. Um *buffer* de 78 bytes é criado para armazenar os parâmetros correspondentes aos oito canais, onde o *byte* da posição 0 assume o valor 2 para indicar o início do quadro de comunicação. Os valores definidos pelo usuário na *Blueprint* são organizados em suas respectivas posições dentro do quadro de comunicação e enviados via porta serial. Caso a transmissão de todos os *bytes* seja bem-sucedida, uma mensagem de confirmação é exibida na interface RV.

Os 13 bytes de dados transmitidos pela placa FES são enviados a uma frequência de 10 Hz (100 ms). Para garantir a correta recepção desses dados na interface de RV, o programa de recepção em C++ deve operar em sincronia com a transmissão da placa. Essa sincronização é implementada por meio da programação em *Blueprint*, conforme ilustrado na Figura 4. No início do ambiente RV (*BeginPlay*), uma interface gráfica baseada em *widgets* da UE é criada para plotar, em tempo real, os gráficos de ângulo e cadência. Em seguida, um *Timer* é inicializado, acionando a *Blueprint* de recepção de dados a cada 100 ms (10 Hz). No encerramento da interface RV (*EndPlay*), o *Timer* é desativado e a contagem interrompida.

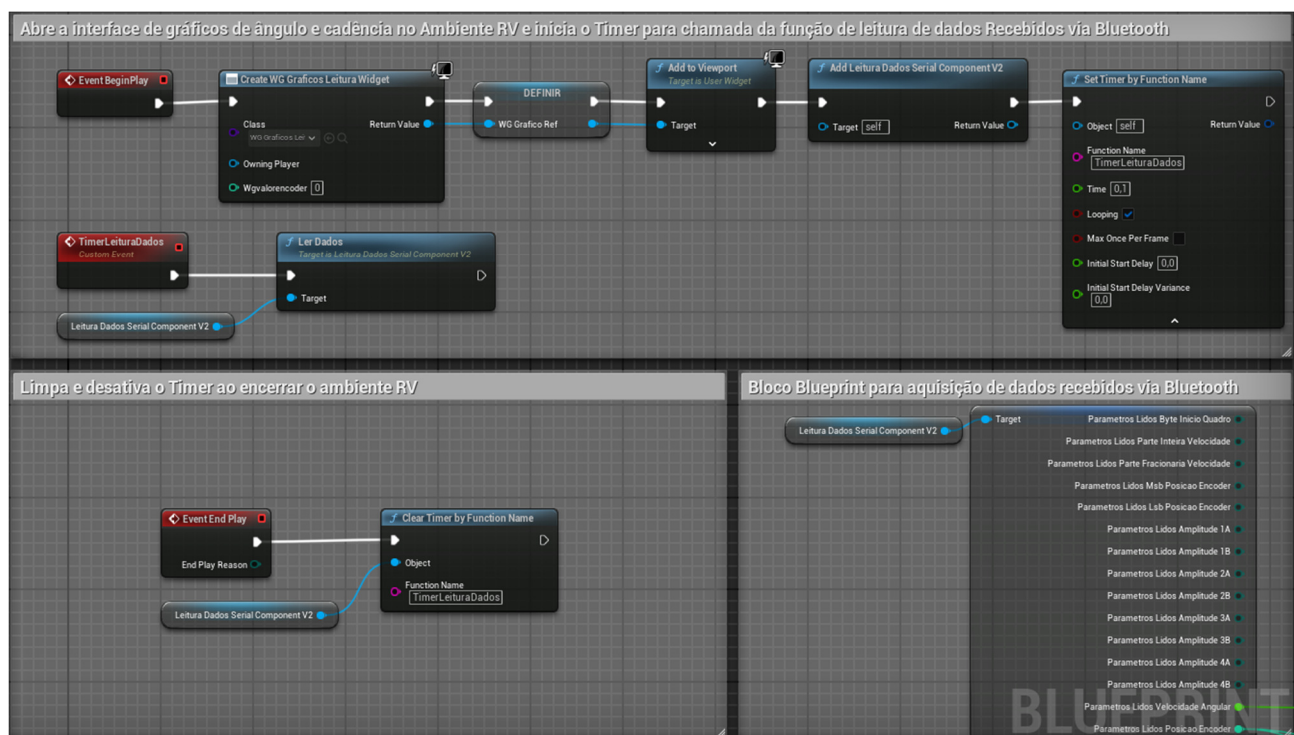


Figura 4. Programação *blueprint* para execução do programa de recebimento de dados a cada 100 ms: A programação em *blueprint* é utilizada para abrir a interface com o gráfico de ângulo e cadência e inicializar o *timer* para a chamada da *blueprint* de recepção de dados a cada 100 ms, permitindo que os dados recebidos sejam acessados no ambiente RV a uma frequência de 10Hz.

A Figura 5 ilustra a programação em *Blueprint* integrada à *SendSerialData*, onde o envio dos parâmetros de estimulação é acionado ao pressionar a tecla E.

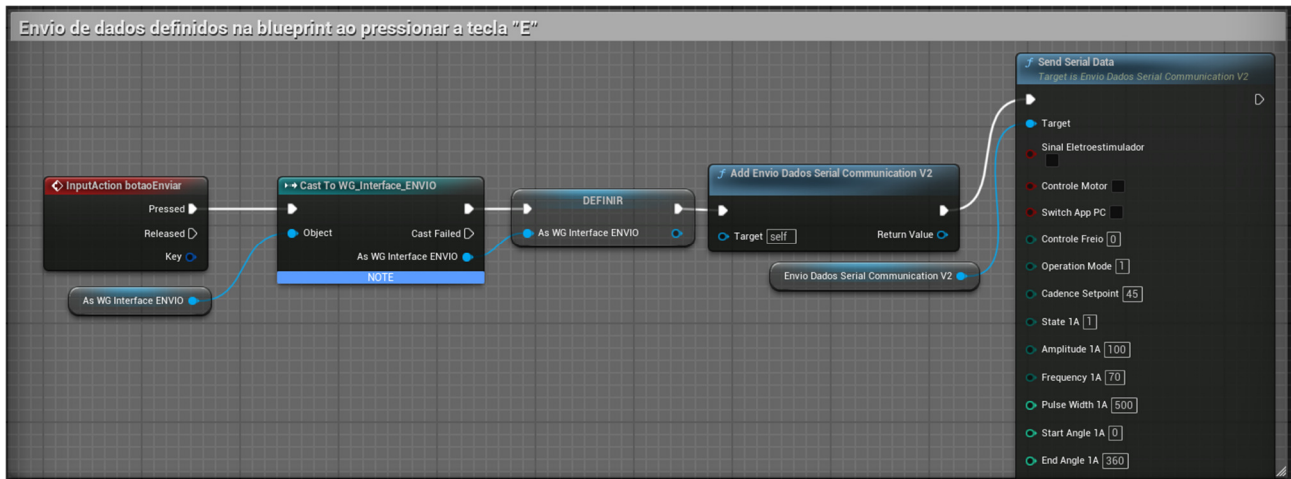


Figura 5: Programação *blueprint* utilizada para definir a entrada do teclado como evento para executar a *blueprint* *SendSerialData*. Permitindo o reenvio dos parâmetros configurados toda vez que a tecla “E” é pressionada.

Figura 6 apresenta a interface em RV, onde o nó *Parametros Lidos PosicaoEncoder* da *Blueprint* de recepção de dados permite sincronizar o giro de um objeto 3D com o ângulo de pedalada real. Além disso, um gráfico exibe a variação angular de 0° a 360° , sendo automaticamente zerado pelo sensor *zero cross* do encoder para evitar acúmulo de erros. O gráfico de cadência tem escala máxima de 80 RPM, embora os valores observados na plataforma sejam em torno de 45 RPM.

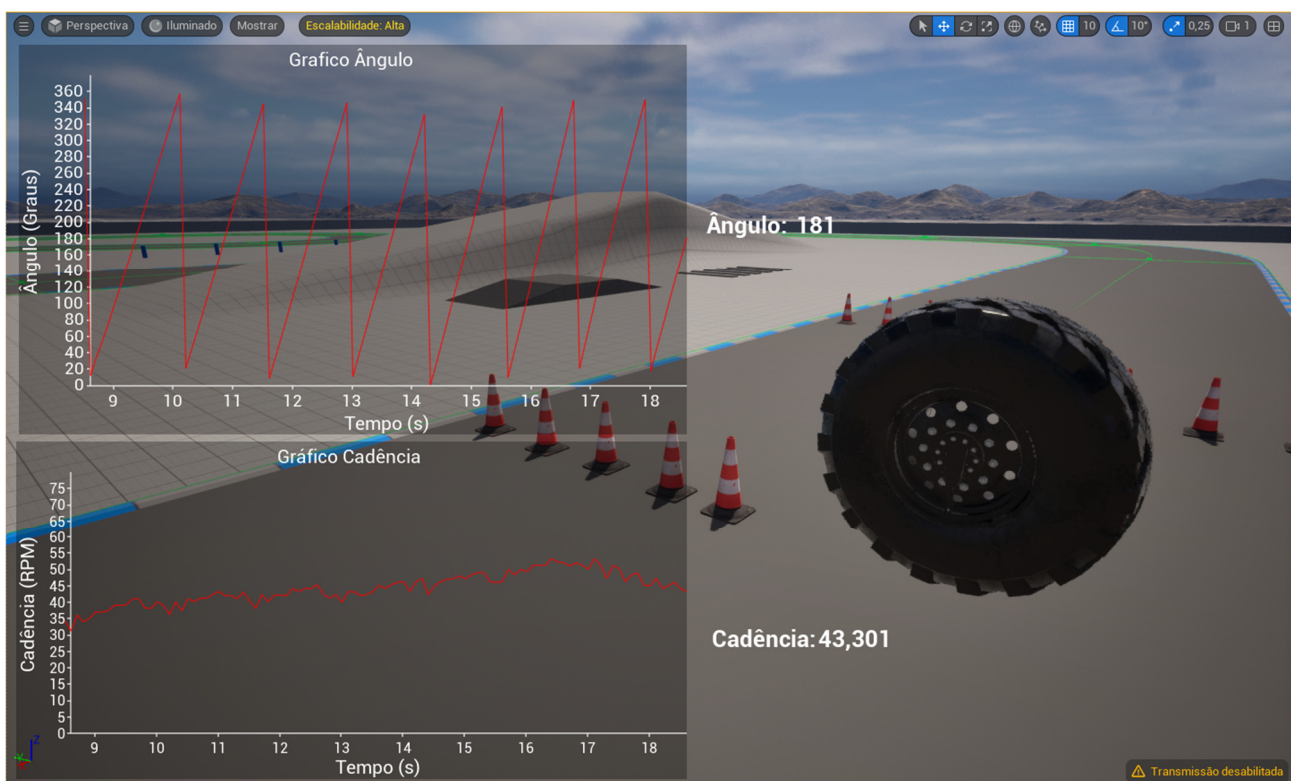


Figura 6. Representação da interface de RV: Os dados de ângulo que são enviados a um ambiente de uma pista de corrida em RV são utilizados para gerar o movimento de giro de um objeto 3D de roda que é sincronizado com o movimento real de pedalada do usuário por uma frequência de 10 Hz. Também são plotados em tempo real um gráfico de ângulo e cadência a partir dos dados recebidos pela *blueprint* de recepção de dados.

A análise do tempo de comunicação dos programas de envio e recebimento de dados em C++ foi realizada com analisadores lógicos e o software Saleae Logic 1.1.30. Observou-se uma latência de *0,97 ms* entre quadros de comunicação consecutivos, considerando uma taxa de transmissão de *100 ms*.

5. Discussão e Considerações Finais

O desenvolvimento da plataforma *V-FES-Cycling* demonstrou potencial na integração entre a FES e a RV para a reabilitação de pacientes com LME. A interface de comunicação projetada possibilitou a sincronização em tempo real dos dados do ângulo de pedalada, adquiridos pelo sensor encoder, com o ambiente virtual, proporcionando maior imersão e interatividade. Esse avanço diferencia-se de estudos anteriores que, em sua maioria, utilizaram apenas sistemas comerciais sem integração completa entre hardware e software ou baseados em RV semi-imersiva. Adicionalmente, a interface desenvolvida na UE permitiu ajustes em tempo real dos parâmetros de estimulação, possibilitando personalização do tratamento dos pacientes ao treinamento.

Desafios futuros incluem o aprimoramento do algoritmo de comunicação para maior precisão e a minimização dos atrasos na comunicação *Bluetooth*. Melhorias planejadas envolvem o aumento da taxa de atualização do sistema para valores superiores a *10 Hz*, um ambiente virtual mais realista e imersivo, e a reestruturação do protocolo de comunicação para integração de um freio eletromagnético, permitindo a simulação de diferentes resistências de pedalada em terrenos virtuais. Além disso, será implementado um sistema de segurança para desativação automática da eletroestimulação por software e por botão de emergência, aumentando a confiabilidade da plataforma. Para estimular a motivação dos usuários, um *exergame* de corrida em RV será desenvolvido, e estudos futuros avaliarão sua usabilidade e impacto na adesão e motivação de indivíduos com LME. No entanto, a eficácia clínica dessa plataforma ainda precisa ser avaliada em ensaios com pacientes, para validar seus benefícios e possíveis uso com uma ferramenta para reabilitação motora de pacientes com LME.

Financiamento: Essa pesquisa foi financiada pela Fundação de Apoio à Pesquisa do Distrito Federal (FAPDF) por meio do Edital 10/2023 - Programa FAPDF Learning, projeto No. 00193-00002152/2023-95.

Agradecimentos: Os autores agradecem a todos os participantes que contribuíram para este estudo, bem como à equipe da plataforma BEM-TE-VI.

Conflito de interesse: Os autores declaram não haver conflito de interesses. Os financiadores não tiveram nenhuma função no desenho do estudo; na coleta, análise ou interpretação dos dados; na redação do manuscrito ou na decisão de publicar os resultados.

Referências

1. CUESTA-GOMEZ, A. et al. Effects of virtual reality associated with serious games for upper limb rehabilitation in patients with multiple sclerosis: randomized controlled trial. *Journal of NeuroEngineering and Rehabilitation*, Springer Nature, v. 17, p. 90, 2020. Acesso em: 13 Dez. 2024. Disponível em: <<https://doi.org/10.1186/s12984-020-00718-x>>.
2. CYRINO, G. F. HARPYGAME: Um Jogo Sério Customizável com Interface Multimodal para Reabilitação de Indivíduos Pós-AVE. Dissertação (Dissertação de Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Brasil, Março 2019. Acesso em: 23 Nov. 2024. Disponível em: <<http://dx.doi.org/10.14393/ufu.di.2019.1286>>.
3. DUFFELL, L. D. et al. The effects of FES cycling combined with virtual reality racing biofeedback on voluntary function after incomplete SCI: a pilot study. *Journal of NeuroEngineering and Rehabilitation*, Springer, v. 16, p. 1–15, 2019. Acesso em: 16 Dez. 2024. Disponível em: <<https://doi.org/10.1186/s12984-019-0619-4>>.
4. ESFAHLANI, S. S. et al. ReHabgame: A non-immersive virtual reality rehabilitation system with applications in neuroscience. *Heliyon*, Elsevier Ltd, v. 4, 2 2018. ISSN 24058440. Acesso em: 02 Fev. 2024.
5. FATTAL, C. et al. Training with FES-assisted cycling in a subject with spinal cord injury: Psychological, physical and physiological considerations. *The Journal of Spinal Cord Medicine*, Taylor & Francis, v. 43, n. 3, p. 402–413, 2020. Acesso em: 04 Jan. 2025. Disponível em: <<https://doi.org/10.1080/10790268.2018.1490098>>.
6. HTC Corporation. HTC Vive Pro 2 – Overview. 2025. Acesso em: 06 Jan. 2025. Disponível em: <<https://www.vive.com/us/product/vive-pro2/overview/>>.
7. MAGALHÃES, T. C. Development and Validation of a High-Capacity Functional Electrical Stimulation System and of a Reinforcement Learning Algorithm Applied to the FES-Assisted Cycling Modality for a Participant with Paraplegia. Tese (Tese (Doutorado em Engenharia Elétrica)) — Universidade Federal de Minas Gerais, Escola de Engenharia, Belo Horizonte, Brazil, 2022. Acesso em: 05 Out. 2024. Disponível em: <<http://hdl.handle.net/1843/48539>>.

8. MOLINA, K. I. et al. Virtual reality using games for improving physical functioning in older adults: a systematic review. *Journal of NeuroEngineering and Rehabilitation*, BioMed Central, v. 11, p. 156, 2014. Acesso em: 06 Jan. 2025. Disponível em: <<http://www.jneuroengrehab.com/content/11/1/156>>.
9. MONGEOT, L. B. de et al. Combining FES and Exoskeletons in a Hybrid Haptic System for Enhancing VR Experience. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 31, p. 4812–4820, 2023. Acesso em: 10 Jan. 2025.
10. ROJO, A. et al. PedaleoVR: Usability study of a virtual reality application for cycling exercise in patients with lower limb disorders and elderly people. *PLOS ONE*, Public Library of Science, v. 18, n. 2, p. 1–20, 02 2023. Acesso em: 10 Dez. 2024. Disponível em: <<https://doi.org/10.1371/journal.pone.0280743>>.
11. SANTANA, L. et al. Neuromuscular disorders in women and men with spinal cord injury are associated with changes in muscle and tendon architecture. *The Journal of Spinal Cord Medicine*, Taylor & Francis, v. 46, n. 5, p. 742–752, 2023. Acesso em: 11 Jan. 2025. Disponível em: <<https://doi.org/10.1080/10790268.2022.2035619>>.
12. SOUSA, H. de; BALBINO, S. Development of Immersive Virtual Reality Interface for Lower-Limb Robotic Rehabilitation. In: 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). [S.l.: s.n.], 2018. p. 124–128. Acesso em: 23 Nov. 2024.