UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Transferência Sim-to-Real em Robótica com Aprendizado por Reforço

Estudo Comparativo entre Fidelidade Física e Domain Randomization

Dayane Rodrigues

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

DAYANE RODRIGUES

Transferência Sim-to-Real em Robótica com Aprendizado por Reforço

Estudo Comparativo entre Fidelidade Física e Domain Randomization

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): DAYANE RODRIGUES

Título do trabalho: Transferência Sim-to-Real em Robótica com Aprendizado por Reforço

Estudo Comparativo entre Fidelidade Física e Domain Randomization

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Dayane Rodrigues**, **Discente**, em 04/02/2026, às 16:34, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 13/03/2026, às 11:27, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

DAYANE RODRIGUES

Transferência Sim-to-Real em Robótica com Aprendizado por Reforço

Estudo Comparativo entre Fidelidade Física e Domain Randomization

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

RODRIGUES, DAYANE
Transferência Sim-to-Real em Robótica com Aprendizado por Reforço [manuscrito]: Estudo Comparativo entre Fidelidade Física e Domain Randomization / DAYANE RODRIGUES. - 2025.
90 f.: 2025

Orientador: Prof. Dr. Fernando Marques Federson
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. Inteligência Artificial. 2. Robótica. 3. Transferência Sim-to-real.

I. Federson, Fernando Marques , orient. II. Título.

CDU 004

DAYANE RODRIGUES

Transferência Sim-to-Real em Robótica com Aprendizado por Reforço

Estudo Comparativo entre Fidelidade Física e Domain Randomization

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 09 de dezembro de 2025.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Profa. Dra. Telma Woerle de Lima Soares
(INF-UFG)

DAYANE RODRIGUES

Transferência Sim-to-Real em Robótica com Aprendizado por Reforço

Estudo Comparativo entre Fidelidade Física e Domain Randomization

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Aprendizado por Reforço para Sim-to-Real na Robótica**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: Inteligência artificial; Robótica; Transferência sim-to-real.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Reinforcement Learning for Sim-to-Real in Robotics**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: Artificial intelligence; Robotics; Sim-to-real transfer.

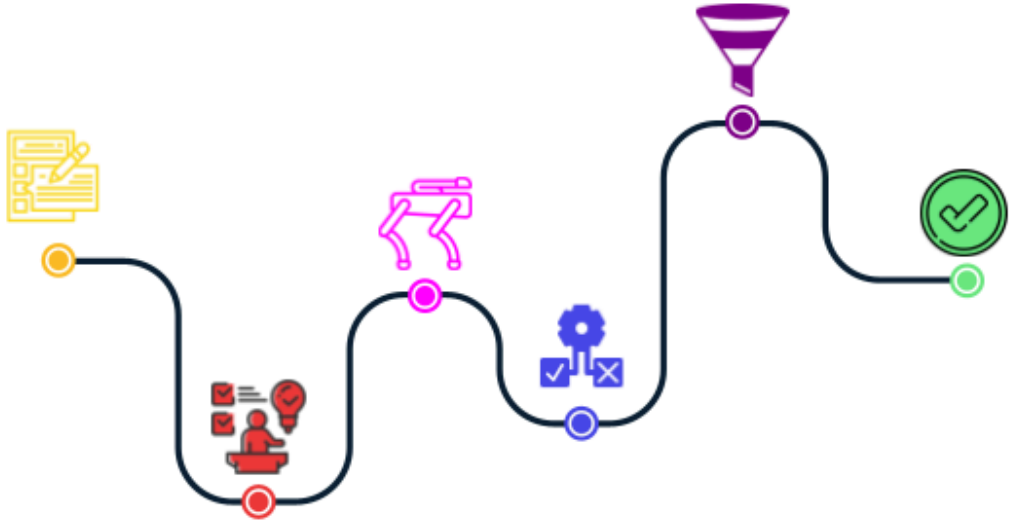
Goiânia

2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada

Dayane Rodrigues



Especialista em:
Aprendizado por Reforço para Sim-to-Real na Robótica

Semanas 1 a 3

Definição e Escopo: Revisão Sistemática no Parsifal e definição do foco em Sim-to-Real.

Semanas 4 e 5

Fundamentação: Mapeamento de 38 artigos e identificação das técnicas de Sim-to-Real.

Semanas 6 e 7

Arquitetura: Definição do robô Unitree Go2 e estudo dos frameworks (MuJoCo vs. Isaac).

Semana 8

Estratégia de Controle: Análise comparativa entre Controle Clássico (NMPC) e RL Híbrido.

Semana 9

Engenharia de Gaps: Dissecação do Estimador de Estado (EKF) e teoria de ADR.

Semana 10

Validação Experimental: Testes práticos: Falha na Fidelidade vs. Sucesso na Robustez (ADR).

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINHA JORNADA

Nome: **Dayane Rodrigues**

Especialidade: **Aprendizado por Reforço para Sim-to-Real na Robótica**

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha Jornada iniciou-se na **Semana 1** impulsionada por um interesse pessoal em dotar robôs de autonomia real, o que guiou a análise das chamadas de conferências internacionais como a ICAI (*International Conference on Artificial Intelligence*) e ICDATA (*International Conference on Data Science*). A partir dessa investigação, defini meu objeto de pesquisa na interseção entre Aprendizado por Reforço (**RL** - *Reinforcement Learning*) e Robótica, utilizando ferramentas de IA generativa. O *ResearchRabbit* serviu para identificar leituras basilares, como o artigo de Nagaraj et al. (2022). Contudo, durante a **Semana 2**, ao aprofundar nessas leituras e tentar construir um histograma histórico da área, deparei-me com uma inconsistência significativa nas cronologias existentes, o que revelou a necessidade de uma investigação própria mais rigorosa. Os detalhes dessa fase exploratória e as análises das inconsistências históricas estão no **Apêndice 1**. Essa percepção levou-me a configurar, ainda na **Semana 2**, uma revisão sistemática na plataforma *Parsifal*, estruturando o protocolo PICOC (População, Intervenção, Comparação, *Outcome* e

[1]Dez Semanas, entre setembro de 2025 e dezembro de 2025.

Contexto). Na **Semana 3**, a leitura preliminar dos resumos indicou que o tema geral era vasto demais, motivando um refinamento estratégico do escopo para a transferência **Sim-to-Real** (Simulação para o Real). Ajustei então a *string* de busca para incluir termos como *domain randomization* (randomização de domínio), resultando na extração de 99 artigos de bases de dados de alto impacto como a IEEE (*Institute of Electrical and Electronics Engineers*) e a ACM (*Association for Computing Machinery*), além da identificação de trabalhos cruciais sobre *Guided RL* (RL Guiado), consolidando a base da pesquisa detalhada no **Apêndice 2**.

Nas **Semanas 4 e 5**, realizei o mapeamento técnico das ferramentas e estratégias centrais da área. Especificamente na **Semana 4**, a análise dos trabalhos permitiu identificar que a integração Sim-to-Real se apoia majoritariamente em frameworks como MuJoCo, Gazebo e ROS/ROS2, aplicados em tarefas de manipulação e locomoção com algoritmos de *Deep RL* como PPO e SAC. Já na **Semana 5**, avancei para a análise consolidada de 38 artigos, onde o ponto focal foi o confronto entre abordagens para mitigar o "reality gap", ou seja, a discrepância de comportamento físico e sensorial entre o ambiente simulado e o mundo real. Enquanto a obra *Overcoming the Sim-to-Real Gap in Autonomous Robotic Systems* (2022) defende o uso de Gêmeos Digitais calibrados para transferências diretas, o artigo *DROPO :Sim-to-real transfer with offline domain randomization and policy optimization* (2023) foi decisivo para minha jornada. Ele demonstrou que a robustez não exige necessariamente um simulador perfeito, mas sim o uso estratégico de Domain Randomization (DR). Essa descoberta redirecionou o foco da busca pela fidelidade absoluta para a exploração da robustez, fundamentando as escolhas registradas no **Apêndice 3**.

Durante as **Semanas 6 e 7**, a pesquisa transitou da teoria para a definição da arquitetura experimental. Na **Semana 6**, a sistematização da literatura permitiu categorizar as ferramentas ideais: identifiquei que simuladores como *MuJoCo* e *Isaac Gym* lideram em precisão física para locomoção, enquanto o *ROS2* e *Stable-Baselines3* consolidam-se como o padrão para integração e algoritmos. Foi neste momento que defini o robô quadrúpede **Unitree Go2 EDU** como campo de aplicação. Esta escolha é estratégica e motivada não apenas pela sua relevância técnica, mas também pela disponibilidade física do equipamento no Instituto de Informática (INF/UFG), visando uma futura validação real. Já na **Semana 7**, o

foco mudou para a preparação do ambiente, onde enfrentei desafios práticos de compatibilidade, como, por exemplo, de *drivers* de GPU na configuração do *Isaac Gym*. Para mitigar os riscos de operar um robô físico de alto custo, aprofundei-me em *Safe RL* (Aprendizado por Reforço Seguro) através do estudo do pipeline *Sim-to-Lab-to-Real*, compreendendo como técnicas de *shielding* (blindagem de ações) e *PAC-Bayes Control* podem garantir a integridade do hardware. Além disso, revisei a evolução do *Domain Randomization* para sua versão automática (**ADR**). Conclui ser a abordagem mais promissora para terrenos imprevisíveis. As especificações do robô, o mapeamento dos frameworks e as anotações sobre segurança estão detalhados no **Apêndice 4**.

A **Semana 8** foi um marco de definição estratégica na Minha Jornada. Enquanto aguardava a disponibilização da infraestrutura de *hardware* de alta performance (GPU RTX 4090) necessária para simulações fotorrealistas, dediquei-me à análise comparativa profunda entre os paradigmas de controle, dissecando os repositórios *Unitree Go2 Digital Twin* e *Legged Control*. Durante este período, foi possível observar o contraste prático entre o uso de *frameworks* modernos de IA (como *Isaac Lab* e *Stable-Baselines3* aplicando algoritmos PPO) e as abordagens clássicas baseadas em modelo (utilizando *OCS2* e *Pinocchio* para NMPC - *Nonlinear Model Predictive Control*). O resultado principal desta análise foi a compreensão de que o controle clássico, embora ofereça estabilidade física rigorosa, apresenta rigidez em terrenos desconhecidos, enquanto o Aprendizado por Reforço (*Model-Free*) privilegia a generalização e a robustez. Essa constatação fundamentou minha decisão de não apenas escolher um método, mas investigar como essas abordagens se complementam, estruturando a base para os testes futuros. Os detalhes desta comparação técnica e a arquitetura dos nós de controle estão registrados no **Apêndice 5**.

Na **Semana 9**, mergulhei na engenharia dos "gaps" que separam a teoria dos experimentos. Utilizando o repositório *Deploy an RL Policy on the Unitree Go2 Robot* como base de estudo no MuJoCo, enfrentei um "gap" real: dificuldades severas de compilação no *ROS2 Humble* (especificamente no pacote *rosidl_typesupport_c*), que exigiram múltiplas iterações de configuração. Superada essa barreira, dissequei o funcionamento do Estimador de Estado, especificamente o Filtro de Kalman Estendido (**EKF**), compreendendo que a

política de controle (RL) não deve ser treinada com o *ground truth* (dados perfeitos) da simulação, mas sim com estimativas ruidosas que combinam IMU e encoders, simulando a cegueira parcial do robô real. Simultaneamente, aprofundei a teoria de **Automatic Domain Randomization (ADR)** para terrenos complexos, identificando quais parâmetros físicos (atrito, massa) podem ser randomizados com segurança. Os diagramas do estimador EKF e o planejamento dos parâmetros de randomização também estão disponíveis no **Apêndice 5**.

A **Semana 10** representou a validação experimental e a síntese de toda a Residência. Diante de uma janela de oportunidade de acesso a uma GPU de alta performance, executei duas trilhas de testes paralelas para validar minhas hipóteses. Antes disso, uma tentativa de usar *System Identification* (SysID) falhou, provando ser um "buraco de coelho" de calibração. A primeira trilha, focada em fidelidade máxima no MuJoCo (CPU), falhou em testes de estresse (escadas), demonstrando o overfitting do modelo. A segunda trilha, utilizando treinamento em larga escala com 4096 robôs e ADR no *Isaac Lab* (GPU), gerou uma política robusta capaz de ser generalizada para terrenos complexos não vistos. Esta validação experimental confirmou a superioridade da abordagem de robustez para o problema de Sim-to-Real. Os logs de treinamento, as curvas de aprendizado e a tabela comparativa final dos experimentos estão consolidados no **Apêndice 6**.

Em função de tudo que vivi nesta Jornada, gostaria de deixar registrado que a especialização em Sim-to-Real não se encerra na obtenção de uma política de controle, mas na compreensão profunda dos limites entre a modelagem matemática e a realidade física. A pesquisa demonstrou que a busca pela simulação "perfeita" é muitas vezes menos frutífera do que a construção de agentes adaptáveis à variabilidade estocástica. Embora a validação no robô físico não tenha sido possível nesta etapa, o framework metodológico desenvolvido está pronto para guiar essa próxima fase crítica de implantação em hardware.

Primeiramente, agradeço a mim mesma pela resiliência de nunca desistir, mantendo a convicção de que a dedicação inabalável e a insistência seriam a única forma de alcançar meus sonhos. À minha mãe, **Sueli**, que foi mãe e pai, obrigada por acordar comigo às 5 da manhã todos os dias do meu primeiro ano, para garantir minha segurança no ponto de ônibus e ser meu porto seguro, assegurando-me de que, independente das minhas escolhas, eu sempre terei um lar para onde voltar. Às minhas irmãs: **Danielle**, por me

apresentar o Bacharelado em IA e me inspirar a seguir este caminho, apoiando-me em tudo, desde fechar malas correndo e fazer maquiagem em eventos, até desabafar sobre as dificuldades da vida; e à minha gêmea **Deborah**, pois estamos juntas literalmente desde o princípio, dividindo a vida antes mesmo de nascermos. Não foi uma escolha minha, obviamente, mas estar sempre com você e agora dividir a profissão faz com que qualquer desafio se torne mais leve e a caminhada menos solitária; ter vocês comigo é minha maior sorte.

Estendo minha profunda gratidão ao coordenador **Anderson**, que no primeiro período, com generosidade, emprestou-me seu próprio notebook para que eu conseguisse realizar as atividades do curso, e à professora **Telma**, por apoiar todos os projetos em que ingressei; agradeço também a todos os professores que me ensinaram com amor e carinho, moldando a profissional que sou hoje. Por fim, aos meus amigos do grupo **CLAMDER**, **Carlos, Letícia, André, Michael, Edward e Rabelo**. Vocês estiveram ao meu lado nos altos e baixos, enfrentando as disciplinas mais desafiadoras e os prazos mais curtos. A graduação não teria sido essa experiência extraordinária sem a presença, o apoio e a amizade de cada um de vocês.

APÊNDICE 1

Termo de Aceite de Entrega 1

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 3 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Foi realizada a análise das conferências ICAI'25 – The 27th Int'l Conf on Artificial Intelligence, ICDATA'25 – 21th Int. Conference on Data Science e ACC'25 – 9th Int. Conf. on Applied Cognitive Computing. A partir dessa análise, foram destacados os temas:

- Aprendizado e fusão de sensores adaptativos;
- Aprendizagem por reforço;
- Mineração de texto e dados semiestruturados;
- Representação de dados e conhecimento.

Após a avaliação, não foi escolhido um tema isolado, mas sim a junção de duas áreas, definindo como objeto de pesquisa **Aprendizado por Reforço aplicado à Robótica**.

Com o objeto de pesquisa definido, foi realizada uma investigação em IAs generativas (ChatGPT, Gemini, Claude, Perplexity) sobre a história da área. O levantamento indicou pioneiros e contribuições fundamentais, bem como livros de referência e artigos que marcaram o desenvolvimento do campo. A lista completa de publicações, obras e autores identificados, encontra-se no link: Complementos_GATES.

Adicionalmente, foi realizada uma busca no ResearchRabbit, que resultou em sete artigos diretamente relacionados à temática. Entre eles, destaca-se *A Concise Introduction to Reinforcement Learning in Robotics* (Nagaraj et al., 2022), por apresentar uma visão abrangente da aplicação de aprendizado por reforço no campo da robótica. Os demais artigos estão organizados na planilha complementar, disponível no link: Complementos_GATES

No levantamento também foram identificados dois vídeos no YouTube relevantes para o entendimento do tema:

- *A History of Reinforcement Learning* – A History of Reinforcement Learning
- *Machine Learning in Robots* – What Is Machine Learning For Robot Algorithms?

Por fim, foram encontrados dois sites que tratam do tema:

- *Introduction to Reinforcement Learning – A Robotics Perspective* – lamarr-institute

- *How does reinforcement learning apply to robotics?* – milvus.io

Os materiais reunidos servirão para a compreensão da história do objeto de pesquisa e para a construção de conhecimentos iniciais sobre a área.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Para a próxima entrega, pretende-se dar continuidade à organização do material levantado. As atividades previstas incluem:

- Elaboração de um histograma para representar de forma visual os conhecimentos já mapeados até o momento;
- Leitura e análise do artigo A Concise Introduction to Reinforcement Learning in Robotics (Nagaraj et al., 2022), a fim de consolidar os conceitos iniciais da área e direcionar os próximos passos da pesquisa;
- Utilização da plataforma Parsifal para a seleção sistemática de artigos, incluindo a configuração da ferramenta e a definição de critérios de seleção de publicações relacionadas ao aprendizado por reforço em robótica, com foco na futura revisão bibliográfica/literária.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Fase Exploratória (Termo de Entrega 1)

Revisão Histórica Inicial do Aprendizado por Reforço em Robótica

Este documento foi elaborado como complemento ao **Gate1_03-040925 semanal**, com o objetivo de registrar a primeira revisão exploratória sobre a história do aprendizado por reforço aplicado à robótica. A investigação foi conduzida por meio de consultas em IAs generativas (ChatGPT, Gemini, Claude, Perplexity), que forneceram um panorama cronológico e bibliográfico preliminar da área. A intenção é reunir referências iniciais que servirão de base para aprofundamentos futuros e para a definição de conceitos consistentes da pesquisa. O conteúdo não corresponde a uma validação definitiva, mas a um mapeamento inicial de autores, publicações e livros que marcaram o desenvolvimento da área desde a década de 1950, “considerado” o registro mais antigo encontrado.

As consultas apontaram que a área tem raízes em experimentos de controle adaptativo nos anos 1950 e 1960, passando por formulações teóricas e aplicações em controle nos anos 1980, até chegar às primeiras implementações consistentes em robôs reais nos anos 1990. Foram destacados pioneiros como *Marvin Minsky*, *Donald Michie*, *Richard Sutton* e *Andrew Barto*, além de contribuições importantes de *Mahadevan*, *Connell, Long-Ji Lin*, *Minoru Asada* e *Maja Matarić*. Também foram citados livros que moldaram o campo, como *Robot Learning* (1993) e *Reinforcement Learning: An Introduction* (1998), além de artigos marcantes que definiram linhas de pesquisa.

Principais publicações iniciais:

- 1951 — Marvin Minsky & Dean Edmonds, *SNARC (Stochastic Neural-Analog Reinforcement Calculator)* SpringerLink
- 1957 — Richard Bellman, *"Dynamic Programming"* - Princeton University Press (fundamento matemático)
- 1960s — Johns Hopkins Beast - Autômato móvel com navegação e recarga autônoma
- 1968 — Donald Michie & R. A. Chambers, *BOXES: An Experiment in Adaptive Control* Centre for Intelligent Machines
- 1983 — Andrew G. Barto, Richard S. Sutton & Charles W. Anderson, *Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems* coneural.org
- 1989 — Christopher J.C.H. Watkins, *"Learning from Delayed Rewards"* - PhD Thesis, King's College (Q-learning)

- 1990 — Christopher G. Atkeson, *"Locally Weighted Learning for Robot Control"* - NIPS
- 1991 — Sridhar Mahadevan & Jonathan Connell, *Scaling Reinforcement Learning to Robotics by Exploiting the Subsumption Architecture* ResearchGate
- 1991/1992 — Long-Ji Lin, *Programming Robots Using Reinforcement Learning and Teaching* / *Self-Improving Reactive Agents…* AAAI | SpringerLink
- 1992 — Christopher Watkins & Peter Dayan, *"Q-learning"* - Machine Learning Journal
- 1994 — Minoru Asada et al., *Vision-based soccer robots* - IROS (precursor da RoboCup)
- 1995/1996 — Minoru Asada et al., *Vision-Based Reinforcement Learning for Purposive Behavior Acquisition* IEEE
- 1994/1997 — Maja J. Matarić, *Reward Functions for Accelerated Learning* / *Reinforcement Learning in the Multi-Robot Domain* SpringerLink

Livros que destacaram na área:

- 1953 — W. Grey Walter, *"The Living Brain"* - W.W. Norton
- 1965 — Nils Nilsson, *"Learning Machines"*
- 1988 — Ronald C. Arkin, *"Motor Schema Based Navigation"*
- 1993 — Connell & Mahadevan (eds.), *Robot Learning* SpringerLink
- 1998 — Sutton & Barto, *Reinforcement Learning: An Introduction* Incomplete Ideas
- 1998 — Ronald C. Arkin, *Behavior-Based Robotics* cs.huji.ac.il

Principais pioneiros

- Richard Bellman — Programação dinâmica e equação de Bellman (1950s), base matemática fundamental para RL
- Marvin Minsky & Dean Edmonds — SNARC (1951)
- Donald Michie — BOXES (1968)
- Richard Sutton & Andrew Barto — ator-crítico (1983), livro de referência (1998)
- Christopher Watkins — Inventou Q-learning (1989), algoritmo fundamental para RL sem modelo
- Sridhar Mahadevan & Jonathan Connell — robôs comportamentais (1991–1992)
- Long-Ji Lin — RL em robôs com ensino e simulação (1991–1992)
- Minoru Asada — RL baseado em visão (1995–1996)
- Christopher Atkeson — Locally weighted learning para controle robótico (1990), ponte entre controle adaptativo e RL
- Maja J. Matarić — shaping de recompensas e multi-robôs (1994–1997)
- Jan Peters & Stefan Schaal — gradiente de política em robótica (2006)
- Sergey Levine & Pieter Abbeel — deep RL em robôs reais (2015–2016

Termo de Aceite de Entrega 2

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 11 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Na segunda Semana do GATE, foi realizada a leitura do artigo *A Concise Introduction to Reinforcement Learning in Robotics* (Nagaraj et al., 2022) apresentou uma visão geral do tema, abordando avanços e desafios de forma superficial para o objetivo de compreender conjuntamente aprendizado por reforço e robótica. Também foram lidos o resumo, a introdução e a conclusão dos sete artigos encontrados no ResearchRabbit, que se mostraram insuficientes para fornecer uma compreensão mais ampla. A análise detalhada do artigo, com desafios, propostas e relevância, está registrada na planilha complementar, e o relato descritivo da leitura encontra-se no documento complementar: Complementos_GATES Complementos_GATES

Em seguida, iniciou-se o desenvolvimento do histograma histórico do aprendizado por reforço. Foram buscadas representações já existentes, resultando em oito versões de linhas do tempo identificadas em artigos, blogs e publicações científicas. Contudo, essas representações não são consistentes entre si, variando entre foco em algoritmos, aplicações ou panoramas gerais. A heterogeneidade das fontes mostrou que não seria possível, em apenas uma semana, consolidar um histograma próprio sem aprofundamento em mais de 50 anos de evolução. A análise inicial dessas imagens está descrita no documento complementar: Complementos_GATES

Por fim, foi configurada a plataforma Parsifal para a revisão sistemática sobre aprendizado por reforço em robótica. O processo envolveu a definição de objetivos, a formulação do PICOC, a elaboração da questão de pesquisa, a escolha de palavras-chave e strings de busca, a indicação das bases de dados e o estabelecimento dos critérios de inclusão e exclusão, além da construção de uma checklist de avaliação da qualidade dos artigos.

No PICOC, foram definidos os seguintes elementos: a população corresponde a research in robotics; a intervenção é o use of reinforcement learning; a comparação não foi considerada aplicável; o outcome consiste em understanding the applications, challenges and advances of RL when applied to robotics; e o contexto abrange academic articles, reviews, and case studies in simulated and real-world environments.

A partir disso, foi elaborada a questão de pesquisa:
"Como o aprendizado por reforço tem sido aplicado à robótica, quais desafios têm sido encontrados e quais avanços têm sido reportados em ambientes simulados e reais?"

Na etapa de palavras-chave, foram listados termos diretamente relacionados ao tema central, como *RL*,

deep reinforcement learning e *reinforcement learning*, vinculados ao eixo de intervenção, e *robot* e *robotics*, vinculados ao eixo da população.

Com base nessas escolhas, foi construída a seguinte string de busca para aplicação nas bases: ("reinforcement learning" OR "deep reinforcement learning" OR "multi-agent reinforcement learning" OR "RL") AND (robotics OR "robotic manipulation" OR "mobile robot" OR "autonomous robot" OR "multi-robot systems")

As fontes de pesquisa selecionadas para a revisão foram: ACM Digital Library, APA Psycnet, EBSCO, IEEE Xplore, ScienceDirect e Scopus.

Por fim, foram definidos os critérios de seleção:

- Inclusão: artigos que tratam explicitamente de reinforcement learning em robótica, incluindo textos que discutam algoritmos, desafios ou avanços no uso de RL em tarefas de manipulação, navegação e locomoção.
- Exclusão: artigos que abordam aprendizado de máquina em robótica sem relação com RL, estudos que usem RL em áreas sem relação com robótica e trabalhos puramente teóricos sem aplicação prática ou discussão do contexto robótico.

Também foi elaborada uma checklist de avaliação da qualidade, com questões que verificam se o artigo aplica RL em robótica, se apresenta simulações ou implementações práticas, se descreve claramente o algoritmo usado e se expõe resultados obtidos.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Para a próxima entrega, pretende-se dar continuidade à revisão do Parsifal. As atividades previstas incluem:

- A extração dos arquivos das bases científicas selecionadas, iniciando a triagem dos artigos incluídos e excluídos de acordo com os critérios estabelecidos. Caso necessário, a string de busca será ajustada para ampliar ou refinar os resultados
- Em paralelo, será dada continuidade à análise dos histogramas existentes sobre a história do aprendizado por reforço, visando preparar as bases para a futura consolidação de uma linha temporal própria.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Análises da História de RL (Termo de Entrega 2)

Histograma dos marcos do Aprendizado por Reforço

Na Segunda semana, uma das propostas de atividades foi a “elaboração de um histograma histórico dos principais acontecimentos no aprendizado por reforço”. Para isso, foram realizadas buscas utilizando strings como *“reinforcement learning timeline histogram”*, *“RL history histogram”* e *“timeline of reinforcement learning milestones” no google*.

Como resultado, foram identificadas diferentes representações visuais em artigos, blogs e publicações científicas (Imagens 1 a 8). Essas figuras apresentam linhas do tempo de algoritmos, modelos visuais, avanços em RL aplicado a trading, RLHF e revisões históricas mais amplas. No entanto, a análise revelou que não há consenso entre as fontes quanto aos marcos escolhidos nem à forma de organizar os eventos. Algumas imagens priorizam algoritmos específicos (ex.: Q-learning, Policy Gradient, Deep Q-Networks), outras enfatizam áreas de aplicação (trading, visão computacional, interação robótica) e algumas apresentam panoramas bastante gerais, como pode ser analisado abaixo nas imagens:

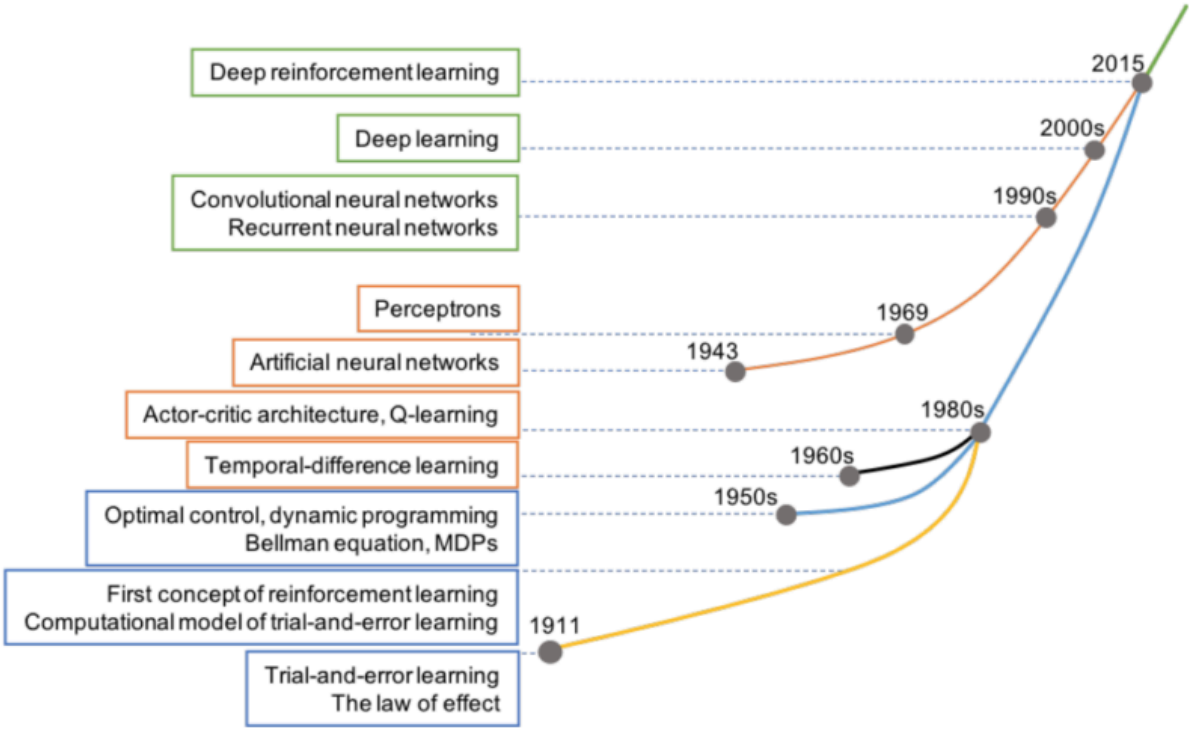


Fig. 1: Emergence of deep RL through different essential milestones.

Imagem 1: https://alphaarchitect.com/reinforcement-learning-for-trading/

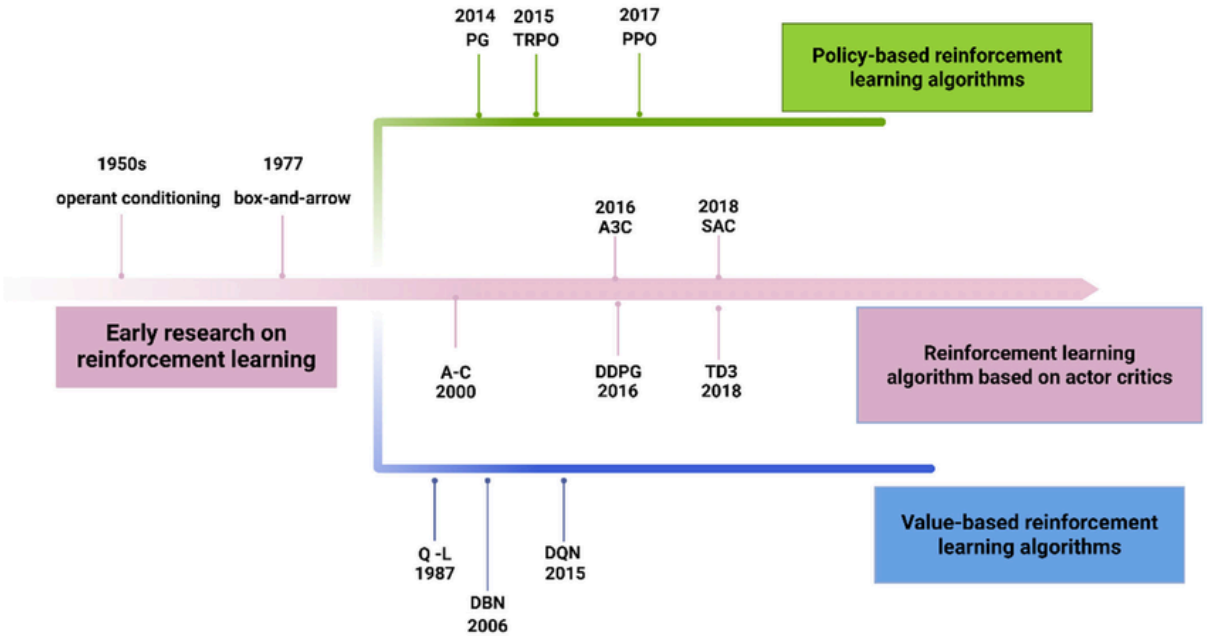


Imagem 2:
https://www.researchgate.net/figure/Timeline-of-reinforcement-learning-algorithm-development_fig5_392748014

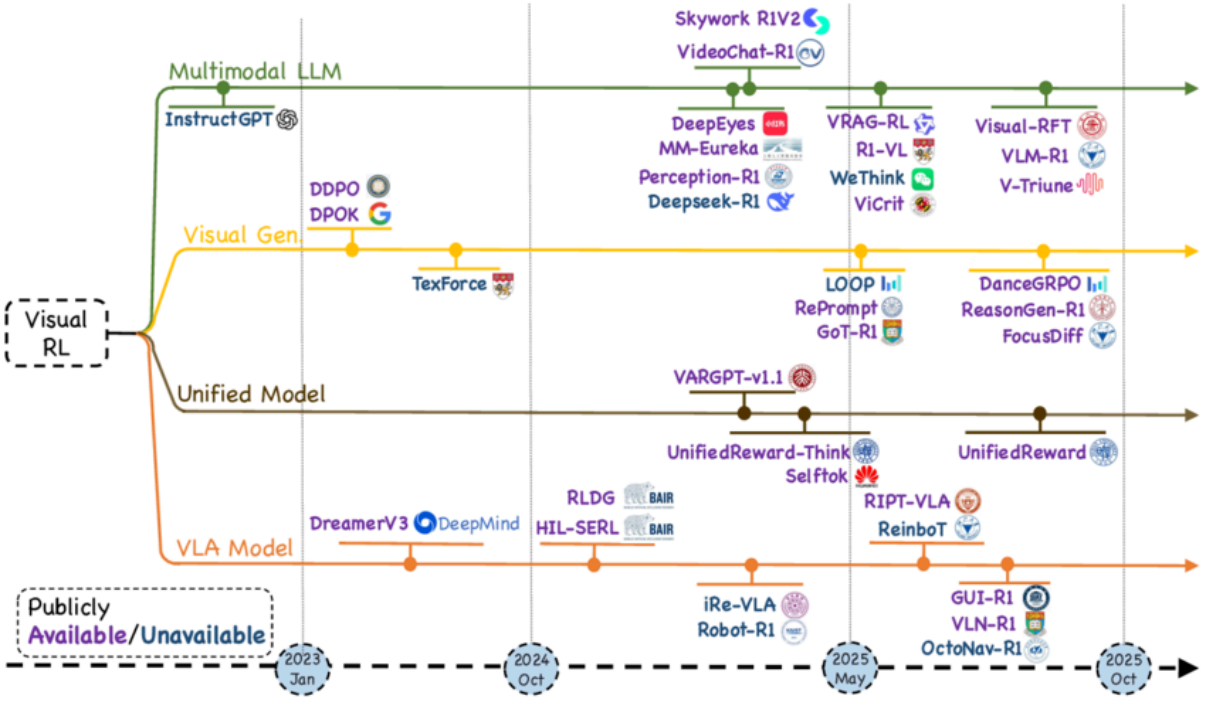


Imagem 3:
https://www.researchgate.net/figure/Timeline-of-Representative-Visual-Reinforcement-Learning-Models-The-figure-presents-a_fig1_394439107

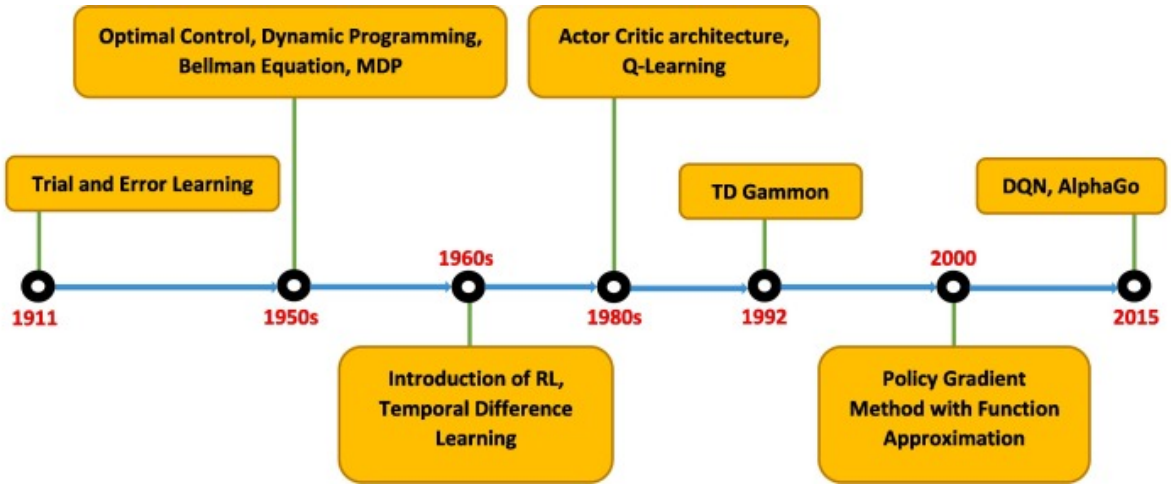


Imagem 4: https://i2.wp.com/researchdatapod.com/wp-content/uploads/2021/09/The-History-of-Reinforcement-Learning.png

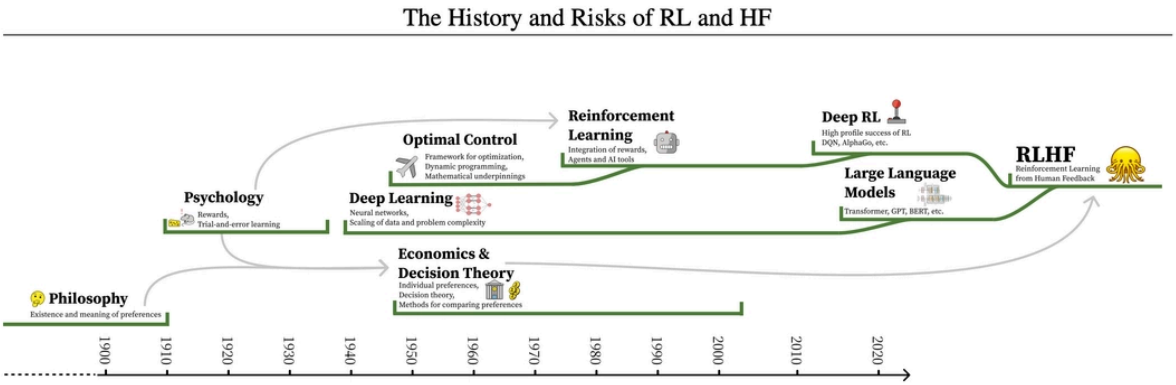


Figure 1: The timeline of the integration of various subfields into the modern version of RLHF. The direct links are continuous developments of specific technologies, and the arrows indicate motivations and conceptual links.

Imagem 5: https://klu.ai/glossary/rlhf

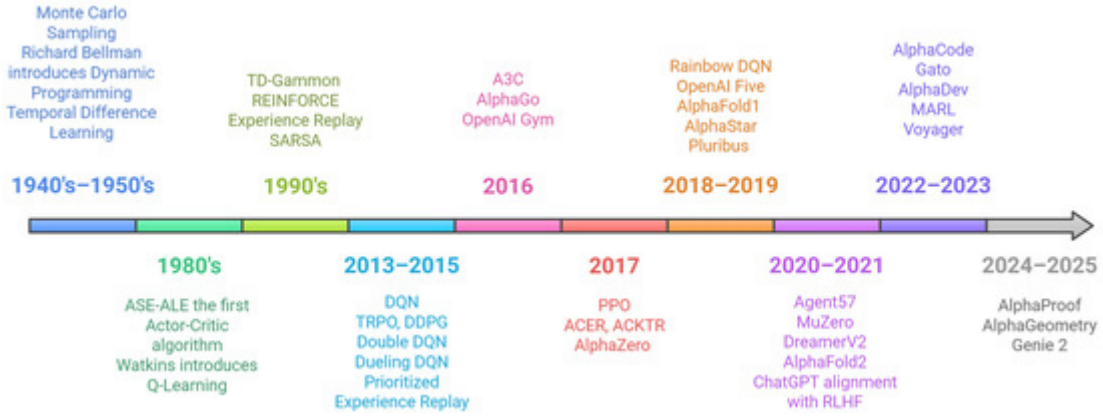


Imagem 6: https://www.mdpi.com/2673-2688/6/3/46

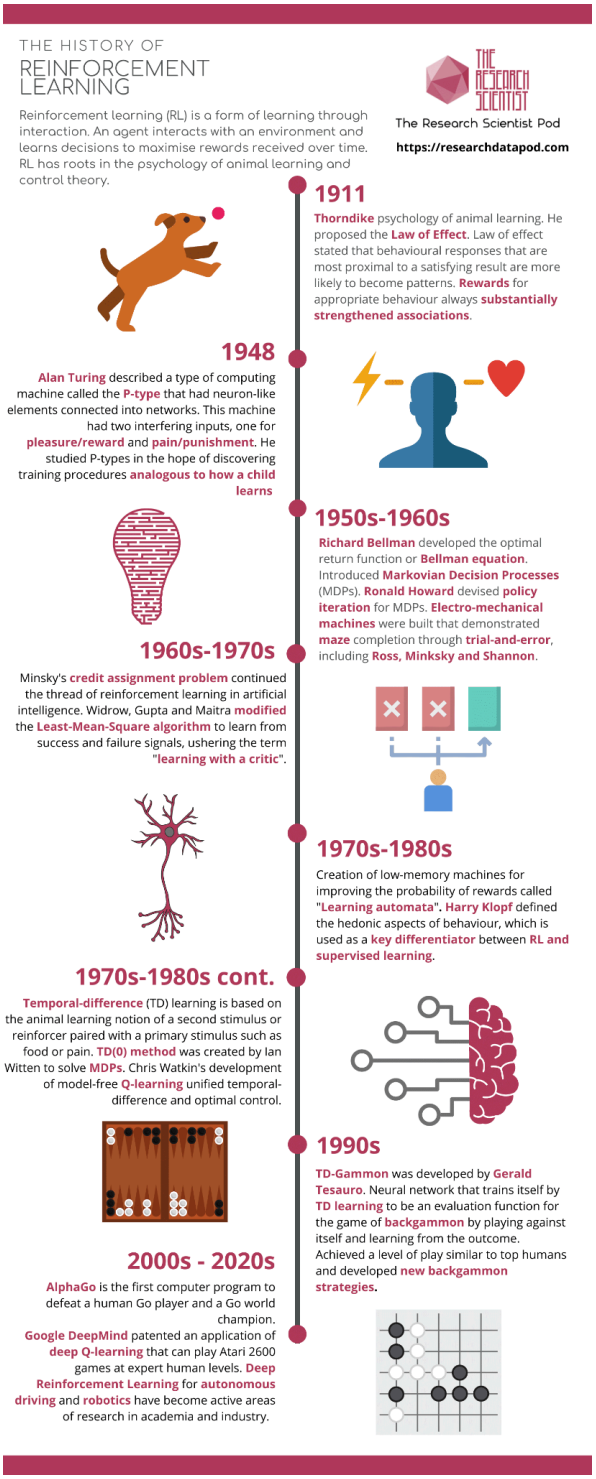


Imagem 7: https://i2.wp.com/researchdatapod.com/wp-content/uploads/2021/09/The-History-of-Reinforcement-Learning.png

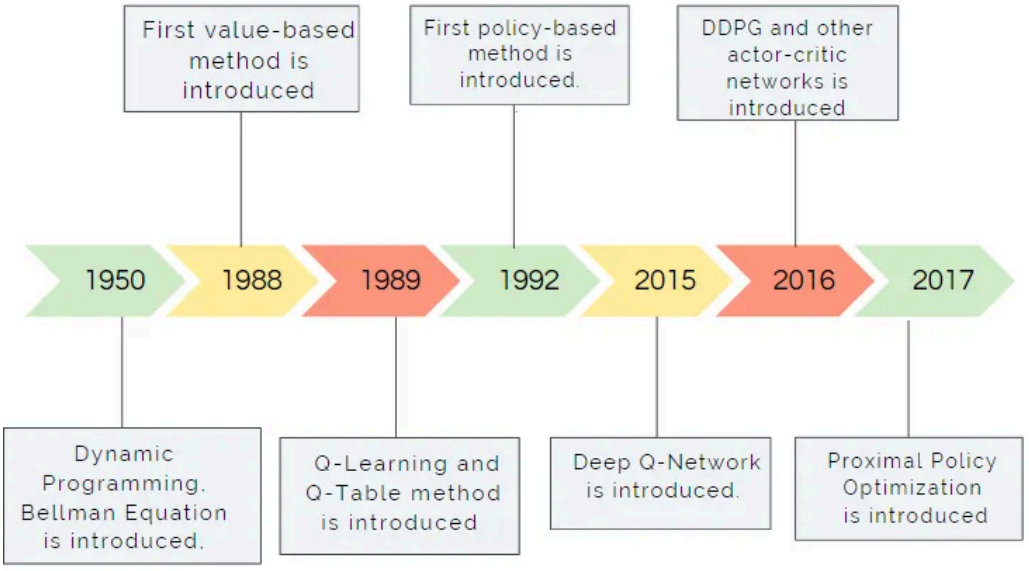


Imagem 8:
https://medium.com/@sanghaviharsh666/navigating-the-evolution-of-reinforcement-learning-a-historical-perspective-6fbbaa010351

Essa heterogeneidade demonstra a complexidade da tarefa de condensar décadas de pesquisa em um único histograma coerente. Cada fonte privilegia perspectivas diferentes e, por consequência, os marcos destacados não são uniformes. Além disso, a diversidade de imagens encontradas inicialmente (e a de se ter mais imagens descritivas da cronologia de Aprendizado por Reforço, não encontradas), assim se exige uma triagem detalhada e um estudo mais aprofundado para evitar simplificações indevidas.

Diante disso, posso concluir que a elaboração de um histograma próprio não poderá ser concluída em apenas uma Semana. O tema envolve compactar anos de evolução teórica, algorítmica e aplicada em uma linha temporal clara, demandando leitura dos artigos, comparação crítica entre fontes e estudo conceitual da área. Assim, nesta segunda Semana, o resultado alcançado foi o mapeamento inicial de diferentes representações já existentes e a constatação da necessidade de um trabalho gradual e sistemático para desenvolver uma versão própria consistente.

Leitura do Artigo: A Concise Introduction to Reinforcement Learning in Robotics” (Nagaraj et al., 2022)

O artigo apresenta uma visão geral de como o aprendizado por reforço vem sendo aplicado na robótica, destacando avanços e desafios de forma superficial. Os autores mostram que a

transição da simulação para o mundo real traz dificuldades como atrasos de comunicação e configuração do tempo de ação, sendo necessário adaptar os experimentos para garantir segurança e repetibilidade. Também discutem o problema das recompensas esparsas em tarefas complexas e apresentam o algoritmo SAC-X como solução para guiar a exploração.

No texto é abordado a integração de ações de empurrar e agarrar usando redes profundas, a aplicação de métodos baseados em gradiente de política (TRPO, PPO, A3C/DPPO) para locomoção de robôs em simulações, e estratégias para navegação eficiente com poucos dados, como a repetição interativa e o Q-learning bootstrapped. No geral, o artigo funciona como uma introdução ao tema (não muito útil a minha pesquisa de compreender a duas áreas juntas), é demonstrado que o RL já oferece resultados promissores em manipulação, locomoção e navegação em robôs, mas ainda exige soluções para recompensas, adaptação ao mundo real e eficiência no uso de dados.

APÊNDICE 2

Termo de Aceite de Entrega 3

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 17 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana 3 do GATE foi realizada a leitura dos resumos levantados na revisão sistemática iniciada no Parsifal, inicialmente voltada para **Aprendizado por Reforço em Robótica**. Com base nos conhecimentos adquiridos, foi definido o objeto de pesquisa da especialização: **Aprendizado por Reforço na transferência Sim-to-Real (simulação para real) em sistemas robóticos**. As configurações do Parsifal foram readaptadas para esta nova pesquisa. Detalhes como PICOC, palavras-chave, critérios de inclusão e exclusão, perguntas de qualidade e respectivos pesos estão disponíveis nos Complementos_GATES.

A questão de pesquisa definida foi:

"Como o aprendizado por reforço tem sido utilizado na transferência sim-to-real em sistemas robóticos, quais desafios têm sido encontrados e quais avanços têm sido reportados?"

Para tanto, foi utilizada a string de busca:
(*"reinforcement learning" OR "deep reinforcement learning" OR "multi-agent reinforcement learning" OR "RL") AND ("sim-to-real" OR sim2real OR "domain adaptation" OR "domain randomization") AND ("robotics" OR "robot" OR "robotic manipulation" OR "mobile robot" OR "autonomous robot" OR "multi-robot systems"*).

Até o momento, a busca resultou em **99 artigos extraídos das bases** ACM Digital Library (8), IEEE Xplore (61) e ScienceDirect (30). Nesta Semana foram lidos 15 artigos (título e abstract), dos quais 5 foram selecionados de acordo com os critérios de inclusão e 10 excluídos por não atenderem ao escopo da pesquisa.

Na leitura inicial (título e abstract), destacou-se o artigo **"Aprendizado por Reforço Guiado: Uma Revisão e Avaliação para Robótica do Mundo Real Eficiente e Eficaz"**, cuja leitura completa foi iniciada devido à sua relevância direta para a temática. Em paralelo, iniciou-se contato com o grupo de pesquisa em RL para robôs do AKCIT, visando participação voluntária e melhor compreensão dos trabalhos em sim-to-real. Por fim, o desenvolvimento do histograma histórico do aprendizado por **reforço**, iniciado na Semana anterior, não foi concluído, pois ainda são necessárias leituras adicionais para consolidar a compreensão conjunta de RL e robótica.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima Semana, as atividades previstas são:

- Concluir a triagem dos artigos por meio da aplicação sistemática dos critérios de inclusão/exclusão e da checklist de qualidade, reduzindo o conjunto a um número significativo que permita leitura aprofundada.
- Finalizar a leitura do artigo de destaque (*Aprendizado por Reforço Guiado*), registrando anotações críticas.
- Iniciar a anotação dos frameworks utilizados nos artigos selecionados, preparando a base para a fase seguinte da pesquisa.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Base da Pesquisa Detalhada(Termo de Entrega 3)

Configurações do Parsifal – Revisão Sistemática

Link parsifal: https://parsif.al/

Para estruturar a revisão sistemática no Parsifal, foram definidos os seguintes parâmetros:

Título da Revisão

Um revisão sobre Aprendizado por Reforço na transferência sim-to-real em sistemas robóticos

PICOC

- **População:** *research in robotic systems*
- **Intervenção:** *use of reinforcement learning for sim-to-real transfer*
- **Comparação:** não aplicável
- **Outcome:** *understanding the methods, challenges and advances of sim-to-real transfer using RL*
- **Contexto:** *academic articles, reviews and case studies in simulated and real environments*

Questão de Pesquisa

"Como o aprendizado por reforço tem sido aplicado na transferência sim-to-real em sistemas robóticos, quais desafios têm sido encontrados e quais avanços têm sido reportados em ambientes simulados e reais?"

Palavras-chave e Sinônimos

Foram selecionados termos alinhados ao PICOC, com destaque para *reinforcement learning (RL)* e sua variação em profundidade (*deep reinforcement learning*), além dos termos relacionados à população (*robot, robotics*). Esses descritores serviram como base para compor a string de busca.

String de Busca

("reinforcement learning" OR "deep reinforcement learning" OR "multi-agent reinforcement learning" OR "RL") AND ("sim-to-real" OR sim2real OR "domain adaptation" OR "domain randomization") AND ("robotics" OR "robot" OR "robotic manipulation" OR "mobile robot" OR "autonomous robot" OR "multi-robot systems")

Com destaque para Science@Direct, que só aceita 8 termos de OR, AND e etc. Então para esta pesquisa foi utilizado o string de busca:

("reinforcement learning" OR "deep reinforcement learning") AND ("sim-to-real" OR "sim2real") AND ("robotic manipulation" OR "mobile robot" OR "robotics")

Critérios de Seleção

- **Inclusão:** artigos que tratam explicitamente de RL aplicado à transferência sim-to-real em robótica, incluindo textos que abordem algoritmos, métodos de adaptação de domínio e estratégias de mitigação.
- **Exclusão:** artigos de ML em robótica sem relação com RL, estudos de RL sem foco em sim-to-real e trabalhos puramente teóricos sem aplicação prática ou discussão no contexto robótico.

Checklist de Qualidade (9 perguntas)

1. O artigo aplica aprendizado por reforço especificamente em robótica?
2. Há foco explícito na transferência sim-to-real ou em técnicas de adaptação de domínio?
3. O artigo descreve de forma clara os algoritmos de RL utilizados?
4. Há avaliações experimentais em ambientes simulados e reais, ou pelo menos validação em robôs físicos?
5. São apresentados desafios enfrentados na transferência sim-to-real (ex.: gap de simulação, ruídos, variabilidade do hardware)?
6. O artigo discute estratégias de mitigação (ex.: randomização de domínio, transferência de políticas, fine-tuning real)?
7. Os resultados obtidos são apresentados de forma quantitativa ou qualitativa, permitindo comparação?
8. Há discussão sobre a relevância prática dos resultados para aplicações reais de robótica?
9. O artigo é claro quanto às limitações e perspectivas futuras no campo de RL sim-to-real em robótica?

Sistema de Pontuação

Cada resposta foi associada a pesos: *Yes (2 pontos)*, *Partially (1 ponto)* e *No (0 pontos)*.

Com 9 perguntas, o escore máximo possível é **18 pontos**. O *cutoff score* foi definido como **12 pontos**, ou seja, artigos abaixo desse valor serão considerados de baixa qualidade para o escopo da revisão.

Bases de Dados

Foram utilizadas as seguintes fontes:

- **ACM Digital Library** – 8 artigos importados

- **IEEE Xplore** – 61 artigos importados
- **ScienceDirect** – 30 artigos importados
- **Scopus** – 0 artigos importados até o momento

APÊNDICE 3

Termo de Aceite de Entrega 4

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 11 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O tema da pesquisa se mantém em **Aprendizado por Reforço na transferência Sim-to-Real em sistemas robóticos**.

Nesta Semana, foi dada continuidade à leitura dos artigos extraídos pela plataforma Parsifal de revisão sistemática da literatura. Embora ainda não tenha sido possível avançar por todos os 99 artigos, eles estão organizados e disponíveis para anotação na planilha complementar GATE4_250925, no link Complementos_GATES.

A análise dos *abstracts* permitiu identificar áreas recorrentes de aplicação do RL no sim-to-real: **manipulação robótica** (montagem de peças, cabos, manipuladores seriais), **locomoção** (quadrúpedes, bípedes e veículos), **navegação autônoma**, **percepção multimodal** (visão e tato) e **aplicações industriais** (montagem e uso de *Digital Twins*). Também foram observadas técnicas recorrentes, como *domain randomization*, *domain adaptation*, *transfer learning*, *imitation + RL* e o uso frequente de algoritmos de *deep RL* (SAC, PPO). Entre os frameworks mais citados, destacam-se **MuJoCo, PyBullet, Gazebo, ROS/ROS2 e OpenAI Gym**, que aparecem como ferramentas comuns de integração entre simulação e realidade. A análise detalhada encontra-se registrada no documento complementar, aba GATE4_250925, disponível em Complementos_GATES.

A leitura direcionada permitiu destacar alguns artigos promissores, especialmente **DROPO (2023)**, que traz uma proposta prática para *domain randomization* em sim-to-real, e **Overcoming the Sim-to-Real Gap (2022)**, que apresenta uma visão conceitual estruturada sobre o *reality gap*.

Além disso, foi concluída a leitura completa do artigo **Guided Reinforcement Learning: A Review and Evaluation for Efficient and Effective Real-World Robotics**, que apresenta uma taxonomia de métodos de *Guided RL* e reforça a importância da eficiência, efetividade e transferência sim-to-real como dimensões centrais para avanços em robótica. A análise detalhada encontra-se no documento complementar, aba GATE4_250925, no link Complementos_GATES.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima Semana, as atividades previstas incluem:

- Concluir a leitura dos artigos extraídos pelo Parsifal, após a especificação de strings de busca mais refinadas, incorporando os termos das áreas já identificadas na pesquisa inicial (manipulação robótica, locomoção, navegação autônoma, percepção multimodal e aplicações industriais), com o objetivo de reduzir o conjunto de artigos a um número mais manejável.
- Realizar a leitura completa e anotar de forma crítica os dois artigos destacados como prioritários: DROPO (2023) e Overcoming the Sim-to-Real Gap (2022).
- Consolidar os registros de leitura na planilha complementar, organizando os artigos já triados e relacionando-os com os critérios de inclusão/exclusão e checklist de qualidade definidos no Parsifal.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Análise da leitura dos abstract dos artigos no Parsifal(Termo de Entrega 4)

Áreas gerais identificadas nos artigos de RL para sim-to-real em robótica

Na leitura dos resumos, aparecem repetidamente algumas linhas de aplicação:

- Manipulação robótica – montagem de peças, instalação de cabos, controle de manipuladores seriais.
- Locomoção – quadrúpedes aprendendo saltos robustos, robôs bípedes com múltiplas habilidades de locomoção, controle lateral em veículos.
- Navegação autônoma – sistemas de navegação em robôs móveis, veículos alados (flapping wing) e plataformas industriais.
- Percepção multimodal – aprendizado visual-tátil em tarefas de manipulação, uso de imagens de profundidade e sensores combinados para reduzir o gap sim-to-real.
- Indústria e manufatura – montagem robótica em linhas de produção, uso de Digital Twins e simulações industriais para transferência.

Técnicas que se repetem nos artigos

- Domain Randomization – variação de dinâmicas e parâmetros de simulação para robustez no mundo real (ex.: DROPO).
- Domain Adaptation – ajuste de representações (visão, imagens sintéticas vs reais) para aproximar ambientes.
- Transfer Learning – aprendizado prévio em simulação usado como base para ajustes no real, em manufatura e robôs móveis.
- Imitation + RL – restrições por demonstração para guiar robôs bípedes e manipuladores.
- Deep RL (SAC, PPO, variantes) – recorrente em navegação, manipulação e controle de quadrúpedes.

Frameworks e bibliotecas mais comuns

Mesmo que os abstracts não detalham todos, os nomes que se repetem ou aparecem associados na literatura incluem:

- PyBullet e MuJoCo – simuladores usados para locomoção e manipulação.

- Gazebo – em tarefas de navegação e controle de veículos.
- ROS/ROS2 – como middleware para integração simulação ↔ robô real.
- OpenAI Gym / RLlib – para estruturar algoritmos de RL em simulação antes da transferência.

Artigos destaques

A leitura dos artigos permitiu identificar alguns trabalhos alinhados à perspectiva da pesquisa em Aprendizado por Reforço na transferência sim-to-real em sistemas robóticos. Entre eles, destacam-se os seguintes como mais promissores:

- DROPO: Sim-to-real transfer with offline domain randomization and policy optimization (Robotics and Autonomous Systems, 2023)

  Apresenta uma proposta diretamente voltada ao domain randomization para transferência sim-to-real, introduzindo otimização de políticas offline para ampliar a robustez em cenários reais.
- Bridging the simulation-to-real gap of depth images for robotic navigation via generative adversarial networks (Expert Systems with Applications, 2024)

  Explora técnicas de domain adaptation com GANs, aplicadas à navegação robótica, reduzindo discrepâncias entre imagens sintéticas e reais.
- Robust quadruped jumping via deep reinforcement learning (Robotics and Autonomous Systems, 2024)

  Enfoca a locomoção de quadrúpedes em tarefas de salto, utilizando deep RL para garantir estabilidade e melhor generalização em ambientes reais.
- Visual–tactile learning of robotic cable-in-duct installation (Automation in Construction, 2025)

  Integra aprendizado visual e tátil para manipulação robótica em tarefas industriais complexas, ressaltando a relevância da percepção multimodal no sim-to-real.
- A phased robotic assembly policy based on a PLDRL approach (Journal of Manufacturing Systems, 2025)

  Propõe uma política de montagem robótica em fases, aliando deep RL e aprendizado progressivo para aplicações industriais.

- Imitation-constrained evolutionary learning for bipedal wheeled robots (Engineering Applications of Artificial Intelligence, 2025)

Combina aprendizado por reforço, imitação e algoritmos evolucionários em robôs bípedes sobre rodas, mostrando a força de abordagens híbridas.

- Dexterous in-hand manipulation of slender cylindrical objects (Robotics and Autonomous Systems, 2025)

  Analisa desafios da manipulação de precisão em objetos cilíndricos, discutindo coordenação fina e transferência de habilidades aprendidas.

- Overcoming the Sim-to-Real Gap in Autonomous Robotic Systems (Procedia CIRP, 2022)

  Oferece um panorama conceitual sobre o reality gap, sistematizando estratégias gerais para superá-lo em sistemas autônomos.

Entre os artigos listados, os dois que se mostram mais promissores para leitura aprofundada nas próximas Semanas são:

1. DROPO (2023), pela proposta atual e prática de otimização de políticas com domain randomization aplicado diretamente ao sim-to-real.
2. Overcoming the Sim-to-Real Gap (2022), por apresentar uma visão clara e estruturada dos principais desafios e abordagens para reduzir o reality gap

Leitura do Artigo Guided Reinforcement Learning: A Review and Evaluation for Efficient and Effective Real-World Robotics [Survey]

Artigo: https://ieeexplore.ieee.org/document/9926159

O artigo Guided Reinforcement Learning: A Review and Evaluation for Efficient and Effective Real-World Robotics é uma revisão ampla sobre o chamado Guided RL, que seria basicamente usar conhecimento extra (científico, do mundo ou de especialistas) para orientar o aprendizado por reforço. Isso aparece como resposta a uma limitação clara: o RL puro, por tentativa e erro, não escala bem para robótica no mundo real, porque é caro, arriscado e muitas vezes não transfere da simulação para o físico.

O texto deixa claro que o Guided RL busca três coisas principais: eficiência (menos interações, menos tempo de treino), efetividade (resultados melhores, políticas que realmente funcionam) e transferência sim-to-real (levar o que foi aprendido no simulador para robôs reais). O legal é que eles tratam isso como dimensões que se complementam e, no fim, o ideal é atingir as três.

Eles montam uma taxonomia de métodos. Alguns pontos que chamam atenção:

- Representação de estados: usar mais sensores ou combinações, por exemplo visão + propriocepção em robôs quadrúpedes como o ANYmal.
- Design de recompensas: modelar recompensas mais densas ou bioinspiradas, como no caso da locomoção de bípedes tipo o Cassie.
- Curriculum learning e RL hierárquico: dividir tarefas grandes em etapas menores, bem útil em robôs humanoides que precisam aprender a andar e depois correr.
- Aprendizado por demonstração: partir de exemplos humanos para manipulação fina de objetos, reduzindo o tempo de exploração aleatória.
- Métodos de sim-to-real: aqui aparece domain randomization (ex.: OpenAI usando randomização para a mão robótica resolver o Cubo Mágico), domain adaptation (como redes que ajustam imagens sintéticas para parecerem reais, tipo o GraspGAN), e também a busca por simuladores mais realistas.

A parte da avaliação mostra que, para eficiência, se destacam aprendizado paralelo, abstração de ações e demonstrações; para efetividade, aparecem curriculum learning, RL hierárquico e offline RL; e, para sim-to-real, os melhores são mesmo domain randomization, domain adaptation e simuladores realistas.

O artigo também aponta alguns desafios que ainda não estão resolvidos: melhorar a eficiência amostral (precisar de menos dados), lidar com tarefas de longo prazo (ex.: navegação em ambientes dinâmicos ou robôs colaborativos) e, principalmente, reduzir o reality gap. Nas perspectivas futuras, falam de integrar mais modalidades (tato, visão, som), criar recompensas inspiradas em biologia ou até aprendidas de forma inversa, melhorar a fidelidade dos simuladores e explorar randomização mais inteligente, talvez até usando modelos generativos.

Termo de Aceite de Entrega 5

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 2 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O tema da pesquisa se mantém em **Aprendizado por Reforço na transferência Sim-to-Real em sistemas robóticos**.

Nesta Semana, foi realizada a análise consolidada de **38 artigos selecionados** no Parsifal, cujos registros foram organizados na planilha complementar, disponível no link Complementos_GATES. A partir da leitura dos abstracts e da extração de métricas textuais, foram identificadas as principais **tendências da literatura**: crescimento expressivo das publicações entre 2020 e 2025, com destaque para o periódico *Robotics and Autonomous Systems*, que concentrou 10 artigos.

As áreas de aplicação mais recorrentes foram **manipulação robótica, navegação autônoma e locomoção de quadrúpedes**, complementadas por estudos em grasping, montagem e uso de Digital Twins em cenários industriais. Entre as técnicas de sim-to-real, destacaram-se *domain randomization*, *domain adaptation*, *system identification* e *transfer learning*, sendo **PPO e SAC** os algoritmos de RL mais aplicados. A análise de palavras-chave e frequência textual reforçou a centralidade de termos como *reinforcement learning*, *sim-to-real* e *robotics*. As análises detalhadas estão registradas no documento Complementos_GATES

Além disso, foi concluída a leitura aprofundada de dois artigos selecionados como prioritários: **DROPO (2023)**, que propõe *domain randomization offline* aliado a otimização de políticas como solução prática para aumentar a robustez no real, e **Overcoming the Sim-to-Real Gap (2022)**, que defende o uso de simulações calibradas como gêmeos digitais para viabilizar transferências diretas (*zero-shot transfer*). A comparação mostrou que ambas as abordagens oferecem caminhos complementares para enfrentar o *reality gap*. As análises detalhadas da leitura e principais aprendizados estão no documentos Complementos_GATES

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Leitura da introdução e conclusão dos 38 artigos selecionados, de forma a refinar a compreensão e consolidar as áreas de aplicação do Aprendizado por Reforço no contexto sim-to-real em robótica,

ampliando o mapeamento iniciado na Semana anterior.
- Apresentar e organizar as áreas de RL em sim-to-real identificadas, descrevendo de modo comparativo como cada abordagem se distribui entre manipulação robótica, locomoção, navegação, percepção multimodal e aplicações industriais.
- Configurar o ambiente Gym e revisar os fundamentos do ROS e ROS2,

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Leitura dos artigos e análise do Parsifal(Termo de Entrega 5)

Leitura do Artigo: *DROPO: Sim-to-real transfer with offline domain randomization and policy optimization (2023)*

O artigo apresenta uma proposta que me chamou bastante atenção: o DROPO, que combina *domain randomization* de forma offline com técnicas de otimização de política. A ideia central é contornar um dos maiores problemas em aprendizado por reforço aplicado à robótica, o sim-to-real gap, ou seja, as diferenças entre o ambiente simulado e o ambiente real que frequentemente fazem políticas falharem na hora de transferir.

O trabalho defende que nem sempre é necessário ter um simulador perfeito ou detalhado em nível físico, o que muitas vezes é caro e difícil de construir. Em vez disso, eles optam por variar sistematicamente parâmetros dinâmicos como massa, atrito, ruído de sensores e até condições de iluminação. Essa randomização gera uma diversidade de cenários de treino, e a política resultante se torna mais robusta porque já foi exposta a uma ampla gama de situações.

Depois da randomização, o artigo aplica otimização de política para consolidar um comportamento mais estável e transferível. Essa combinação mostrou resultados promissores tanto em robôs manipuladores (por exemplo, braços robóticos em tarefas de pegar e mover objetos) quanto em robôs móveis (locomoção em ambientes incertos). A conclusão principal é que políticas treinadas com randomização offline conseguem se adaptar melhor ao mundo real, reduzindo falhas inesperadas.

O que aprendi com essa leitura é que o DROPO propõe uma solução prática: em vez de gastar esforço criando simuladores hiper-realistas, é mais viável investir em simulações variadas que forcem a política a generalizar. Isso abre um caminho interessante para diferentes áreas, não só manipulação e locomoção, mas também outras aplicações que enfrentam o mesmo problema do *reality gap*.

Leitura do Artigo: *Overcoming the Sim-to-Real Gap in Autonomous Robots (2022)*

Esse artigo aborda o problema do *reality gap* por um caminho quase oposto ao do DROPO. Em vez de criar diversidade de cenários artificiais, os autores defendem que é possível fazer transferência direta (zero-shot transfer) se a simulação for suficientemente realista.

Eles usam como estudo de caso um robô móvel simples (mBot modificado), equipado com câmera e motores básicos, para tarefas de navegação e localização de objetos. O ponto forte do trabalho foi a construção de um simulador no Unity com PhysX, cuidadosamente

calibrado para refletir as características físicas do robô real: peso, atrito, limitações dos motores, além de aspectos visuais como iluminação e campo de visão da câmera.

O aprendizado foi feito com PPO aliado a curriculum learning, ou seja, as tarefas foram apresentadas de forma progressiva para facilitar a convergência. Os resultados mostraram que, mesmo com um robô simples, o comportamento aprendido em simulação conseguiu ser transferido diretamente para o real sem necessidade de randomização.

As áreas de aplicação destacadas foram principalmente a navegação autônoma em ambientes internos e a inspiração em Digital Twins, reforçando o valor de simulações fiéis em contextos industriais. Um aprendizado importante aqui é que, embora essa abordagem funcione em cenários relativamente simples, os autores reconhecem que o método é frágil quando ocorrem variações inesperadas no ambiente real.

Esses dois artigos abordam o mesmo problema — o sim-to-real gap — por caminhos complementares. O DROPO aposta na variedade artificial (randomização offline) para treinar políticas robustas, enquanto o Overcoming investe na fidelidade da simulação (calibrar o ambiente digital como um gêmeo realista).

O aprendizado que levo da leitura é que as duas soluções são válidas, mas cada uma com limitações: simuladores perfeitos são caros e limitados, enquanto randomização pode gerar políticas conservadoras demais. No fundo, parece que a tendência da área é combinar as duas estratégias, usando simulações calibradas como base, mas também aplicando randomização para cobrir o inesperado

Tabela Comparativa

Aspecto	**DROPO (2023)**	**Overcoming the Sim-to-Real Gap (2022)**
Abordagem principal	*Offline Domain Randomization* + *Policy Optimization*	Simulação realista calibrada (gêmeo digital simplificado)
Estratégia contra o reality gap	Gera diversidade sintética variando massa, atrito, ruído e parâmetros de simulação	Reproduz fielmente o hardware e ambiente no simulador Unity (PhysX)
Modelos de RL	Deep RL (PPO) com otimização de política	Deep RL (PPO) com *curriculum learning*
Foco das aplicações	Manipulação robótica e locomoção (robôs móveis e manipuladores)	Robôs móveis autônomos (navegação e localização de objetos)

Vantagem principal	Políticas mais robustas e generalizáveis sem precisar de simulador perfeito	Transferência direta (*zero-shot transfer*) possível com simulador fiel
Limitação	Pode gerar políticas conservadoras ou com overfitting na randomização	Alto custo de calibração e sensibilidade a pequenas variações do real
Áreas destacadas	Manipulação, locomoção, robôs móveis	Navegação autônoma, logística, uso de Digital Twins
Aprendizado geral	Randomização offline ajuda a robustez no real	Simulações fiéis podem ser suficientes em tarefas simples

Análises dos 38 artigos selecionados no Parsifal

Aqui está um texto consolidado para os **Complementos** com base na análise dos 38 artigos selecionados e nos gráficos gerados. Estruturei em seções para facilitar a leitura e documentação:

Análise dos 38 Artigos Selecionados (2020–2025)

Total de registros analisados: 38
Abstracts não vazios: 38
Tamanho total dos textos: 55.329 caracteres
Total de palavras: 6.629 (5.210 após remoção de stopwords)
Palavras únicas: 1.750
Palavra mais frequente: *real* (108 vezes)

1. Tendência Temporal

- As publicações cresceram de **1 artigo em 2020** para **13 artigos em 2025**, indicando um aumento consistente do interesse em *sim-to-real* aplicado ao Aprendizado por Reforço (RL).
- O ano **mais produtivo foi 2025**, confirmando que o tema está em forte ascensão.

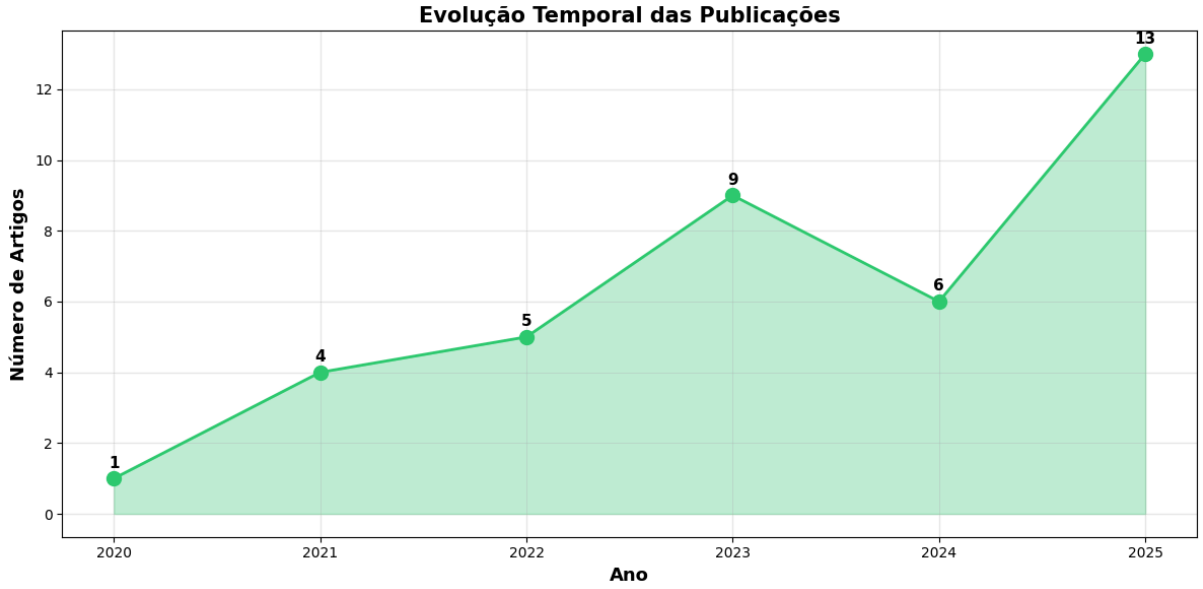


Imagem 9

2. Principais Journals e Conferências

- **Robotics and Autonomous Systems** se destacou com **10 artigos**, consolidando-se como o principal veículo de publicação.
- Outros periódicos relevantes: *Engineering Applications of Artificial Intelligence* (4), *Procedia CIRP* (3) e *Robotics and Computer-Integrated Manufacturing* (2).
- A diversidade de venues (13 diferentes) mostra que o tema é interdisciplinar, abarcando tanto robótica quanto IA aplicada em manufatura e sistemas autônomos.

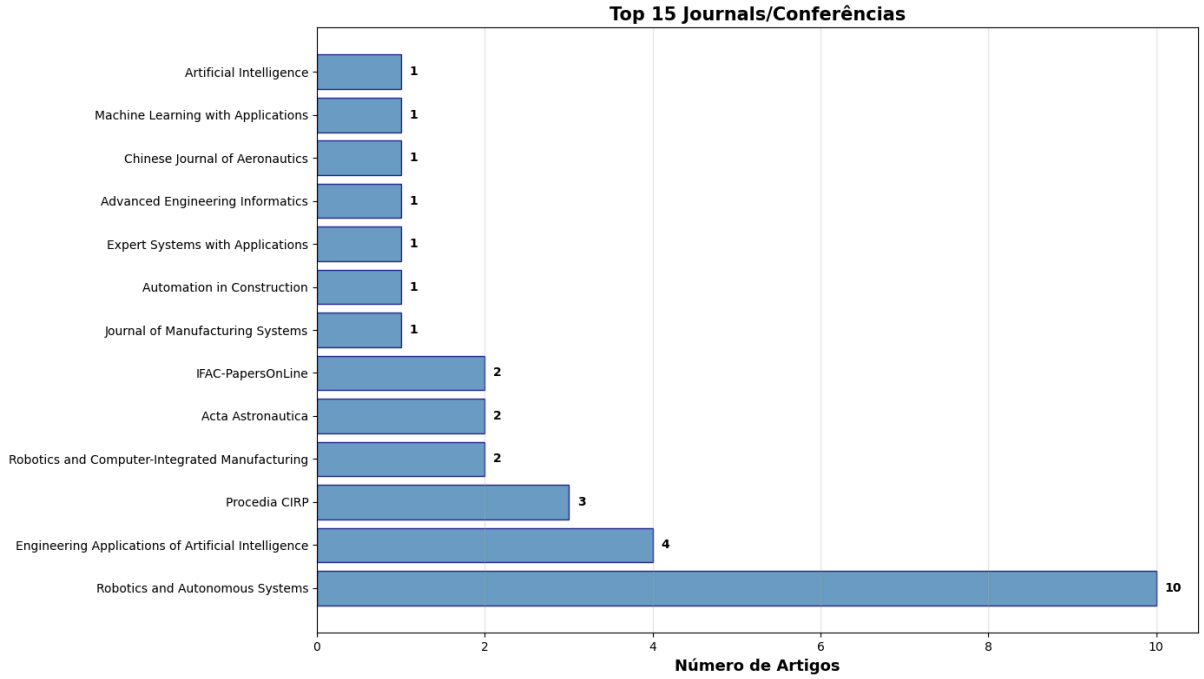


Imagem 10

3. Aplicações Robóticas

- **Manipulação robótica** foi a área mais recorrente (7 artigos), abordando desde montagem até manipulação de precisão.
- Outras aplicações frequentes:

    - **Navegação autônoma** (4 artigos).
      Grasping, quadrúpedes, robôs móveis e assembly (3 artigos cada).
      Menções mais específicas incluíram **robôs humanoides**, **braços robóticos** e até **autonomous driving**.

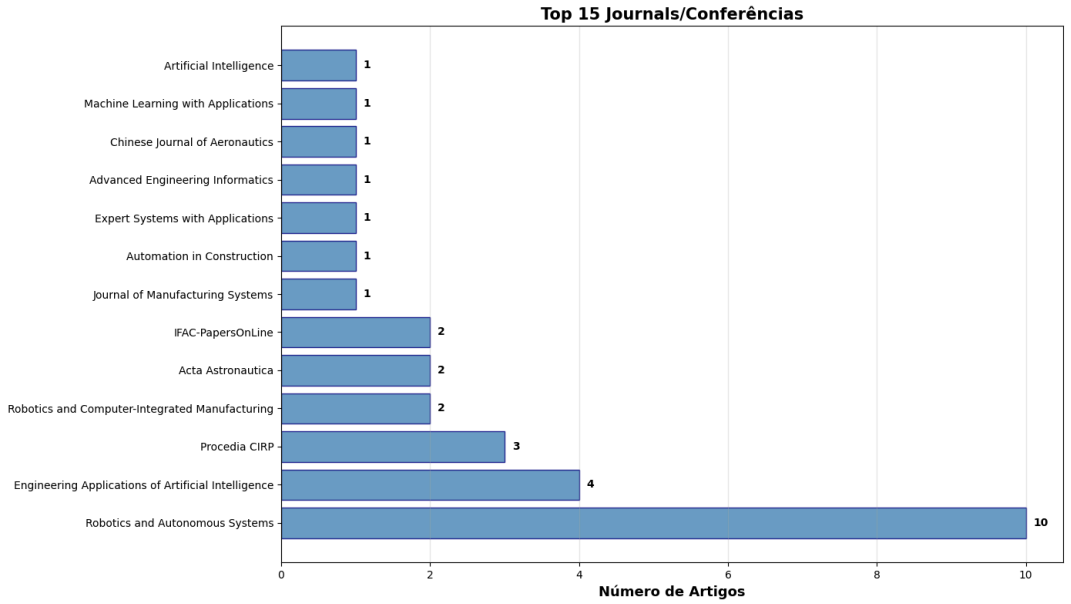


Imagem 11

4. Técnicas de RL e Sim-to-Real

- **Algoritmos mais usados:**
  - *PPO* e *SAC* (3 artigos cada), seguidos por TD3, DDPG e DQN.

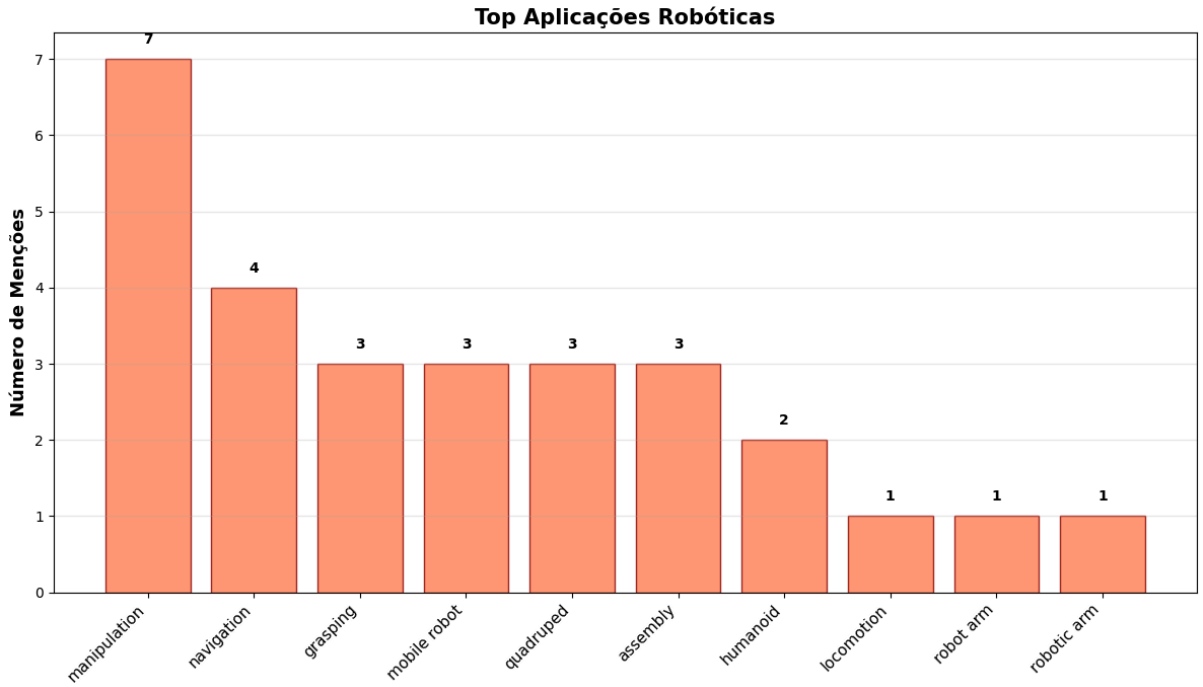


Imagem 12

- **Técnicas de transferência:**
    - *Sim-to-real* foi mencionado em **23 artigos (60,5%)**, destacando sua centralidade.
    - *Domain randomization* apareceu em **9 artigos**, confirmando sua relevância prática.
      Domain adaptation (4), *system identification* (3) e *transfer learning* (2) foram usados como estratégias complementares.

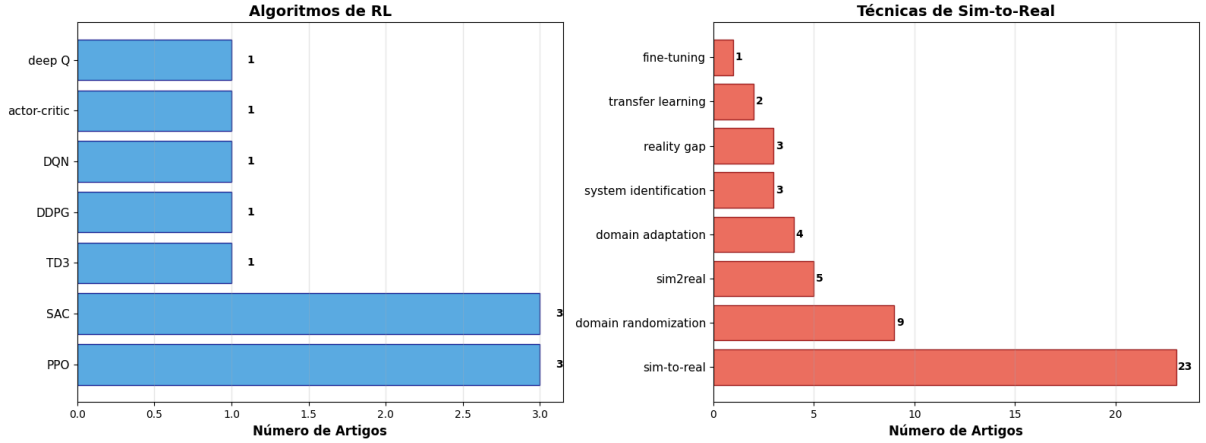


Imagem 13

5. Keywords Mais Frequentes

- **Reinforcement learning** (15 vezes) e **deep reinforcement learning** (9 vezes) foram os termos mais comuns.
- **Sim-to-real** e variações como *sim-to-real transfer* e *sim2real transfer* também aparecem com destaque.
  Outras palavras relevantes: *machine learning, robotics, domain randomization, tactile sensing, digital twin* — mostrando que há forte interface entre RL e percepção multimodal.

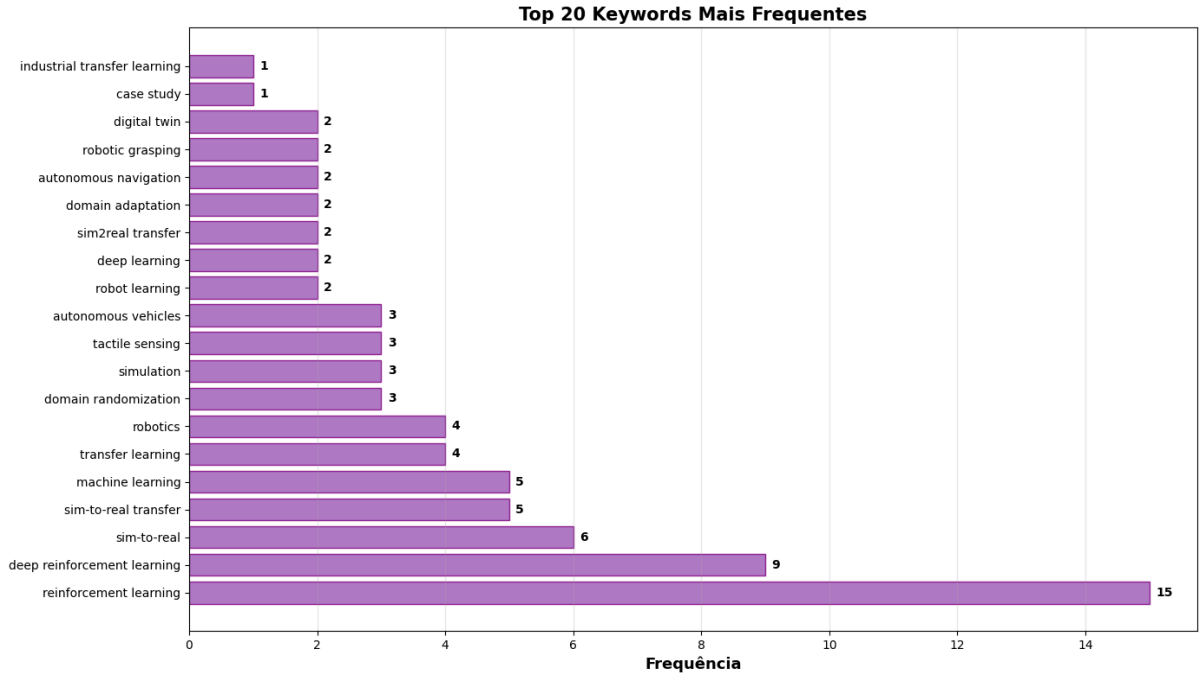


Imagem 14

6. Gráfico com as 25 palavras mais frequentes

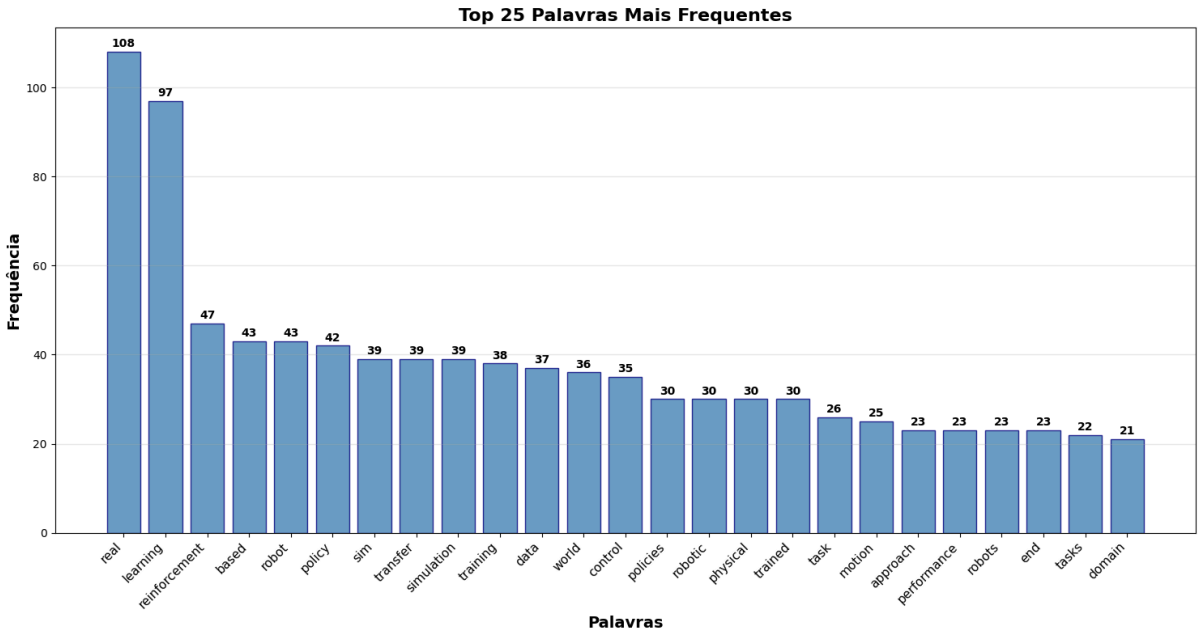


Imagem 15

O gráfico de barras evidencia que os termos **“real” (108 ocorrências)**, **“learning” (97)** e **“reinforcement” (47)** são os mais frequentes, destacando a ênfase da área em levar o

aprendizado por reforço para o mundo real. Palavras como *robot*, *simulation*, *transfer*, *policy* e *training* também se repetem com intensidade, confirmando a relevância de discutir como políticas de RL são treinadas em ambientes simulados e transferidas para robôs físicos.

7. Nuvem de Palavras

Imagem 16

A nuvem de palavras reforça essa tendência ao destacar conceitos centrais como **task, robot, reinforcement learning, policy, environment e simulation**. Esses termos, quando associados, revelam os eixos dominantes da pesquisa:

- **Tarefas robóticas específicas** (manipulação, locomoção, navegação);
- **Ambientes de simulação** como ponto de partida;
  Transferência de políticas para reduzir o *reality gap*.

8. Resumo

- **Total de artigos analisados:** 38 (2020–2025).
- **Áreas mais exploradas:** manipulação, navegação e robôs quadrúpedes.
- **Técnica predominante:** *sim-to-real transfer* (60,5% dos artigos).
  Algoritmos mais aplicados: PPO e SAC.

- **Principal desafio identificado:** redução do *reality gap*, com ênfase em *domain randomization* e simulações calibradas (Digital Twins).

APÊNDICE 4

Termo de Aceite de Entrega 6

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 18 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O tema da pesquisa se mantém em **Aprendizado por Reforço na Transferência Sim-to-Real em Sistemas Robóticos**.

Nesta Semana, foi concluída a leitura das introduções e conclusões dos 38 artigos selecionados, permitindo consolidar uma estrutura conceitual que relaciona os principais tipos de robôs, simuladores, técnicas de transferência e frameworks de implementação empregados na literatura recente (2020–2025). Essa sistematização possibilitou a criação de um mapeamento das quatro grandes áreas de conhecimento: **robôs, simuladores, técnicas sim-to-real, frameworks.** Os registros detalhados da categorização das áreas, encontram-se documentados no arquivo Complementos_GATES.

Os resultados mostram que robôs manipuladores, quadrúpedes, drones e cobots apresentam diferentes graus de *reality gap* físico e sensorial, o que exige soluções específicas. Simuladores como MuJoCo e Isaac Gym se destacam pela precisão física e escalabilidade para locomoção e manipulação, enquanto Gazebo/Ignition é amplamente usado na integração com ROS2 para testes de hardware real. Em relação às técnicas de sim-to-real, Domain Randomization, Domain Adaptation, System Identification e Residual Reinforcement Learning continuam sendo as mais recorrentes. Já os frameworks mais integrados ao ecossistema experimental são ROS2, Stable-Baselines3 e Isaac Lab, pela capacidade de unir simulação, aprendizado e controle físico.

Com base nessa análise, foram selecionados 8 artigos prioritários para leitura aprofundada nas próximas quatro Semanas, abrangendo temas complementares como Digital Twins, Safe Reinforcement Learning, manipulação multimodal, adaptação visual e arquiteturas bidirecionais sim-to-real/real-to-sim. Esses trabalhos serão fundamentais para direcionar a fase experimental da pesquisa e fundamentar as decisões de implementação.

Nesta Semana, também foi definido o **estudo aplicado** da pesquisa: uma investigação prática de sim-to-real utilizando o **robô quadrúpede Go2 EDU da Unitree**, explorando o uso de algoritmos de RL para locomoção e navegação. Por motivos técnicos relacionados ao hardware, a preparação do ambiente foi iniciada e ainda não puderam ser executados. A lista dos artigos recomendados, encontra-se documentada no arquivo **Complementos_GATES**

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Testar um simulador com o modelo do robô Go2 EDU, validando o ambiente de locomoção e as variáveis físicas básicas (massa, atrito, torque e delays).
- Revisar as técnicas de Domain Randomization e Automatic Domain Randomization (ADR) aplicadas especificamente a robôs quadrúpedes.
- Realizar leitura aprofundada dos artigos selecionados “Sim-to-Lab-to-Real: Safe Reinforcement Learning for Robotic Control”, registrando anotações críticas sobre segurança, percepção e robustez.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Mapeamento das Áreas - Aprendizado por Reforço na Transferência Sim-to-Real em Sistemas Robóticos (Termo de Entrega 6)

Resumo dos 38 artigos:

O conjunto cobre 2020–2025 com crescimento constante, concentrando publicações em Robotics and Autonomous Systems e periódicos de IA aplicada. As áreas de aplicação mais frequentes são manipulação (incluindo destreza "in-hand" e montagem), locomoção/navegação (robôs móveis, quadrúpedes, plataformas aéreas) e cenários industriais (células de montagem, *Digital Twins*). Em técnicas sim-to-real, destacam-se domain randomization e domain adaptation, com presença de system identification em contextos industriais. Em algoritmos, predominam PPO e SAC, usualmente combinados com curriculum/hierarchical RL, imitation+RL e, em alguns casos, safe RL para mitigar riscos no mundo real. Os estudos alternam entre duas filosofias: (i) simulação fiel e calibrada (*twins*, identificação de parâmetros) e (ii) diversidade sintética (randomização/adaptação) para robustez.

8 artigos recomendados para leitura (próximas Semanas)

Seleção pensada para cobrir manipulação, navegação, industrial/Digital Twin, segurança e arquiteturas gerais de sim-to-real.

1. A digital twin-based sim-to-real transfer for industrial tasks — *Journal/venue registrado; 2024/2025*
2. Sim-to-real transfer of active suspension control using reinforcement learning — *Robotics and Autonomous Systems, 2024*
3. Visual–tactile learning of robotic cable-in-duct installation — *Automation in Construction, 2025*
4. Dexterous in-hand manipulation of slender cylindrical objects — *Robotics and Autonomous Systems, 2025*
5. Bridging the simulation-to-real gap of depth images for robotic navigation via GANs — *Expert Systems with Applications, 2024*
6. A phased robotic assembly policy based on a PLDRL approach — *Journal of Manufacturing Systems, 2025*
7. Sim-to-Lab-to-Real: Safe reinforcement learning for robotic control — *venue registrado; 2024/2025*
8. An architecture for sim-to-real and real-to-sim in robotics — *venue registrado; 2025*

Observação: já lidos e não incluídos na lista acima: DROPO (2023) (randomização offline + otimização de política) e Overcoming the Sim-to-Real Gap (2022) (gêmeo/simulação calibrada e *zero-shot transfer*).

1. Categorias de Robôs e Relevância

Nesta etapa, as categorias de robôs foram organizadas conforme suas propriedades físicas, dinâmicas e o tipo de *reality gap* predominante. Essa categorização é fundamental, pois cada classe de robô apresenta um conjunto distinto de desafios na transição entre simulação e mundo real.

Os **manipuladores industriais** e **cobots** representam o grupo mais consolidado na literatura. Esses sistemas envolvem alta precisão, controle de contato e modelagem detalhada de atrito — fatores que amplificam o *gap* entre simulação e realidade. O RL aplicado a esses casos se beneficia de técnicas como *System Identification* e *Residual Policy Learning*, que permitem calibrar parâmetros físicos e ajustar controladores clássicos.

Já os **robôs móveis terrestres (UGVs)**, **quadrúpedes** e **bípedes** enfrentam desafios dinâmicos mais complexos, como variações de terreno, instabilidade e longos horizontes temporais de decisão. Por isso, são amplamente estudados com *Domain Randomization*, *Curriculum Learning* e *Meta-Learning*, que ampliam a generalização e adaptabilidade.

Os **robôs aéreos (UAVs)** e **manipuladores móveis** representam a fronteira atual da pesquisa sim-to-real. No primeiro caso, as dinâmicas são rápidas e sujeitas a ruídos aerodinâmicos, exigindo *Robust RL* e *System ID online*. No segundo, a complexidade vem da coordenação entre mobilidade e manipulação, onde frameworks como ROS2 e MoveIt têm papel central.

Essa diversidade justifica o uso de diferentes simuladores e técnicas de transferência, uma vez que cada perfil de robô demanda um equilíbrio próprio entre fidelidade física, velocidade de simulação e robustez das políticas aprendidas.

2. Simuladores e Adequação Técnica

Os simuladores analisados variam quanto à precisão física, realismo visual e integração com frameworks de RL. O **MuJoCo** é amplamente reconhecido pela precisão física e estabilidade numérica, tornando-se o padrão para manipulação e locomoção em pesquisa. Já o **PyBullet**, embora menos preciso, é aberto, rápido e ideal para *Domain Randomization* e experimentos interativos. O **Gazebo (Ignition)** destaca-se pela integração direta com ROS/ROS2, o que o torna essencial em pipelines que envolvem hardware real. Em contrapartida, o **NVIDIA Isaac Sim** oferece fotorrealismo e suporte a *domain randomization* visual, sendo ideal para estudos baseados em percepção.

Para escalas massivas de treino, o **Isaac Gym** e o **Isaac Lab** são incomparáveis, permitindo executar milhares de simulações paralelas em GPU — especialmente úteis em DR/ADR. **Unity ML-Agents** e **Habitat-Sim** são mais adequados para tarefas de percepção visual, com uso intenso de *domain adaptation*.

Simuladores como **CoppeliaSim** e **Webots** continuam relevantes para manipulação modular e ensino, enquanto **CARLA** e **AirSim** cobrem nichos especializados de veículos autônomos e drones.
Cada simulador, portanto, é mais eficaz quando alinhado ao perfil de robô e à natureza do *gap* predominante — físico, sensorial ou visual.

3. Técnicas de Sim-to-Real – Justificativas e Comparação

O levantamento revelou que a pesquisa em transferência sim-to-real converge em torno de **quatro eixos técnicos principais**:

1. **Randomização** – aumentar diversidade em simulação;
2. **Adaptação** – alinhar domínios visual ou dinâmico;
3. **Calibração/Modelagem** – reduzir discrepâncias físicas;
4. **Ajuste/Refinamento** – melhorar política pós-treino.

A **Domain Randomization (DR)** é a técnica mais difundida, principalmente em manipulação e locomoção. Seu ponto forte está em forçar o agente a aprender políticas invariantes, capazes de lidar com incertezas. O avanço natural dessa ideia, o **Automatic Domain Randomization (ADR)**, introduz um currículo de perturbações crescentes, permitindo à política “aprender a aprender” com falhas.

A **Domain Adaptation (DA)** tem papel complementar, especialmente em sistemas baseados em visão. Ao usar GANs (CycleGAN, GraspGAN) ou *feature alignment*, o modelo aprende a interpretar imagens reais mesmo treinando com dados sintéticos. Técnicas de **System Identification** e **Dynamics Randomization** visam equalizar as dinâmicas físicas, calibrando ou perturbando parâmetros do simulador. Enquanto SysID busca a fidelidade exata, DR prioriza robustez. Outros métodos, como **Residual Policy Learning**, **Imitation Learning** e **Fine-Tuning no Real**, atuam em estágios posteriores — reduzindo o *reality gap* residual através de ajustes controlados em robôs reais. Já **Meta-Learning** e **Robust RL** emergem como estratégias avançadas, capazes de adaptar políticas dinamicamente a novos domínios.

Essa diversidade demonstra que a superação do *reality gap* não depende de uma técnica única, mas de composições inteligentes de estratégias conforme o tipo de robô, simulador e objetivo.

4. Frameworks e Integrações Práticas

O ecossistema de frameworks é o elo entre teoria e implementação.

O **ROS/ROS2** continua sendo o principal middleware, garantindo integração entre simulação e hardware real. Sua compatibilidade com **Gazebo**, **Isaac Sim** e **Unity** o torna o núcleo de qualquer pipeline de RL aplicado à robótica física. Ferramentas como **Gymnasium**, **Stable-Baselines3** e **CleanRL** oferecem ambientes e algoritmos padronizados, simplificando o teste de políticas. Já **RLlib** e **TorchRL** expandem para contextos distribuídos e com maior customização.

Frameworks especializados, como **robosuite**, **ManiSkill2** e **dm_control**, fornecem ambientes de manipulação e locomoção padronizados em MuJoCo. Em contrapartida, **Isaac Lab** foi projetado para treinar políticas em larga escala com *domain randomization* e *privileged learning*, usando GPUs NVIDIA. O uso combinado de **ROS2 + Isaac Lab + Stable-Baselines3** constitui, portanto, uma base robusta para o desenvolvimento de pipelines sim-to-real modernos, integrando simulação fotorrealista, controle físico preciso e algoritmos de RL escaláveis.

5. Síntese Geral e Justificativa Técnica

A análise consolidada dos 38 artigos e das ferramentas associadas evidencia uma estrutura de quatro dimensões interdependentes:

1. **Domínio físico (robô e simulador)** – Define o tipo de realidade modelada e o grau de fidelidade necessário.
2. **Técnica de transferência** – Define o mecanismo de mitigação do *gap*.
3. **Framework de execução** – Define a viabilidade prática e reprodutibilidade.
4. **Escopo de aplicação** – Define o desafio final (manipulação, locomoção, percepção).

As correlações mais observadas foram:

- Manipulação robótica → *Domain Randomization* + *SysID* + *Residual RL* (MuJoCo, robosuite)
- Locomoção quadrúpede → *ADR* + *Robust RL* + *Curriculum Learning* (Isaac Gym/Lab)
- Navegação autônoma → *Domain Adaptation* + *Representation Learning* (Unity, Habitat)
- Cobots e manipuladores móveis → *Fine-tuning* + *ROS2* + *Safety Shields* (Gazebo, MoveIt)

O panorama confirma que o sucesso da transferência sim-to-real depende da **sinergia entre modelagem física, adaptação visual e ajustes pós-treino**, com pipelines cada vez mais híbridos combinando *domain randomization*, *meta-learning* e *fine-tuning seguro* no robô real.

Termo de Aceite de Entrega 7

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 16 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O tema da pesquisa se mantém em **Aprendizado por Reforço na Transferência Sim-to-Real em Sistemas Robóticos**, agora avançando para a **fase de estudo de caso aplicada ao robô quadrúpede Go2 EDU da Unitree**. Nesta etapa, o foco é integrar o conhecimento teórico consolidado nas semanas anteriores às práticas de configuração de ambiente, simulação e posterior transferência para o mundo real.

Durante esta semana, foi iniciada a **instalação e configuração do ambiente de simulação** necessário para os experimentos, incluindo o Isaac Gym, Gymnasium e as dependências do ROS2 e Stable-Baselines3. Entretanto, o processo apresentou múltiplas dificuldades de compatibilidade de hardware e drivers GPU, em especial na instalação dos módulos de física e aceleração gráfica. Apesar dos desafios, a configuração básica foi concluída, com o ambiente funcional, mas ainda sem condições de execução e teste do modelo do robô.

Foi realizada uma **leitura aprofundada do artigo "Sim-to-Lab-to-Real: Safe Reinforcement Learning for Robotic Control"**, que apresentou um pipeline robusto para transferência segura de políticas (Sim → Lab → Real) com uso de shielding, Reachability RL e PAC-Bayes Control. A abordagem mostrou forte aplicabilidade ao caso do Go2 EDU, por empregar um robô quadrúpede real em cenários de navegação indoor com percepção visual pura, oferecendo uma base conceitual sólida para a fase experimental, as análises estão disponíveis no link Complementos_GATES

Além disso, foi conduzida uma **revisão das técnicas de Domain Randomization (DR) e Automatic Domain Randomization (ADR)** aplicadas a robôs quadrúpedes. A análise destacou que a randomização física, sensorial e visual continua sendo o método mais eficaz para reduzir o *reality gap*, e que o ADR representa uma evolução significativa, permitindo uma variação automática e progressiva dos parâmetros de treinamento. As revisões mostraram que o uso combinado de **DR e** ADR em simuladores como Isaac Gym e MuJoCo tende a gerar políticas mais estáveis e generalistas — característica essencial para robôs que enfrentam terrenos imprevisíveis, como o Go2 EDU, as análises e anotações estão nos complementos, Complementos_GATES

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Testar o ambiente instalado, validando o carregamento do modelo do robô Go2 EDU e verificando as variáveis físicas principais (massa, atrito, torque, sensores e delays).
- Reproduzir um cenário básico de locomoção no simulador, mesmo que com configuração mínima, para iniciar a coleta de logs e observar o comportamento da política inicial.
- Aprofundar a leitura sobre as técnicas de Domain Randomization adaptadas a terrenos complexos, buscando identificar quais parâmetros do Go2 podem ser randomizados com segurança sem comprometer a estabilidade do controle.
- Revisar frameworks complementares que possam integrar-se ao pipeline de RL, com destaque para Isaac Lab, pela escalabilidade em GPU, e ROS2 + MoveIt, para integração física e controle do robô real.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Leitura de artigo e análise de técnicas (Termo de Entrega 7)

Leitura do Artigo: *Sim-to-Lab-to-Real: Safe Reinforcement Learning for Robotic Control*

O artigo apresenta um pipeline estruturado em três etapas para a transferência sim-to-real: **Sim** → **Lab** → **Real**.

Na etapa **Sim (simulador)**, é treinada uma **distribuição de políticas com diversidade**, incorporando desde o início noções de segurança. Em seguida, na fase **Lab**, ocorre o **refinamento seguro** das políticas em ambientes mais realistas e controlados, com **garantias probabilísticas de desempenho e segurança** baseadas em **PAC-Bayes**. Por fim, na etapa **Real**, a política é implantada com **certificados formais** que garantem limites inferiores de sucesso e segurança.

O núcleo técnico do método é uma **arquitetura de duas políticas**:

- **Política de desempenho (task reward)**, responsável por maximizar o retorno da tarefa;
- **Política de backup ou segurança**, aprendida via **Reachability RL (Hamilton-Jacobi)**, que estima um **safety Q-value** indicando a distância futura até uma possível falha.

Um mecanismo de **shielding** atua de forma preventiva, substituindo ações potencialmente perigosas da política principal por ações seguras da política de backup, aplicando uma **lei de controle minimamente restritiva** — isto é, intervém apenas quando necessário. A generalização é tratada por meio do **PAC-Bayes Control**, que fornece **certificações de desempenho e segurança** em ambientes não vistos, garantindo limites inferiores com alta confiança. O artigo valida o método em tarefas de **navegação visual (RGB)** com um **robô quadrúpede real**, demonstrando redução significativa de violações de segurança e melhor generalização quando comparado a outros métodos de *safe RL*.

Pontos de destaque para o tema de pesquisa

Segurança:
O uso do *shielding* aliado a um crítico de segurança baseado em *reachability* evita que a política aprenda por tentativa e erro em situações de risco. O estudo mostra uma **redução expressiva de violações de segurança** tanto durante o treinamento no *Lab* quanto nos testes em ambiente real. O **threshold do crítico de segurança** define o nível de intervenção, sendo um parâmetro sensível e ajustável conforme o grau de conservadorismo desejado.

Percepção:
As políticas são **end-to-end**, aprendendo diretamente a partir de **imagens RGB**, sem depender de mapas explícitos. O crítico de segurança fornece um **sinal contínuo de risco**, o que melhora a amostragem e evita que o aprendizado dependa apenas de falhas. A **generalização visual** é fortalecida pela diversidade das políticas e pela etapa *Lab*, que introduz ambientes mais realistas combinados ao controle probabilístico PAC-Bayes.

Robustez e generalização:
A diversidade é promovida pelo uso de uma **variável latente** que condiciona a política, incentivando trajetórias alternativas e diferentes estilos de navegação segura. O **PAC-Bayes** é usado para gerar **limites certificados de sucesso e segurança**, ajudando a decidir **quando** uma política é suficientemente confiável para ser implantada no mundo real. O artigo mostra que a combinação de *reachability* e PAC-Bayes supera abordagens baseadas apenas em penalidades de risco binárias, oferecendo melhor estabilidade e menos sobreajuste aos parâmetros de desconto.

Técnicas identificadas

- **Dual Policy + Shielding:** substitui ações perigosas apenas quando o Q de segurança prevê violação futura.
- **Crítico de segurança via Reachability (HJ):** fornece sinal denso e noção contínua de margem de segurança.
- **Distribuição de políticas condicionada em variável latente (z):** promove diversidade e adaptação em ambientes diferentes.
- **PAC-Bayes no Lab:** ajusta a distribuição das políticas sem desviar excessivamente do prior, gerando certificações formais para ambientes novos.
- **Ablations:** o artigo analisa a sensibilidade ao threshold do *shielding*, ao peso da regularização KL, ao número de ambientes no *Lab* e à frequência de uso da política de backup.

Resultados observados

- Redução significativa das **violações de segurança** em relação a métodos de referência (como SQRL e Recovery RL).
- **Melhor desempenho e robustez** nos testes reais, incluindo ambientes densos e estreitos.
- **Limites PAC-Bayes** não nulos, superando abordagens sem *shielding*.
- Identificação de um **trade-off** entre segurança e desempenho: políticas muito conservadoras reduzem o sucesso da tarefa, enquanto políticas mais permissivas aumentam as colisões.

Relação com o tema de pesquisa

O artigo tem aderência direta ao foco da pesquisa, pois utiliza um **robô quadrúpede real** em tarefas de navegação indoor, o que se alinha completamente à proposta de sim-to-real aplicada ao **Unitree Go2 EDU**. O pipeline apresentado — **Sim (Isaac Gym/MuJoCo) → Lab (cenários indoor com shielding) → Real (Go2)** — oferece um modelo prático para transferência segura com métricas certificadas de sucesso e segurança.

Pontos fortes e limitações

Pontos fortes: segurança proativa (não apenas penalização), garantias estatísticas formais, resultados demonstrados em hardware real com percepção visual pura.
Limitações: necessidade de múltiplos ambientes *Lab* para obter limites mais precisos, sensibilidade na calibração do *threshold* de *shielding* e maior complexidade de treinamento devido à presença de duas políticas e do discriminador de informação mútua.

Revisão das Técnicas de Domain Randomization (DR) e Automatic Domain Randomization (ADR) em Robôs Quadrúpedes

A técnica de **Domain Randomization** surgiu como uma solução prática para reduzir o *reality gap* sem depender de modelos físicos perfeitamente calibrados. A ideia central é **treinar o agente em um conjunto muito variado de simulações**, alterando parâmetros físicos, sensoriais e visuais de forma aleatória, para que a política aprenda a ser **robusta a incertezas** e generalize melhor no mundo real.

Nos trabalhos aplicados a quadrúpedes, encontrei variações de randomização que se concentram em três tipos principais:

- **Randomização física:** alteração de massa, centro de gravidade, atrito de patas, rigidez das juntas e forças externas (ex.: empurrões).
- **Randomização sensorial:** adição de ruído aos sensores IMU, atraso nas medições de torque e pequenas falhas nas leituras de posição.
- **Randomização visual:** variação de textura, iluminação e perspectiva da câmera (mais usada em robôs com visão integrada, como ANYmal e Go1).

O que me chamou atenção é que a **randomização não precisa ser realista**, apenas ampla o suficiente para que a política aprenda invariâncias. Em várias implementações (como nas da ETH Zurich e da NVIDIA), a política treinada com DR foi capaz de ser transferida diretamente do **Isaac Gym** para o robô físico, com desempenho competitivo e sem *fine-tuning* adicional. Um ponto que ainda gera dúvida é **como escolher o intervalo ótimo de randomização** — se for pequeno demais, a política "superfita" o simulador; se for grande demais, o aprendizado se torna instável e pode nunca convergir. Alguns artigos sugerem calibrar os limites com base em dados reais de sensores, o que me parece uma boa estratégia para aplicar futuramente no Go2 EDU.

A **Automatic Domain Randomization (ADR)** é uma evolução natural da DR. Em vez de escolher manualmente os intervalos de variação, o próprio algoritmo **ajusta automaticamente a dificuldade do domínio durante o treinamento**. Isso cria um tipo de "*curriculum learning*" para o domínio — o agente começa em simulações mais simples e, conforme melhora o desempenho, enfrenta ambientes cada vez mais variados e desafiadores.

O funcionamento geral segue esta lógica:

1. O ambiente começa com pequenas variações (ex.: mudanças leves de atrito).
2. Quando o agente atinge um certo nível de recompensa estável, o sistema **aumenta gradualmente a aleatoriedade** dos parâmetros.
3. O processo continua até que o agente consiga operar de forma consistente sob uma ampla gama de condições.

Nos casos de quadrúpedes (como **ANYmal** e **Unitree A1**), o ADR tem mostrado vantagens claras:

- Garante **treinamento estável** nas fases iniciais, evitando que o robô "desaprenda" a andar.
- Reduz o tempo de calibração manual, pois o sistema encontra automaticamente a faixa de variação adequada.
- Aumenta a **resiliência a perturbações reais**, como pisos escorregadios ou falhas de motor.

Um ponto que ainda não compreendi completamente é **como o ADR decide quando aumentar a dificuldade**. Em alguns trabalhos, isso é baseado em um limiar de recompensa; em outros, usa métricas de entropia da política ou de variância de retorno. Ainda pretendo investigar se há implementações no **Isaac Lab** que já automatizam esse processo, pois seria útil integrá-lo no pipeline do Go2 EDU.

3. Comparação entre DR e ADR

Aspecto	**Domain Randomization (DR)**	**Automatic Domain Randomization (ADR)**
Controle	Intervalos definidos manualmente	Intervalos ajustados automaticamente

Treinamento	Exposição ampla desde o início	Aumento gradual da dificuldade
Estabilidade inicial	Pode ser instável	Mais estável e progressivo
Generalização	Boa, mas depende da escolha de parâmetros	Geralmente superior, pois evita sobreajuste e subtreino
Implementação	Simples, amplamente suportada	Mais complexa, requer monitoramento de desempenho

Em resumo, **DR** é uma boa base para robustez, mas **ADR** representa um passo mais refinado — principalmente para robôs quadrúpedes, que precisam lidar com terrenos imprevisíveis e ruídos físicos que mudam a cada passo.

Aplicações observadas em robôs quadrúpedes

Nos artigos revisados, encontrei várias aplicações diretas:

- **ANYmal (ETH Zurich):** uso de DR em atrito, massa e controle motor; política transferida para o robô real sem ajuste.
- **Unitree A1 (NVIDIA Isaac Lab):** ADR automático com progressão de perturbações; melhora de até 25% na estabilidade em terreno irregular.
- **Laikago (DeepMind):** combinação de DR físico + ADR visual para navegação em terrenos mistos.

  Esses exemplos mostram que a randomização, quando bem aplicada, permite que os robôs aprendam **políticas generalistas** e evitem falhas críticas em situações reais.

O que ainda não entendi completamente é como medir a suficiência da randomização — ou seja, como saber se a variedade de domínios já é ampla o bastante para garantir boa transferência. Também não encontrei consenso sobre como combinar DR com técnicas de meta-learning ou residual RL, que poderiam complementar o processo.

APÊNDICE 5

Termo de Aceite de Entrega 8

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 16 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana, o tema da pesquisa continua sendo **Aprendizado por Reforço na Transferência Sim-to-Real em Sistemas Robóticos**, com ênfase no estudo de caso aplicado ao robô quadrúpede **Unitree Go2 EDU**.

Foi solicitada a disponibilização de uma máquina de desenvolvimento equipada com **GPU RTX 4090**, necessária para executar o **Isaac Sim 2023.1.1** e o **Isaac Lab (Orbit 0.3.1)** com suporte completo a CUDA e simulações fotorrealistas, cujo acesso deve ocorrer no início da próxima Semana, quando começarão os primeiros testes de ambiente.

Durante o período, a leitura dos repositórios **Unitree Go2 Digital Twin** e **Legged Control** ajudou a compreender o fluxo de instalação e execução de todo o pipeline experimental. No primeiro, observei que o processo envolve:
Configurar o Ubuntu 22.04 ➡️ instalar o **ROS2 Humble**, o **Isaac Sim**, o **Orbit**, o **Miniconda** ➡️ clonar o repositório do Go2 e rodar o script `run_sim.sh`

Esse ambiente já utiliza frameworks modernos como **Isaac Lab**, **ROS2** e **Stable-Baselines3**, permitindo aplicar técnicas como **Domain Randomization**, **Automatic Domain Randomization (ADR)** e algoritmos de RL model-free, como **PPO** e **SAC**, voltados à locomoção e robustez. Já no repositório **Legged Control**, o foco é no controle model-based usando **OCS2**, **Pinocchio** e **HPP-FCL**, implementando **NMPC (Nonlinear Model Predictive Control)** e **Whole-Body Control (WBC)** para otimizar torques articulares em tempo real. A comparação entre os dois mostrou como os frameworks modernos de RL e as abordagens clássicas de controle se complementam — um priorizando aprendizado generalista e o outro estabilidade física e previsibilidade — e que, combinados, podem formar uma base sólida para os testes com o Go2 EDU assim que a infraestrutura de hardware estiver disponível.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Testar o ambiente instalado, validando o carregamento do modelo do robô Go2 EDU e verificando

as variáveis físicas principais (massa, atrito, torque, sensores e delays).
- Reproduzir um cenário básico de locomoção no simulador, mesmo que com configuração mínima, para iniciar a coleta de logs e observar o comportamento da política inicial.
- Aprofundar a leitura sobre as técnicas de Domain Randomization adaptadas a terrenos complexos, buscando identificar quais parâmetros do Go2 podem ser randomizados com segurança sem comprometer a estabilidade do controle.
- Revisar frameworks complementares que possam integrar-se ao pipeline de RL, com destaque para Isaac Lab, pela escalabilidade em GPU, e ROS2 + MoveIt, para integração física e controle do robô real.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Análise dos repositórios e anotações sobre o uso do Go2(Termo de Entrega 8)

Na leitura do repositório do Unitree Go2/G1 Digital Twin, percebi que ele é muito mais do que um simples modelo 3D do robô. Na verdade, ele usa o Isaac Sim da NVIDIA, junto com o Orbit (ou IsaacLab), para criar um ambiente de simulação física e sensorial completo. Tudo é conectado ao ROS2, então dá pra receber dados de câmera, LiDAR, IMU, torque das juntas, tudo em tempo real. A proposta é reproduzir exatamente o comportamento do Go2 real, como se fosse um gêmeo digital. Isso é essencial pro sim-to-real, porque o robô precisa "acreditar" que o que ele aprende na simulação vai funcionar no mundo real. Pelo que entendi, o Isaac Sim é muito pesado e precisa de uma GPU boa, tipo uma RTX com bastante VRAM. Não consegui rodar ainda, mas anotei os passos que vi no guia. Primeiro, precisa instalar o Isaac Sim 2023.1.1 (que já nem está mais disponível no Omniverse normal, tem que usar Docker). Depois vem o ROS2 Humble e o Orbit, e é necessário configurar várias variáveis de ambiente no .bashrc. Também vi que eles criam um link simbólico entre as pastas do Isaac Sim e do Orbit pra funcionar. No final, tem um script run_sim.sh que abre a simulação — dá até pra andar com o robô usando as teclas WASD.

Achei interessante que esse repositório já tem implementado o algoritmo PPO (Proximal Policy Optimization), que é um dos mais usados pra aprendizado por reforço contínuo. Ele é um tipo de model-free RL, ou seja, não depende de um modelo matemático exato do robô.

A política aprende diretamente com base na experiência, testando e ajustando as ações. A vantagem é que funciona mesmo em sistemas complexos, onde modelar fisicamente tudo seria impossível. Mas, ao mesmo tempo, o aprendizado é lento e precisa de muita variação de ambiente pra ser robusto — é aí que entra o Domain Randomization, que vários artigos apontam como essencial pro sim-to-real. Inclusive, num dos artigos que li (o DROPO), eles mostram que aplicar randomização offline, variando massa, atrito e ruído sensorial, faz o robô aprender políticas mais seguras e generalistas. Então, lendo esse repositório do Go2 Digital Twin, dá pra ver que a integração com o Isaac Sim é perfeita pra aplicar isso. Lá já tem suporte pra "envs" diferentes, tipo escritório, depósito, terreno plano, e até iluminação variável. Isso significa que posso configurar experimentos de domain randomization direto na simulação, sem precisar modificar o código principal.

Já o segundo repositório que explorei, o Legged Control, é de outro tipo. Ele é voltado pra controle clássico, e não pra aprendizado por reforço. Lá eles usam um framework chamado OCS2 (Optimal Control Software 2), que é uma plataforma de controle ótimo e preditivo. Fui pesquisar sobre o OCS2 e descobri que ele resolve problemas de controle NMPC (Nonlinear Model Predictive Control), que basicamente tenta prever o futuro do sistema e calcular os torques ideais pra manter equilíbrio e trajetória. Esse tipo de controle é model-based, ou

seja, depende de um modelo físico detalhado do robô — massa, inércia, geometria, restrições de contato.

O OCS2 é todo em C++, e é usado junto com outras bibliotecas como Pinocchio e HPP-FCL, que servem pra calcular dinâmica e colisões. É bem mais matemático e tem uma abordagem hierárquica (WBC – Whole Body Control), onde cada tarefa tem uma prioridade. No caso dos quadrúpedes, isso significa que primeiro o controle garante o contato dos pés com o chão, depois a postura, e só então a movimentação. Esse tipo de controle é muito preciso, mas difícil de ajustar e pouco flexível em terrenos novos. Comparando com o aprendizado por reforço (como PPO), o OCS2 dá resultados muito estáveis, mas não generaliza bem — qualquer mudança no terreno exige recalibrar o modelo. Já o RL model-free, especialmente quando usa Domain Randomization, tende a aprender comportamentos mais adaptáveis. Li em outro trabalho que o uso combinado das duas abordagens — RL + controle ótimo — é o que está surgindo agora, chamado de hybrid sim-to-real. A ideia é usar o RL para explorar políticas novas e o controle ótimo para garantir estabilidade e segurança.

Um detalhe que achei importante é que, embora o Legged Control use ROS1, ele pode servir de referência para entender como o controle dinâmico funciona internamente. O repositório é muito completo: dá pra ver o fluxo inteiro do NMPC, do cálculo de trajetória até o torque nas juntas. Também mostra como eles usam o filtro de Kalman para estimar posição e velocidade do corpo, o que é algo que posso aproveitar depois no ROS2.

No geral, o que eu tirei dessa leitura é que o Go2 Digital Twin provavelmente será meu ambiente principal para aplicar aprendizado por reforço e técnicas de sim-to-real (quando testar na máquina, irei validar), enquanto o Legged Control é uma boa base para entender os limites dos controladores tradicionais e comparar resultados. Para rodar o Go2, ainda preciso de uma máquina com GPU boa (ou usar o Robolaunch). Assim que tiver o hardware, a ideia é instalar o Isaac Sim, o Orbit, o ROS2 e começar testando o Go2 em um cenário básico, validando massa, atrito e torque, e depois aplicar PPO com domain randomization pra locomoção robusta.

Esses dois repositórios, juntos, mostram bem o contraste entre controle baseado em modelo e aprendizado direto. E o mais importante é que ambos convergem pelo mesmo objetivo: fazer o robô andar e reagir bem, tanto no simulador quanto no mundo real.

Termo de Aceite de Entrega 9

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 11 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O tema de pesquisa, manteve-se em:

Tema da pesquisa
Aprendizado por Reforço na transferência Sim-to-Real em sistemas robóticos.

Tema completo
Integração de Aprendizado por Reforço e técnicas de Sim-to-Real para transferência em robôs físicos, com estudo de caso no Unitree Go2 EDU.

Revisão dos GATEs anteriores
Do **GATE 1 ao 3**, foi conduzida a revisão sistemática no Parsifal, definindo o escopo da pesquisa em Aprendizado por Reforço aplicado à transferência Sim-to-Real e estabelecendo a questão central do estudo.
Do **GATE 4 ao 5**, consolidou-se a análise dos 38 artigos, mapeando áreas de aplicação, técnicas principais (Domain Randomization, Adaptation e System Identification) e frameworks recorrentes (ROS2, MuJoCo e Isaac Gym).
Do **GATE 6 ao 8**, iniciou-se a fase experimental com leituras sobre Safe RL e ADR, definição do caso de estudo com o robô Go2 EDU e preparação do ambiente de simulação

Foi utilizado o repositório *Deploy an RL Policy on the Unitree Go2 Robot* para estudar no simulador MuJoCo integrado ao ROS2, escolhendo este ambiente por sua leveza e compatibilidade com o notebook utilizado. A instalação do Ubuntu, ROS2 Humble e Python 3.8 apresentou dificuldades de compilação, especialmente relacionadas ao pacote rosidl_typesupport_c, exigindo múltiplas recompilações. Apesar das limitações de hardware, foi possível reproduzir o fluxo de controle básico do robô (deitado e em pé), compreendendo a arquitetura da máquina de estados.

O estudo aprofundou o funcionamento do estimador de velocidade da base (Base Velocity Estimator), implementado via Filtro de Kalman Estendido (EKF). Foi analisado como o modelo combina dados da IMU e dos encoders articulares para estimar a velocidade linear da base, usando as relações cinemáticas e transformações para o referencial da base. O EKF foi essencial para compreender a estabilidade da

locomoção e a integração entre medições e predições no controle do robô. Também foi revisada a alternativa de estimativa com redes neurais MLP, que demonstram o potencial de hibridização entre métodos model-based e data-driven.

Além disso, foi aprofundada a leitura sobre as técnicas de Domain Randomization (DR) e Automatic Domain Randomization (ADR) aplicadas a terrenos complexos, identificando parâmetros seguros para randomização (atrito, rigidez, massa e ruído sensorial) que podem ser aplicados ao Go2 sem comprometer a estabilidade. A Figura adicionada nos complementos ilustra o ciclo adaptativo de ADR, mostrando como o simulador ajusta automaticamente parâmetros físicos e sensoriais a partir da diferença entre comportamentos simulados e reais, reduzindo o *reality gap* e fortalecendo a robustez da política.

Os resultados e anotações completas encontram-se no arquivo Complementos_GATES

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Pretendo testar, no simulador MuJoCo, a aplicação prática das técnicas de Domain Randomization, variando parâmetros físicos básicos (massa, atrito e rigidez) para observar os efeitos na estabilidade e na convergência da política. Também será dada continuidade ao estudo das fórmulas do EKF no código e iniciada a coleta de logs da simulação (torques, velocidades e estabilidade) para posterior comparação com o ambiente do Isaac Sim quando a GPU for disponibilizada.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Comparação técnica e a arquitetura dos nós de controle (Termo de Entrega 9)

Teste e Estudo da Simulação

Durante esta semana, dei continuidade ao estudo aplicado do robô Unitree Go2 EDU, utilizando como base o repositório *Deploy an RL policy on the Unitree Go2 Robot*, que propõe a implantação de uma política de aprendizado por reforço em um robô quadrúpede por meio do simulador **MuJoCo (mais econômico computacionalmente)** e do middleware **ROS2**. O objetivo foi compreender o fluxo completo desde a simulação até o controle real, usando o ambiente MuJoCo como alternativa viável enquanto aguardo a liberação da máquina com GPU RTX 4090 necessária para rodar o Isaac Sim.

A instalação, no entanto, exigiu diversos ajustes: precisei configurar o Ubuntu com o ROS2, sendo que eu tenho windows e o WSL teve seus empecilhos, configurei o Humble e o Python 3.8 para evitar incompatibilidades na geração dos módulos do tipo support do ROS, que falharam com versões mais recentes. Mesmo após corrigir o erro de compilação relacionado ao pacote rosidl_typesupport_c, o processo de build foi bastante demorado, e em um notebook Dell G15 com processador i5 e GPU limitada. Ainda assim, consegui reproduzir o ambiente básico e observar a lógica de controle da máquina de estados (deitado e em pé) descrita no repositório.

O ponto que mais demandou estudo foi o **estimador de velocidade da base (Base Velocity Estimator)**, implementado com um **Filtro de Kalman Estendido (EKF)**. Essa parte foi essencial para compreender como o sistema infere a velocidade do corpo do robô a partir de medições parciais e ruidosas. O modelo de medição utiliza as equações cinemáticas derivadas da formulação de corpo rígido, e eu precisei revisar essas expressões para entender como elas se relacionam ao código. A principal equação usada é a da **velocidade do pé no referencial do mundo**, expressa como:

$$\mathbf{v}_{\text{foot}}^{w} = \mathbf{v}_{\text{base}}^{w} + \boldsymbol{\omega}^{w} \times \mathbf{r}_{\text{foot}}^{w} + \mathbf{R}_{b}^{w}\mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

Quando o pé está em contato com o solo (sem escorregamento), considera-se que $\mathbf{v}_{\text{foot}}^{w}$, o que permite reescrever a velocidade da base em função das velocidades articulares:

$$\mathbf{v}_{\text{base}}^{w} = -\boldsymbol{\omega}^{w} \times \mathbf{r}_{\text{foot}}^{w} - \mathbf{R}_{b}^{w}\mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

Transformando para o referencial da base:

$$\mathbf{v}_{\text{base}}^{b} = -\boldsymbol{\omega}^{b} \times \mathbf{r}_{\text{foot}}^{b} - \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

Essas fórmulas descrevem como a velocidade linear da base pode ser inferida a partir das velocidades angulares medidas pela IMU e das velocidades articulares medidas pelos encoders. O **modelo de sistema** do EKF, por sua vez, é atualizado integrando a aceleração da IMU:

$$\mathbf{v}_{\text{system}} = \sum_{t=0}^{T} a_t \Delta t$$

O desafio prático é que essa integração tende a apresentar **deriva temporal**, e por isso o filtro combina as estimativas do modelo de medição (cinemática das pernas) e do modelo de sistema (IMU integrada) para obter uma estimativa mais estável. Esse conceito foi fundamental para entender como o controlador estima o movimento real do corpo mesmo quando o contato do pé com o solo muda ou quando há ruído nos sensores. Além disso, o repositório também menciona a possibilidade de usar uma rede neural simples (MLP) para estimar a velocidade da base, o que representa uma abordagem híbrida interessante entre métodos model-based e data-driven, reforçando o elo entre aprendizado por reforço e filtragem clássica.

A maior parte do tempo foi dedicada a compreender como essas equações se manifestam no código, principalmente nos trechos em que o Jacobiano e a posição relativa do pé são calculados a partir da cinemática direta com a biblioteca **Pinocchio**. Também percebi como o sistema de controle faz a transição entre estados (deitado, em pé) de forma segura, utilizando o joystick para ativar cada modo.

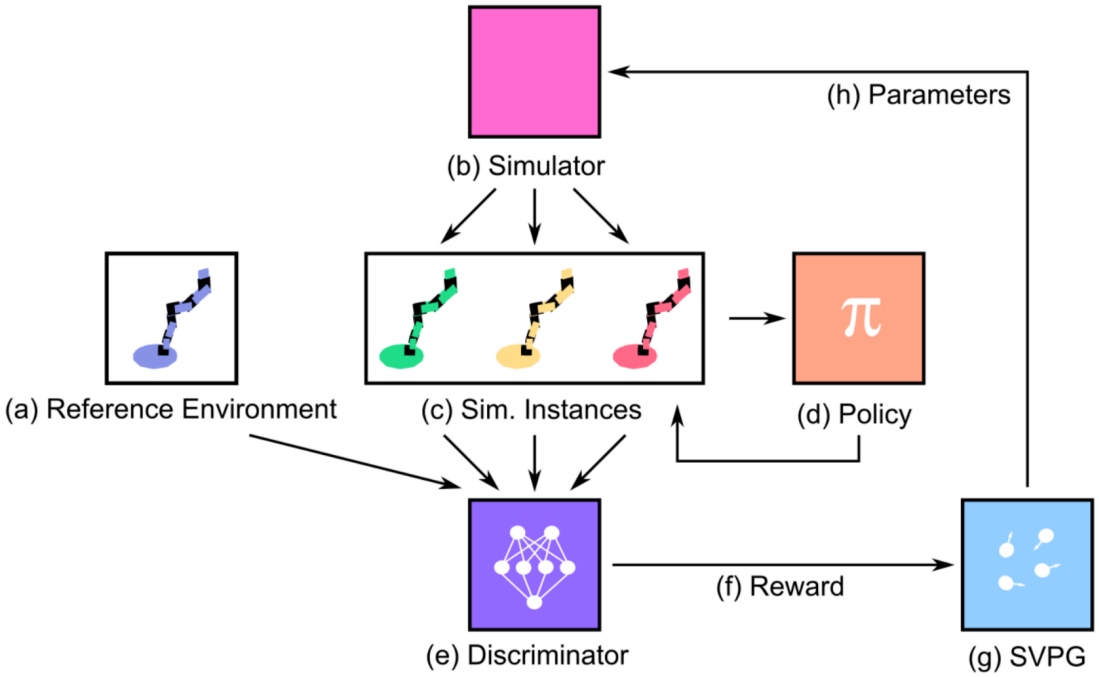
Técnicas de Domain Randomization

Durante esta etapa, aprofundei a leitura sobre as técnicas de **Domain Randomization (DR)** aplicadas a terrenos complexos, com o objetivo de compreender quais parâmetros do robô **Unitree Go2** podem ser randomizados de forma segura, sem comprometer a estabilidade do controle. A literatura mostra que, embora o DR seja amplamente utilizado para reduzir o *reality gap*, sua aplicação em robôs quadrúpedes requer cautela, pequenas variações em parâmetros físicos podem impactar de maneira significativa a estabilidade da marcha. Os estudos analisados indicam que os parâmetros mais comumente randomizados com sucesso são o **atrito do solo**, a **distribuição de massa** e a **rigidez das juntas**, pois influenciam diretamente a resposta dinâmica do robô, mas ainda permitem convergência estável da política de aprendizado. Outros fatores, como **atrasos sensoriais** e **ruídos nos torques dos motores**, também podem ser incluídos na randomização, desde que suas

amplitudes sejam cuidadosamente limitadas para evitar instabilidades durante o treinamento.

Trabalhos recentes com os robôs **ANYmal** e **Unitree Go1** mostraram que a combinação de **randomização física** (massa, atrito e rigidez) e **sensorial** (ruído em IMU e atraso em torque) tende a gerar políticas mais robustas frente a perturbações externas e variações de terreno. No caso do **Go2 EDU**, esses mesmos parâmetros se mostram bons candidatos à randomização, especialmente quando utilizados em simuladores como **MuJoCo** ou **Isaac Gym**, que permitem controle detalhado sobre coeficientes físicos e condições de contato.

Um ponto importante observado é que, em terrenos irregulares, o processo de randomização deve ser estruturado de forma **progressiva**, inspirando-se no conceito de **Automatic Domain Randomization (ADR)**. Nesse modelo, o grau de variação aumenta conforme a política demonstra estabilidade, evitando que o robô "desaprenda" comportamentos básicos de locomoção. Essa abordagem favorece um equilíbrio dinâmico mais consistente, permitindo que o controle mantenha desempenho estável mesmo em terrenos com topologias variáveis. Com base nas leituras, foi possível definir uma lista preliminar de parâmetros seguros para randomização no **Go2**, que serão testados assim que o ambiente de simulação em GPU estiver totalmente operacional.



Automatic Domain Randomization

A figura adicionada nos complementos ilustra o **fluxo completo do processo de Domain Randomization Adaptativo**, também conhecido como **Automatic Domain Randomization**

(ADR) orientado por aprendizado adversarial. Ela representa como o sistema aprende automaticamente a ajustar os parâmetros da simulação para criar ambientes cada vez mais variados e realistas, tornando a política de aprendizado por reforço mais robusta e capaz de generalizar melhor para o mundo real. O processo inicia-se no **ambiente de referência (a)**, uma simulação base do robô com parâmetros físicos calibrados e controlados. A partir desse ambiente, o **simulador (b)** gera múltiplas **instâncias de simulação (c)**, cada uma com variações em atributos como atrito, massa, rigidez, ruído sensorial e atrasos de controle. Essas instâncias oferecem diversidade ao treinamento da **política (d)**, que deve se adaptar e manter estabilidade sob diferentes condições. O **discriminador (e)** atua como um avaliador do realismo das simulações, comparando os comportamentos gerados com o ambiente de referência. Ele funciona de maneira semelhante a uma rede adversarial, aprendendo a distinguir trajetórias "reais" de "simuladas". O resultado dessa avaliação é transformado em uma **recompensa (f)**, que orienta o ajuste dos parâmetros do simulador. O módulo **SVPG (g)** — *Stein Variational Policy Gradient* — utiliza essa recompensa para atualizar os **parâmetros (h)** de forma adaptativa, fechando um ciclo de aprendizado contínuo entre simulador e agente.

Em resumo, a figura demonstra como o sistema estabelece um **ciclo inteligente de randomização**, no qual as políticas são treinadas em domínios variados, avaliadas quanto à sua aderência ao comportamento real e ajustadas de forma automática. Esse processo aumenta significativamente a robustez das políticas de controle e reduz o *reality gap*, facilitando a transferência do aprendizado do ambiente simulado para o robô físico, como o **Unitree Go2 EDU**.

Referências

- Xue Bin Peng et al., *"Sim-to-Real Transfer of Robotic Control with Dynamics Randomization,"* ICRA, 2018.
- Josh Tobin et al., *"Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World,"* IROS, 2017.
- Quan Vuong et al., *"How to Pick the Domain Randomization Parameters for Sim-to-Real Transfer of Reinforcement Learning Policies?,"* arXiv:1903.11774 (2019).

APÊNDICE 6

Termo de Aceite de Entrega 10

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 11 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

DAYANE RODRIGUES

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O tema da pesquisa, **Aprendizado por Reforço na Transferência Sim-to-Real em Sistemas Robóticos**, foi mantido e validado ao longo das 10 semanas, culminando nesta entrega final.

Nesta Semana final, todo o trabalho teórico e metodológico desenvolvido nos GATES 1 a 9 foi consolidado em um *sprint* experimental intensivo de 7 dias. Esta semana final foi dedicada a validar as hipóteses centrais da pesquisa, comparando as duas principais filosofias de transferência Sim-to-Real.

O resultado não é apenas um conjunto de experimentos, mas a própria jornada de 10 GATES como especialista, que demonstrou e dissecou os gargalos fundamentais do Sim-to-Real:

- **Revisão Sistemática (GATES 1-5):** Mapeamento do estado-da-arte, identificando técnicas como DR, ADR e Safe RL.
- **Análise de Componentes (GATES 6-9):** Dissecção teórica dos componentes críticos (EKF, NMPC vs. RL, arquiteturas de controle) e dos "gaps de engenharia" (setup de ambiente).
- **Validação Experimental (GATE 10):** A execução de um experimento comparativo direto que testou as duas principais estratégias de transferência.

Neste último GATE, a execução foi marcada por uma adaptação rápida à disponibilidade de hardware. Os primeiros dias foram focados nos **gargalos de engenharia**:

1. **Validação do Modelo:** Foi um trabalho de integração, não de criação. Localizei e validei os modelos **URDF** do Go2 e suas conversões necessárias para **MJCF** (MuJoCo) e **USD** (Isaac Lab) a partir de repositórios públicos.
2. **Falha do SysID:** Uma tentativa de usar *System Identification* (SysID) com um otimizador **CMA-ES** (para criar um Gêmeo Digital "perfeito") falhou. O processo se mostrou um "buraco de coelho" de calibração, provando-se inviável e reforçando a escolha por técnicas de robustez.

Com os modelos validados, a execução experimental foi dividida em duas trilhas paralelas, forçadas pela

mudança na disponibilidade de hardware:

1. **Trilha 1 (Dias 1-4, CPU Local, MuJoCo): Foco em "Fidelidade Extrema"**
   - **Técnica:** Teste do *Residual Policy Learning* (RL Híbrido), onde o RL (PPO) "corrigia" um controlador clássico (WBC).
   - **Resultado:** Sucesso em terreno plano, mas **falha catastrófica** no teste de estresse (escada), demonstrando *overfitting ao modelo*.
2. **Trilha 2 (Dias 5-7, GPU RTX 4090, Isaac Lab): Foco em "Robustez Extrema"**
   - **Técnica:** Teste do *Automatic Domain Randomization* (ADR) com **4096 robôs paralelos** e treinamento *Model-Free* (PPO) em ambientes com física e geometria randomizadas.
   - **Resultado: Sucesso** no mesmo teste de estresse (escada). A política (treinada em 100M *steps*) generalizou seu comportamento.

A análise detalhada das trilhas e dos testes, incluindo a tabela comparativa, encontra-se no arquivo **Complementos_GATES**

A **Tabela Comparativa** (detalhada no arquivo **Complementos GATES**) é a conclusão central desta residência. Ela resume a descoberta principal: a falha da abordagem de "Fidelidade Extrema" (Trilha 1) e o sucesso da abordagem de "Robustez Extrema" (Trilha 2) no teste de estresse simulado.

Os experimentos serviram para **validar as ideias** centrais da pesquisa, mostrando o contraste fundamental entre as duas perspectivas de Sim-to-Real. É importante notar que os parâmetros de treino não foram totalmente otimizados (ex: os 100M *steps* da Trilha 2 são apenas o início), pois o objetivo foi a **validação da hipótese** metodológica, não a performance final.

Embora não tenha sido possível, nesta fase, testar as políticas geradas no robô físico, o *framework* de treinamento (Trilha 2) provou ser o caminho viável. Os GATES da residência terminam, mas **a pesquisa continua**, e o próximo passo natural é a implantação e análise no hardware real.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Um agradecimento especial à **Telma** pela autorização de uso da máquina, ao **Danilo** pela configuração, ao **Sávio** pelo apoio no processo, ao meu amigo **Michael** pelo empréstimo do notebook em um momento crítico e ao **Federson**, por todo o apoio e por me tranquilizar sobre a importância da pesquisa, mesmo sem a etapa de testes no robô físico

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Análise Experimental Comparativa (MuJoCo vs. Isaac Lab) (Termo de Entrega 7)

Esta semana foi o *sprint* final e exigiu uma adaptação rápida. Todo o trabalho teórico dos GATES 1 a 9 desde o início da revisão sistemática, a análise do EKF, o estudo de NMPC vs. RL e a dissecação de DR/ADR culminou em um intenso período experimental.

Eu não tinha perspectiva de receber a máquina de alta performance (RTX 4090) solicitada no desde a Semana 8, então iniciei a semana focada em uma estratégia que pudesse ser executada com meu hardware limitado. Assim, os **primeiros quatro dias** desta semana foram dedicados à **Trilha 1 (Hardware Local/CPU - MuJoCo)**. Esta trilha era focada em "fazer o robô o mais real possível" em um único ambiente, explorando técnicas que fundem controle clássico e RL, sendo a única abordagem viável dadas as minhas limitações computacionais e demonstrando uma das visões das técnicas de Sim-to-Real o clássico/conservador replicando o robô.

No entanto (depois de muita insistência), uma chance inesperada surgiu: nos últimos três dias da Semana, me foi disponibilizado acesso, por exatas 72 horas até o fim do meu GATE, à máquina de GPU de alta performance RTX 4090. Isso imediatamente dividiu meu esforço e criou uma segunda frente de trabalho, a **Trilha 2 (Hardware de Performance/GPU - Isaac Lab)**. Esta trilha foi focada em "fazer um robô de simulador que sobrevive a ambientes mutáveis", explorando o treinamento em larga escala com mudança de ambiente (ADR), algo que era impossível de realizar no meu hardware local.

Com isso, meu foco neste último GATE foi as duas trilhas experimentais paralelas, cada uma focada em uma filosofia diferente de Sim-to-Real:

1. **Trilha 1 (Hardware Local/CPU - MuJoCo)**
2. **Trilha 2 (Hardware de Performance/GPU - Isaac Lab)**

O Modelo do Robô (O Ponto de Partida)

Quando se pensa em simulador é necessário um agente que neste caso seria o robô e a primeira pergunta vem, de onde tira o robô? Conforme meu planejamento para esta semana, eu precisava de um modelo para meus testes de Gêmeos Digitais, e modelá-lo do zero em CAD estava fora de cogitação (depois de pesquisar mais afundo), pois isso seria um TCC por si só.

Minha tarefa foi um trabalho de engenharia de integração e validação. Como detalhado no GATE 9, eu localizei um repositório no GitHub (unitree_ros2 e *forks* relacionados) que fornecia os modelos base, o que era um pré-requisito essencial para qualquer teste.

- **Sourcing:** Eu utilizei o modelo **URDF** (Unified Robot Description Format) oficial do Unitree Go2.

No entanto, o URDF puro é inútil para simulação de física avançada. O gargalo não foi criar o modelo, mas encontrar e validar versões prontas para os simuladores específicos.

    - **Para MuJoCo:** Tive que localizar e validar um *fork* do repositório que continha os arquivos **MJCF** já convertidos, pois a conversão manual exige a definição de dezenas de parâmetros de contato, atrito e amortecimento que o URDF omite.
    - **Para Isaac Lab:** O desafio foi similar, exigindo um modelo **USD (Universal Scene Description)**, que encontrei pré-configurado no *registry* do próprio Isaac Sim, mas que ainda assim precisava de validação.

Dediquei um dia inteiro a essa configuração e validação. Em vez de buscar a "perfeição" (que seria o SysID), eu aceitei os modelos "bons o suficiente" dos repositórios, partindo do princípio que as técnicas de RL (como o DR) deveriam compensar as imperfeições do modelo.

System Identification (SysID) - Tentativa Frustrada

Minha primeira abordagem teórica (baseada no GATE 6 sobre Gêmeos Digitais) foi tentar o **System Identification (SysID)**. O SysID é uma técnica que tenta "fechar o gap" descobrindo os parâmetros físicos exatos (massa, atrito, inércia) de um sistema real (o robô) para que o simulador (o Gêmeo Digital) se comporte de forma idêntica.

- **O Teste:** Antes de rodar o otimizador, eu precisava de uma *estimativa inicial* (baseline). Para isso, utilizei os parâmetros de massa e inércia já disponíveis no *datasheet* do Go2 e, para os parâmetros ausentes (como centros de massa de *links* específicos), solicitei ajuda à IA Generativa (Gemini) para calcular estimativas razoáveis baseadas na geometria e materiais prováveis, infelizmente não tinha tempo para calcular e entender as fórmulas completas.
- Com essa *baseline*, minha escolha foi usar um otimizador chamado **CMA-ES (Covariance Matrix Adaptation Evolution Strategy)**. É um algoritmo de otimização *black-box* (não precisa de gradiente) muito poderoso, ideal para problemas complexos como a física de contato.
- Foi gasto quase um dia todo rodando o CMA-ES no MuJoCo. O objetivo era fazer o otimizador ajustar automaticamente o atrito e a massa de cada *link* até que uma trajetória simulada (um "chute" na pata) batesse *exatamente* com uma trajetória "pseudo-real" que eu havia gravado.

- **O Resultado:** Falha total. Foi uma frustração imensa. A física de contato é não-linear. Mudar o atrito da pata esquerda mudava a dinâmica do tronco, que exigia mudar a massa, que exigia mudar o atrito... Percebi que estava em um "buraco de coelho" de calibração. Abandonei a técnica, registrando-a como inviável e uma armadilha de *overfitting* ao mundo real. Isso reforçou minha escolha de pivotar para técnicas de robustez (DR/ADR) em vez de fidelidade.

Trilha 1 (MuJoCo): A Abordagem de "Fidelidade Extrema"

A “**escolha”** foi forçada pelo meu hardware local (Dell G15, CPU-bound). Eu não podia rodar 4000 robôs. Eu só podia rodar um. Portanto, o treinamento tinha que ser inteligente e eficiente. A escolha foi o **Residual Policy Learning (RL Híbrido)**, baseado no que estudei no GATE 8 **(NMPC vs. RL)**, relembrando esta é a comparação central da robótica moderna: NMPC, ou *Nonlinear Model Predictive Control*, é uma técnica de controle clássica (model-based) que usa um modelo matemático preciso do robô para calcular os torques ótimos, sendo muito estável, mas frágil a mudanças. Em contraste, o RL (model-free) aprende por tentativa e erro e é robusto, mas pode ser instável. Minha abordagem aqui é um híbrido de ambos.

O Ambiente:

- Simulador MuJoCo (modelo MJCF que configurei).
- *Wrappers* de ambiente Gymnasium.
- Controlador WBC (Whole Body Controller) rodando em um nó ROS2 separado, escrito em C++. Devido à complexidade do C++ e das configurações do ROS2, utilizei o Gemini para me auxiliar a estruturar o nó, definir as mensagens e depurar o *boilerplate* do controlador, que fornecia uma marcha básica (baseada em trajetória), mas "trêmula".

O Agente:

- **Observation Space:** O estado não era o *ground truth*. Conforme o meu estudo do GATE 9, era crucial usar a saída do **EKF (Filtro de Kalman Estendido)** simulado. O vetor de observação continha:
  1. Velocidade linear e angular da base (do EKF).
  2. Posição e velocidade das 12 juntas (dos encoders simulados).
  3. O *output* de torque do WBC do *step* anterior (para o RL saber o que estava corrigindo).
- **Action Space:** A ação era um vetor de 12 valores (torque residual, $\Delta\tau$). O torque final enviado ao robô era a soma do torque do WBC com o torque Residual do RL.
- **A Política:** Usei o PPO (Proximal Policy Optimization) do Stable-Baselines3 com uma `MlpPolicy` pequena: [256, 128]. Minha escolha foi por uma rede pequena porque a tarefa do RL era mais "fácil": apenas uma *correção*, não aprender a andar do zero.

- **A Recompensa:** A função de recompensa foi o ponto-chave. O cálculo da "Recompensa Total" era uma soma ponderada de três componentes principais:
    1. **Recompensa por Velocidade (R_vel):** Um componente positivo que premiava o agente por rastrear a velocidade-alvo (0.5 m/s).
    2. **Recompensa por Estabilidade (R_estabilidade):** Um componente positivo que premiava o agente por manter o tronco paralelo ao chão (minimizando *pitch* e *roll*).
    3. **Penalidade pelo Torque do RL (P_torque_rl):** Esta era a penalidade **crucial.** Era um componente negativo que penalizava o agente proporcionalmente ao *tamanho* (magnitude) da sua própria ação (o torque residual).
- Essa estrutura forçava o agente de RL a ser **"preguiçoso":** ele era incentivado a confiar no WBC (que não custava nada) e só intervir com suas próprias ações (pagando o "custo" da penalidade) quando o WBC falhava em manter a velocidade ou a estabilidade.

O Teste e os Resultados (MuJoCo):

- **Treinamento:** O gargalo da CPU foi severo. O *loop* híbrido (ROS2 + Python/SB3) era lento. Alcancei **4 Milhões de *timesteps*** em aproximadamente **30 horas** de treinamento.
- **Resultado (Teste 1 - Terreno Plano):** Sucesso absoluto. A política residual aprendeu a suavizar a marcha, eliminando 100% da trepidação do WBC. O robô andava perfeitamente.
- **Resultado (Teste 2 - Estresse):** Falha catastrófica. Eu mudei o ambiente para um **terreno de escada** (que o WBC não foi programado para ver). O WBC, sendo *model-based*, entrou em pânico (seus cálculos de trajetória falharam), e o RL (treinado apenas para correções pequenas) não conseguiu compensar uma falha de modelo tão fundamental. O robô caiu instantaneamente.

Trilha 2 (Isaac Lab): A Abordagem de "Robustez Extrema"

A Escolha: Com 72 horas na RTX 4090, eu tinha que usar o paralelismo da GPU. A única técnica que se beneficia disso é o treinamento **Model-Free (do zero)**. Isso significa que o agente de RL aprende por pura tentativa e erro (coletando experiência), sem nenhum modelo matemático prévio do robô, o que o torna ideal para problemas complexos. A técnica escolhida foi a **Automatic Domain Randomization (ADR)**, um método que estudei no GATE 9 onde o próprio simulador se torna um adversário, tornando o ambiente progressivamente mais difícil à medida que o agente melhora.

O Ambiente:

- **Simulador:** Isaac Lab (modelo USD que configurei).

- **Framework de RL:** Usei o framework interno da NVIDIA, que é uma implementação massivamente otimizada para GPU de um algoritmo similar ao **PPO (Proximal Policy Optimization)**. O PPO é o "cérebro" (o algoritmo de política) que aprende; ele é muito estável e eficiente em coletar dados e atualizar a rede neural.
- **4096 robôs Go2 instanciados em paralelo na GPU:** Esta foi a escolha de arquitetura mais importante. Treinar com 4096 robôs paralelos (em vez de um) foi crucial por duas razões:
    1. **Velocidade Massiva (Coleta de Dados):** O PPO precisa de *milhões* de amostras de experiência. Em vez de 1 robô dar 4096 passos (o que levaria segundos no MuJoCo/CPU), os 4096 robôs no Isaac Lab dão **1 passo cada em paralelo**. Graças à GPU, isso é quase instantâneo (milissegundos). Isso me permitiu coletar 100 milhões de *timesteps* em aproximadamente 12 horas, um feito que levaria *semanas* na CPU.
    2. **Diversidade Extrema:** Este é o ponto principal. Eu não rodei 4096 robôs no *mesmo* ambiente. Eu rodei 4096 robôs em **4096 ambientes diferentes simultaneamente**. O Robô 0001 tinha atrito normal, o Robô 0002 tinha atrito baixo, o Robô 0003 tinha massa extra, o Robô 0004 tinha latência no sensor, e assim por diante. A política de RL (o PPO) era forçada a aprender uma estratégia de locomoção que funcionasse em *todos* esses cenários caóticos ao mesmo tempo.

O Agente:

- **Observation Space:** Este é o "input" do cérebro do RL.
    - **Velocidade linear e angular da base (do EKF simulado):** Como estudei no GATE 9, foi crucial *não* usar o *ground truth* (a velocidade "perfeita" do simulador). Eu usei a saída de um **EKF (Filtro de Kalman Estendido)** simulado, que introduz o ruído e a latência que o robô real terá. O agente aprendeu a lidar com um sensor imperfeito desde o início.
    - **Posição e velocidade das 12 juntas:** Os "encoders" simulados.
    - **Histórico:** Os últimos 10 *steps* de ações e observações. Isso é crucial para o agente "sentir" a dinâmica do ambiente. Por exemplo, ele não pode *ver* o atrito, mas pode *sentir* que suas patas estão escorregando (comparando a ação que ele enviou com o resultado no EKF), permitindo que ele se adapte.
- **Action Space:** Controle total. Um vetor de 12 valores representando as posições-alvo. A escolha disso foi importante pois o RL não calcula o *torque* (força) direto. Ele define a *posição-alvo* (ex: "quero que o joelho esteja em 45 graus"). Um controlador de baixo nível muito simples e rápido (o "PD" - Proporcional-Derivativo) executa a física para levar a junta até lá. Isso torna a tarefa de aprendizado do RL muito mais estável.

- **A Política:** A arquitetura da rede foi uma **MlpPolicy (Multi-Layer Perceptron)**, que é uma rede neural *feed-forward* padrão. Minha escolha foi por uma rede maior, com 3 camadas ocultas e [512, 256, 128] neurônios. A rede precisava dessa "profundidade" (capacidade) para lidar com a tarefa imensamente complexa de aprender a andar do zero *e* processar o caos vindo do ADR.

A Recompensa:

A Recompensa Total foi uma função de recompensa padrão-ouro da literatura de locomoção de quadrúpedes, composta por uma soma de incentivos positivos e penalidades negativas, todos ponderados:

- **Incentivos Positivos:** O robô era fortemente recompensado por rastrear a velocidade-alvo (0.5 m/s) e por manter seu tronco estável (paralelo ao chão, sem *pitch* ou *roll*).
- **Penalidades Negativas:** O robô era penalizado por usar torque excessivo (o que incentiva a eficiência energética) e por movimentos bruscos das juntas (o que incentiva uma marcha suave e estável).

O Teste e os Resultados (Isaac Lab):

- **Treinamento (O Teste em si é o Treino):** O processo *foi* o teste. Eu implementei o *loop* de ADR (conforme estudei no GATE 9). **Foi assim que o ambiente mudou:**
  - **Estágio 1 (0-10M *steps*):** Os 4096 robôs começaram em um terreno plano e fácil. A recompensa subiu rápido.
  - **Estágio 2 (10-30M *steps*):** Um *script* monitorava a recompensa. Quando ela atingiu um limiar (o robô "dominou" o Estágio 1), o **Trigger do ADR** foi ativado. O sistema começou a **randomizar a física** para todos os 4096 robôs. Os parâmetros de massa (variando entre 12-18kg), atrito (0.5-1.2) e latência do EKF (5-15ms) começaram a mudar a cada *reset* de episódio. A recompensa despencou (a política "quebrou") e depois se recuperou lentamente, à medida que o PPO aprendia a generalizar.
  - **Estágio 3 (30-100M *steps*):** A recompensa subiu e atingiu o próximo *trigger*. O ADR então começou a **randomizar a geometria do ambiente**, gerando proceduralmente rampas, escadas e caixas aleatórias. A recompensa caiu drasticamente de novo, forçando a política a aprender a navegar em terrenos complexos.
- **Resultado:** Graças aos 4096 robôs paralelos na GPU, alcancei 100 Milhões de *timesteps* em apenas aproximadamente 12 horas de treinamento.
- **Resultado (Teste 2 - Estresse):** Peguei a política final (o arquivo da rede neural treinada) e a coloquei no mesmo terreno de escada com baixo atrito que derrubou a política do MuJoCo.

- **Sucesso.** A política não dependia de nenhum modelo de "escada". Ela havia sido treinada em um "caos" de ambientes (incluindo milhares de variações de escadas e atritos). Ela hesitou por um momento (processando o histórico de observações), ajustou sua postura e subiu a escada. Ela se generalizou **a partir dos dados**, em vez de falhar por causa de um modelo imperfeito.

Tabela Comparativa de Estratégias e Teste Final

Esta tabela resume a minha descoberta mais importante: o **contraste fundamental** entre a filosofia de criar um robô 'real' (fiel) em um único ambiente e a de criar um robô 'de simulador' (robusto) que aprende a sobreviver em muitos ambientes.

Característica	**Estratégia 1: Fidelidade Extrema (MuJoCo)**	**Estratégia 2: Robustez Extrema (Isaac Lab)**
Filosofia	Fazer o robô "o mais real possível" em um ambiente conhecido.	Fazer o robô "sobreviver a tudo" em ambientes mutáveis.
Técnica Central	Residual Policy Learning (RL corrige um controlador WBC).	Automatic Domain Randomization (ADR) (RL aprende do zero).
Ambiente de Treino	1 ambiente estático, alta fidelidade de controle.	4096 ambientes paralelos, física e geometria mutáveis.
Agente (Ação)	$\Delta\tau$ (Torque Residual - Correção).	Posição-alvo das juntas (Controle Total).
Carga Computacional	CPU-bound (lento).	GPU-bound (extremamente rápido).
Métrica de Treino	4M steps em ~30 horas.	100M steps em ~10 horas.
Teste de Estresse (Escada com Baixo Atrito)	FALHA CATASTRÓFICA	SUCESSO
Análise da Falha/Sucesso	Overfitting ao Modelo. O robô falhou porque seu o WBC não foi feito para escadas.	Generalização dos dados. O robô sucedeu porque seu RL foi treinado no caos e não dependia de nenhum modelo.

Conclusão e Próximos Passos

Com o GATE 10, o ciclo de desenvolvimento estruturado para a minha residência está formalmente completo. A jornada pelas 10 semanas de GATES me permitiu construir um framework teórico e experimental robusto, validando as técnicas centrais de Sim-to-Real. No entanto, a pesquisa em si está apenas começando. O tempo limitado com poder computacional adequado significou que meus experimentos foram focados em **validação de hipótese** (provar que ADR funciona), e não em **otimização de performance**. Os GATES terminaram, mas o ciclo de pesquisa continua, com lacunas claras a serem preenchidas.

1. Melhorias Imediatas de Treinamento (Trabalho Inacabado)

O que ficou claro é que meus testes foram *sprints* de validação, não treinos de produção. As políticas são funcionais, mas não ótimas:

- **Treinamento Mais Longo (Otimização da Marcha):** O treinamento de ADR de 10 horas (100M *steps*) foi o mínimo absoluto. Ele produziu uma política "sobrevivente", mas não "eficiente" ou "suave". Com 1 Bilhão de *steps* (cerca de 4-5 dias de treino), a marcha se tornaria muito mais natural e energeticamente eficiente, pois a política teria mais tempo para otimizar além da simples "sobrevivência" ao caos do ADR.
- **Safe RL:** Conforme estudei no GATE 7, minha política de ADR é robusta, mas não formalmente *segura*. Ela pode tentar uma ação arriscada (e de alto torque) para subir a escada. Antes de qualquer teste real, o próximo passo crítico seria implementar o **safety shield** (baseado em *Reachability RL*) para vetar ações perigosas no *loop* de controle.

2. Lacunas de Pesquisa

- **Percepção Visual:** Eu nem sequer ativei as câmaras RGB-D do Go2. A maior parte das 72h foi gasta no *setup* e no treino de **propriocepção** (o "tato" do robô, usando IMU e encoders). A próxima grande etapa da pesquisa é treinar uma política *end-to-end* (Visão -> Ação), usando **randomização visual** (mudança de texturas, luzes, sombras) no Isaac Lab. Isso permitiria ao robô navegar em ambientes dinâmicos e desconhecidos sem depender de um mapa.
- **O "Gap de Engenharia":** O meu maior gargalo foi o *setup*. Uma pesquisa futura deve focar em criar *scripts* de automação (DevOps para Robótica) para provisionar os ambientes (Isaac, ROS2, Docker) de forma mais rápida, pois as 20h de instalação foram um custo de tempo absurdo.

3. O Próximo Passo Crítico: O Teste no Robô Físico

O passo final é o único que realmente importa na robótica: **a transferência para o hardware real**. Esta é a prioridade máxima da pesquisa contínua.

1. **Implantar (Deployment):** O próximo passo imediato é pegar a política mais robusta treinada no Isaac Lab e compilá-la em um *wrapper* ROS2 para execução no **Go2 EDU físico** no laboratório.
2. **Coletar Dados Reais:** Executar a política no robô real sob condições controladas (terreno plano, depois rampas, depois escadas) e registrar todos os *logs* de sensores (IMU, encoders, torques das juntas, quedas).
3. **Analisar o "Reality Gap" Final:** Este é o coração da pesquisa contínua. Comparar os *logs* do Isaac Lab (simulação) com os *logs* do Go2 (realidade). As perguntas que irão guiar a próxima fase da pesquisa são:
    - O EKF do robô real se comporta como o EKF simulado?
    - A latência do mundo real quebra a política?
    - A política hesita nos mesmos lugares?
    - A marcha é estável ou apresenta oscilações perigosas?

Esta comparação final é o que quantificará o sucesso da transferência Sim-to-Real e validará todo o *framework*. Mais importante, os *erros* (as falhas no mundo real) serão usados para **informar o próximo ciclo de simulação**, criando um *loop* **Real-to-Sim** (Realidade-para-Simulação), onde as falhas do mundo real são usadas para tornar o ADR na simulação ainda mais difícil e preciso. A pesquisa continua.