



UNIVERSIDADE  
ESTADUAL DE LONDRINA

---

LUIS FERNANDO KAWABATA DE ALMEIDA

**TÉCNICAS PREDITIVAS MAADR E DR EM JOGOS  
MULTIJOGADORES:  
AVALIAÇÃO DO ERRO EM DIFERENTES LATÊNCIAS**

---

Londrina  
2019

LUIS FERNANDO KAWABATA DE ALMEIDA

**TÉCNICAS PREDITIVAS MAADR E DR EM JOGOS  
MULTIJOADORES:  
AVALIAÇÃO DO ERRO EM DIFERENTES LATÊNCIAS**

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Alan Salvany Felinto

Londrina  
2019

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

Almeida, Luis Fernando Kawabata de.

Técnicas preditivas MAADR e DR em jogos multijogadores : Avaliação do erro em diferentes latências / Luis Fernando Kawabata de Almeida. - Londrina, 2019.  
63 f. : il.

Orientador: Alan Salvany Felinto.

Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2019.

Inclui bibliografia.

1. Técnicas preditivas - Tese. 2. Dead Reckoning - Tese. 3. Ambiente virtual distribuído - Tese. 4. Latência - Tese. I. Salvany Felinto, Alan. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

**LUIS FERNANDO KAWABATA DE ALMEIDA**

**TÉCNICAS PREDITIVAS MAADR E DR EM JOGOS  
MULTIJOGADORES:  
AVALIAÇÃO DO ERRO EM DIFERENTES LATÊNCIAS**

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

**BANCA EXAMINADORA**

---

Orientador: Prof. Dr. Alan Salvany Felinto  
Universidade Estadual de Londrina – UEL

---

Prof. Dr. Elieser Botelho Manhas Junior  
Universidade Estadual de Londrina – UEL

---

Prof. Dr. Silvano Cesar da Costa  
Universidade Estadual de Londrina – UEL

Londrina, 30 de Janeiro de 2019.

*Este trabalho é dedicado aos sonhadores que  
mesmo diante as adversidades da vida e da  
sociedade não deixaram de sonhar*

## **AGRADECIMENTO**

Agradeço profundamente a todos os presentes nessa minha jornada. Todos os familiares e amigos que me ajudaram a me manter focado no que mais importa. Agradeço especialmente ao amor da minha vida.

ALMEIDA, Luis Fernando Kawabata de. **Técnicas preditivas MAADR e DR em jogos multijogadores: avaliação do erro em diferentes latências.** 2019. 63 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2019.

## RESUMO

Jogos multijogador contêm mundos virtuais que conectam dezenas de jogadores em uma mesma sessão, resultando em uma grande sensibilidade para atrasos de rede. Uma gama de técnicas preditivas para posições dos jogadores foram desenvolvidas para entregar a melhor experiência para os jogadores. Porém com o grande aumento do número de jogadores presentes e a complexidade das sessões, as limitações de tais técnicas, como atrasos na rede, consistência da informação compartilhada, responsividade aos jogadores e custo de banda de rede, se tornam evidentes. A principal técnica usada hoje em dia é chamada Dead Reckoning (DR). Tomando essa técnica como base, diversos autores propuseram melhorias complexas ao método de predição, culminando na técnica proposta por Kharitonov chamada Motion-Aware Adaptive Dead Reckoning (MAADR). Kharitonov não avaliou o desempenho da técnica MAADR considerando atrasos na rede. Também, até o momento, não foram encontrados, na literatura, trabalhos que relacionam o desempenho da MAADR sob diferentes atrasos da rede, que é um fator com grande influência à consistência dos algoritmos preditivos. A técnica MAADR se mostrou ter um desempenho superior às outras em circunstâncias normais de atrasos de rede. O objetivo desse trabalho é avaliar ambas as técnicas, MAADR e DR, em 5 cenários com diferentes características e complexidades, com atrasos de rede simulados que variam de 0 até 300 ms e representam baixas, médias e altas latências na rede. Os resultados obtidos nesse trabalho mostram que é recomendado o uso da técnica MAADR em circunstâncias em que médios e altos atrasos na rede são esperados ou quando há uma intolerância para a perda da Qualidade de Serviço no jogo, devido ao atraso na rede, e com isso a perda da qualidade da experiência do jogador.

**Palavras-chave:** Dead reckoning, Latência. MAADR. Técnicas preditivas. Ambientes virtuais distribuídos. Unity3D.

ALMEIDA, Luis Fernando Kawabata de. **Evaluation of the Motion-Aware Adaptive Dead Reckoning technique under different network latencies applied in multiplayer games.** 2019. 63 p. Dissertation (Master's degree in Computer Science) – Universidade Estadual de Londrina, Londrina, 2019.

## ABSTRACT

Multiplayer games hold virtual worlds which connect dozens of players in the same session, resulting in a great sensibility for network latency. A various number of player position prediction techniques were developed to deliver the best experience for the players. But with the growing numbers of players and the complexity of sessions, the limitations of such techniques, like network delay, consistency of the shared information, player responsivity and bandwidth cost, becomes evident. The main technique used nowadays is called Dead Reckoning (DR). Taking the DR as a base, several authors proposed improvements to the prediction method, culminating on Kharitonov's proposal technique called Motion-Aware Adaptive Dead Reckoning (MAADR). Kharitonov did not evaluate the performance of the MAADR technique considering network latencies. On the literature, until this moment, were not found any papers that relate the performance of the MAADR technique under different network latencies, which is an important factor for consistency on predictive techniques. With no latencies considered, the MAADR showed good results when compared with other techniques. The proposal of this paper is to evaluate both the MAADR and DR prediction techniques in 5 scenarios with different characteristics and complexities, with simulated network latencies that vary from 0 up to 300 ms and that represent low, medium and high network latency values. The results show that it is recommended the usage of the MAADR technique in circumstances where medium and high network latencies are expected or when intolerance for the loss of QoS, caused by latency, is unwanted.

**Keywords:** Dead reckoning. DR. Latency. MAADR. Predictive techniques. Distributed virtual environments. Unity3D.



## LISTA DE ILUSTRAÇÕES

Figura 1	–	Comportamento de um objeto local e seu respectivo objeto remoto usando a técnica DR para prever a trajetória correta, adaptado de Kharitonov .....	24
Figura 2	–	Correção da trajetória do objeto remoto via interpolação .....	25
Figura 3	–	Relações entre objeto local e objeto remoto dentro da técnica MAADR, adaptado de Kharitonov .....	27
Figura 4	–	Padrões de movimento implementados em dois módulos, adaptado de Kharitonov para controle facilitado .....	33
Figura 5	–	Cenário 1: movimento linear simples .....	37
Figura 6	–	Perfis de velocidade para o cenário 1: Movimento linear .....	38
Figura 7	–	Gráfico de radar com a média dos erros de ambas as técnicas para cada valor de latência para o cenário 1 .....	38
Figura 8	–	Cenário 1: Médias dos erros e desvios padrões ao longo das latências .....	39
Figura 9	–	Médias das diferenças dos erros e grupos para o cenário 1 .....	40
Figura 10	–	Cenário 2: Movimento Circular .....	41
Figura 11	–	Perfis de velocidade para o cenário 2: Movimento Circular .....	42
Figura 12	–	Gráfico de radar com o erro de ambas as técnicas para cada valor de latência para o cenário 2 .....	42
Figura 13	–	Cenário 2: Média dos erros e desvios padrões ao longo das latências para ambas as técnicas .....	43
Figura 14	–	Médias das diferenças dos erros e grupos para o cenário 2 .....	44
Figura 15	–	Cenário 3: Movimento Espiral .....	45
Figura 16	–	Perfis de velocidade para o cenário 3: Movimento Circular Variado .....	45
Figura 17	–	Gráfico de radar com o erro de ambas as técnicas para cada valor de latência para o cenário 3 .....	46
Figura 18	–	Cenário 3: Médias dos erros e desvios padrões através das latências .....	46
Figura 19	–	Médias das diferenças dos erros e grupos para o cenário 3 .....	47
Figura 20	–	Cenário 4: Movimento Zigue-zague .....	48
Figura 21	–	Perfis de velocidade para o cenário 4: Movimento de Zigue-zague .....	49
Figura 22	–	Gráfico de radar com a média dos erros de ambas as técnicas para cada valor de latência no cenário 4 .....	49
Figura 23	–	Cenário 4: Médias dos erros e desvios padrões através das latências .....	50

Figura 24 – Médias das diferenças dos erros e grupos para o cenário 4.....	51
Figura 25 – Cenário 5: Movimento aleatório.....	52
Figura 26 – Perfis de velocidade para o cenário 5: Movimento aleatório .....	52
Figura 27 – Gráfico de radar com o erro de ambas as técnicas para cada valor de latência para o cenário 5 .....	53
Figura 28 – Cenário 5: Médias dos erros e desvios padrões através das latências .....	53
Figura 29 – Médias das diferenças dos erros e grupos para o cenário 5.....	54
Figura 30 – Gráfico de radar com as médias dos erros de cada técnica para cada Cenário sob 0 ms de latência simulada.....	55

## LISTA DE TABELAS

Tabela 1	– P-Valores do teste MW entre as duas técnicas para cada latência no cenário 1 .....	39
Tabela 2	– P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 2 .....	43
Tabela 3	– P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 3 .....	47
Tabela 4	– P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 4 .....	50
Tabela 5	– P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 5 .....	54
Tabela 6	– Tabela da média dos erros de cada técnica em cada cenário .....	56

## LISTA DE ABREVIATURAS E SIGLAS

ADR	<i>Auto-Adaptive Dead Reckoning</i>
AMK	<i>Angular Motion Kernel</i>
API	<i>Application Programming Interface</i>
AVD	<i>Ambiente Virtual Distribuído</i>
CSGO	<i>Counter Strike Global Offensive</i>
DOTA	<i>Defense Of The Ancients</i>
DR	<i>Dead Reckoning</i>
FPS	<i>First Person Shooter</i>
GPU	<i>Graphic's Processing Unit</i>
GS-DR-LL	<i>Globally Synchronized Dead Reckoning Local Lag</i>
KW	<i>Kruskal-Wallis</i>
LMK	<i>Linear Motion Kernel</i>
MDR	<i>Markov Decision Process</i>
MAADR	<i>Motion-Aware Adaptive Dead Reckoning</i>
MW	<i>Wilcoxon-Mann-Whitney</i>
QoS	<i>Quality of Service</i>
RAM	<i>Random Access Memory</i>
RPG	<i>Role-Playing Game</i>
RTS	<i>Real Time Strategy</i>
SW	<i>Shapiro-Wilk</i>
TSS	<i>Trailing State Synchronization</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>14</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>18</b>
<b>2.1</b>	<b>Tipos de gêneros dos jogos eletrônicos .....</b>	<b>18</b>
<b>2.2</b>	<b>Ambientes Virtuais Distribuídos.....</b>	<b>19</b>
<b>2.3</b>	<b>As propostas da literatura .....</b>	<b>21</b>
<b>2.4</b>	<b>Técnicas que priorizam a consistência .....</b>	<b>21</b>
<b>2.5</b>	<b>Técnicas que priorizam a responsividade .....</b>	<b>23</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS.....</b>	<b>29</b>
<b>3.1</b>	<b>Arquitetura de rede.....</b>	<b>29</b>
<b>3.2</b>	<b>Implementações do DR e do MAADR.....</b>	<b>30</b>
<b>4</b>	<b>ANÁLISE E RESULTADOS.....</b>	<b>35</b>
<b>4.1</b>	<b>Considerações gerais da análise dos resultados.....</b>	<b>35</b>
<b>4.2</b>	<b>Cenário 1: Linha com Aceleração Constante .....</b>	<b>36</b>
<b>4.3</b>	<b>Cenário 2: Movimento Circular.....</b>	<b>41</b>
<b>4.4</b>	<b>Cenário 3: Movimento Circular Variado.....</b>	<b>44</b>
<b>4.5</b>	<b>Cenário 4: Movimento zigue-zague. ....</b>	<b>48</b>
<b>4.6</b>	<b>Cenário 5: Movimento aleatório .....</b>	<b>51</b>
<b>4.7</b>	<b>Discussão .....</b>	<b>55</b>
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>58</b>
	<b>REFERÊNCIAS .....</b>	<b>59</b>
	<b>Trabalhos Publicados pelo Autor.....</b>	<b>63</b>

# 1 INTRODUÇÃO

Os jogos fazem parte da sociedade há milhares de anos, desde a Grécia antiga, onde os gregos se reuniam para grandes eventos que envolviam jogos como corridas pedestres, corridas equinas, lutas, entre outras. Tais eventos inspiraram o surgimento das olimpíadas que conhecemos atualmente. As primeiras descobertas de jogos realizados por indivíduos foram os jogos de tabuleiro encontrados em escavações datando 5.000 anos no Egito e na Grécia, tendo posteriormente sendo difundidos no restante da Europa e nas Américas. Os jogos têm como objetivo principal o entretenimento do indivíduo, com o prazer e o aspecto lúdico. Há diversos tipos de jogos como, por exemplo, jogos de tabuleiros, de cartas, de mesa, os que envolvem movimentos corporais, como os esportes, dentre muitos outros. Em todos os jogos há características comuns, são elas os conjuntos de regras e elementos pertencentes a cada categoria, que os jogadores devem seguir e que estipulam os limites de cada jogo [1].

A presença dos jogos é muito forte em nossa sociedade. Com o passar do tempo houve o surgimento de diferentes tipos de jogos e, devido à valorização destes, temos um grande mercado voltado para atender esse público.

Atualmente, com o grande avanço tecnológico, os jogos ganharam a perspectiva de entreter em plataformas eletrônicas, cunhando o conceito dos jogos eletrônicos ou digitais. Esses jogos eletrônicos são atividades lúdicas, em que o usuário age, decide e é condicionado pelas regras presentes no mundo virtual. Essas regras geram narrativas e criam experiências intrigantes e divertidas [1].

Desde o seu início na década de 70, o mercado de jogos eletrônicos cresceu exponencialmente. Mesmo com a mudança no seu formato de fliperamas para consoles portáteis, o mercado continuou crescendo, chegando a ultrapassar o mercado cinematográfico em meados dos anos 2.000, movendo cerca de 116 bilhões de dólares por todo o mundo em 2017 e tendo uma estimada movimentação de 143,5 bilhões para 2020 [2].

Inicialmente, jogos eram feitos com foco para experiências de um único jogador ou multijogador com um mesmo hardware, como no caso da tela dividida no monitor. Porém, com os avanços tecnológicos na área de rede de computadores, esses jogos ganharam a perspectiva de conectar jogadores fisicamente distantes para grandes aventuras cooperativas ou mesmo competitivas. Essa característica, cresceu a ponto da criação de uma categoria para jogos online competitivos, chamada Esportes Eletrônicos (Esports) [3].

A conexão presente em jogos digitais online propõe um nível não precedente de interação com grande distância, possibilitando que jogadores coordenem suas ações em conjunto para alcançar objetivos e derrotar inimigos. Alguns desses jogos envolvem dezenas ou mesmo centenas de jogadores na mesma sessão do jogo em tempo real, o que faz com que uma grande

quantidade de informação tenha necessidade de ser compartilhada. Esse nível de interação é especialmente grande em jogos com alto nível de competitividade, como nos jogos anteriormente citados de Esports e nos jogos atuais, mesmo que não pertençam à categoria de Esports [3].

Para possibilitar AVDs que possibilitem as grandes escalas e consistência, como os presentes nos tipos de jogos citados anteriormente, uma gama de técnicas que modelam os problemas e as arquiteturas foram propostas. Algumas das principais técnicas utilizadas, como definições de áreas de interesse, a técnica chamada Dead Reckoning entre outras, foram propostas na década de 90. Apesar de serem antigas, essas técnicas são tratadas como parte do estado da arte pela indústria, sendo utilizadas na maioria dos casos. Inicialmente, essas técnicas não levaram em consideração o fator de escalabilidade que as aplicações teriam conforme o avanço tecnológico. O crescente número de usuários presente em uma mesma sessão de um jogo, e também o crescente tamanho dos mundos virtuais, com cada vez mais detalhes, fez com que tais técnicas ficassem com seus problemas em grande evidência. Com isso, surge a necessidade de mais estudos que busquem diminuir as consequências das limitações nessas áreas [4].

Jogos eletrônicos multijogadores implementam algoritmos que estão no grupo daqueles definidos como Ambientes Virtuais Distribuídos (AVDs) [5]. Um jogo online é usualmente construído com base em uma arquitetura de cliente-servidor, em que o servidor tem que validar a lógica compartilhada e os clientes têm que renderizar os elementos e as consequências das ações dos jogadores como, por exemplo, personagens e tiros [6]. Por questões de otimização no uso do servidor, é comum que alguns desenvolvedores decidam implementar o AVD de forma que os cálculos da física de objetos e de animações não essenciais sejam feitas somente nos clientes, com o fim de não sobrecarregar os servidores com o custo desses cálculos.

Dentro dos AVDs, mundos virtuais são criados com o fim de propor uma experiência com grande imersão ao jogador. Dentro desses AVDs há dois problemas principais [7] [8]. O primeiro é relacionado à informação contida nesses mundos virtuais e o segundo é relacionado ao compartilhamento dessas informações com os clientes. O primeiro problema se evidencia com o crescimento dos mundos e da quantidade de informação atrelada à complexidade desses mundos. Essa quantidade de informações, que precisam ser compartilhadas com todos os clientes, cresceu exponencialmente, de forma que podem exceder o custo de banda viável. O segundo problema surge quando o primeiro problema é resolvido, de forma que mesmo selecionando uma quantidade de informação viável, a forma do gerenciamento dessa informação ainda precisa ser decidida.

Para lidar com esses problemas, duas linhas de pesquisas foram criadas. A primeira linha de pesquisa abrange o primeiro problema, que é relativo à quantidade de informação dentro de um AVD que precisaria ser compartilhada com todos os jogadores. Nesta linha de pesquisa, a principal proposta é a de áreas de interesse, de forma que somente as informações consideradas como interessantes para um dado cliente é compartilhada com ele [7].

A segunda linha de pesquisa é relacionada ao gerenciamento das informações identi-

cadadas como interessantes entre os clientes e servidores. As informações de posições dos personagens são notórias nas pesquisas desta linha, pois a presença da inconsistência de seus valores pode causar grandes prejuízos à Qualidade do Serviço (QoS) aos jogadores [8].

Dentro desta linha de pesquisa há um grande número de técnicas preditivas propostas por diversos autores. Essas técnicas necessitam balancear o consumo de banda de rede e a consistência entre cada objeto local e remoto [9]. Tais técnicas preditivas podem ser categorizadas entre aquelas que priorizam consistência entre entidades remotas e locais, e aquelas que priorizam a responsividade para o jogador controlando entidades locais, com ambas tendo como finalidade uma técnica com boa consistência entre as informações compartilhadas [10]. As técnicas preditivas podem também fazer parte de uma sub-categoria de modelagens comportamentais para prever o comportamento dos jogadores [11].

O principal algoritmo representativo da categoria que prioriza a responsividade é chamado Dead Reckoning (DR) e consegue aumentar significativamente a consistência entre entidades, sem perder responsividade e simplicidade. A técnica DR é simples e, em vários casos, diminui o erro para níveis aceitáveis da perspectiva dos desenvolvedores. Por isso a DR se tornou uma técnica clássica e amplamente usada na indústria. Alguns jogos clássicos que a utilizam são Quake [12] e Half Life [13]. O amplo uso dessa técnica fez com que diversos autores propusessem melhorias à ela, com o propósito de criar uma técnica melhor, propondo mudanças na parte preditiva ou na parte de condição para taxa de atualização de informações, ou mesmo ambas [14].

Apesar de as melhorias propostas terem bons resultados, todas as citadas neste trabalho, com exceção da proposta de Aggarwall [14], não avaliam o desempenho de suas técnicas de forma comparativa sob diferentes valores de latências. Sob presença de médios e altos valores de latência, tendem a diminuir consideravelmente a QoS [15], tornando este um ponto que deveria ser considerado [14].

Do estudo da literatura e da análise de como a proposta de cada autor incrementou e melhorou a técnica DR, Kharitonov *et al* [9] propuseram a técnica chamada *Motion-Aware Adaptive Dead Reckoning (Dead Reckoning Adaptativo Ciente de Movimento, MAADR)*. Os seus resultados mostraram que o algoritmo proposto tem uma melhor performance que o DR clássico. Apesar da presença da avaliação da precisão do algoritmo, a presença da latência foi desconsiderada.

A proposta deste trabalho é, portanto, uma análise sobre como a latência afeta a efetividade das técnicas preditivas e comportamentais. Tal análise foi feita comparando-se a performance de dois algoritmos proeminentes na literatura. O primeiro algoritmo é o MAADR, que é o algoritmo resultado das diversas melhorias sobre a técnica DR encontrados na literatura até então, e o segundo é a própria técnica clássica DR com marcas temporais, que ainda hoje é a técnica preditiva mais usada na indústria. Para avaliá-las, ambos os algoritmos foram submetidos a 5 diferentes cenários inspirados em avaliações encontradas na literatura, com entradas



de dados de jogadores lidos de um arquivo de texto, a fim de manter consistência entre testes, simulando desde movimentos simples até movimentos complexos.

Os próximos capítulos dessa dissertação se seguem da seguinte forma. O capítulo 2 contextualiza ambas as técnicas avaliadas, assim como o desenvolvimento da literatura até chegar nelas. O capítulo 3 expõe a escolha das ferramentas assim como o que foi implementado para possibilitar a avaliação. O capítulo 4 inicialmente explica os cenários usados na avaliação e os testes estatísticos usados para avaliar os resultados e posteriormente expõe os resultados, tanto graficamente, quanto com análises estatísticas, discutindo-os. Por fim o capítulo 5 contém a conclusão da dissertação.

## 2 FUNDAMENTAÇÃO TEÓRICA

As técnicas propostas na literatura têm diversas características que podem definir os seus usos. Algumas fazem parte de uma cronologia de melhorias aplicadas em uma técnica base. Outras são melhores em cenários específicos, ou mesmo melhores em jogos com gêneros específicos.

### 2.1 Tipos de gêneros dos jogos eletrônicos

Os jogos foram criados para entreter seus jogadores seguindo as limitações de seus componentes físicos e implementando o design possível relativo a essas limitações. Conforme os anos se passaram, e a tecnologia para executar jogos foi aumentando, alguns jogos se destacaram por suas inovações e grande popularidade. Essa popularidade fez com que desenvolvedores desenvolvessem jogos inspirados no que estava fazendo sucesso em cada época, o que, por fim, levou à criação de gêneros que indicam como um jogo é de forma geral, com objetivo de ajustar as expectativas dos jogadores.

Um bom exemplo de uma criação e evolução de um gênero é o nascimento da categoria de jogos de aventura de texto (*text adventures*) [16], ou ficção interativa, como eram também conhecidos, que costumavam propor grandes aventuras exploratórias para o jogador, com grande possibilidade de interações com o mundo do jogo, porém, limitada a mostrar somente textos sobre o que estava acontecendo no jogo. Com o avanço da tecnologia, o gênero evoluiu para aventuras gráficas. Com o mesmo foco em exploração, o jogo possibilita ao jogador interagir usando o cursor com os diversos objetos presentes à vista, com respostas textuais dos personagens e a possibilidade de ver as consequências de suas ações em uma resposta visual. A grande mudança que a presença de gráficos causou nessa categoria de jogos fez com que o gênero evoluísse, ganhando um novo nome de "apontar e clicar" (*Point-and-click*) [17]. Os diferentes gêneros existentes propõem características de jogos diferentes, com jogabilidades diferentes e, conseqüentemente, comportamento dos jogadores diferente.

Alguns gêneros destacados possuem diferenças nas suas formas de jogar e interagir com o jogo. Como exemplo, tem-se os jogos de Tiro em primeira pessoa (*First Person Shooter*, FPS), que teve seus primeiros experimentos na década de 70, com jogos como Maze War e Spasim, mas a real popularização do gênero de FPS surgiu na década de 90. O jogo que tornaria definitiva a popularidade do gênero FPS seria lançado no de 1993, DOOM [18] pela Id Software, levando as técnicas propostas um passo à frente e mostrando a capacidade dos jogos feitos para computadores na época. Com gigantesca popularidade, e diversos jogos inspirados posteriores, o gênero FPS ficou consolidado na indústria dos jogos eletrônicos e atualmente temos diversos jogos sendo lançados todos os anos com o gênero FPS misturado com outros gêneros, como

RPGs, Open World, Aventura, entre outros.

Outro gênero destacado é do tipo *Real Time Strategy* (RTS), que pode ser definido como um subgênero do gênero de estratégia, onde o jogo não progride em turnos, mas sim em tempo real. Como o termo RTS foi criado recentemente, depois de muito refinamento e evolução no gênero *Strategy*, os jogos mais antigos categorizados como RTS não foram pensados para serem desse gênero. Isso significa que classificar jogos como sendo os primeiros do gênero se torna um trabalho um pouco difícil. Apesar de atualmente existirem bons títulos de jogos RTS que sejam *single-player*, os mais populares e mais jogados focam em partidas disputadas entre jogadores reais, de forma *multiplayer*. Um bom exemplo, que desde o seu começo teve *multiplayer* e que cresceu muito com o crescimento dos Esports, é o jogo Starcraft [19]. Outro gênero que nasceu a partir do gênero de RTS *multiplayer online* é o chamado MOBA (*Multiplayer Online Battle Arena*), uma modificação (Mod) feita por jogadores em um mapa para o jogo WarCraft 3: Reign Of Chaos [20] chamado *Defense of the Ancients* (DOTA). Esse Mod foi, então, transformado em um jogo completo e independente de outro pela empresa de desenvolvimento de jogos chamada Valve, com o nome de Dota 2, em 2013 [21]. O gênero MOBA tem uma relação muito próxima com Esports, tendo evoluído para o fim de prover grande competitividade entre os jogadores.

O gênero de *Role Playing Games* (RPG) nasceu muito antes dos jogos eletrônicos terem a perspectiva de abordá-lo, em que elementos de fantasia estavam sendo adicionados constantemente aos jogos de guerra de tabuleiro desde a década de 60. Em 1974, um grande título de RPG de mesa foi criado, o *Dungeons And Dragons* [22]. O gênero de mesa e o gênero nascente de jogos eletrônicos seguiram caminhos diferentes. Um título renomado que foi concebido nessa época deve ser mencionado, *Rogue* [23], lançado em 1980. Posteriormente, vários jogos com grande popularidade foram lançados como *Baldur's Gate* (1998)[24] e *Fallout* (1997)[25].

Por fim, temos os jogos de Corrida. Esse gênero foi criado no fim da década de 60, sendo um dos grandes títulos dessa época o jogo *Space Race* [26], da Atari, de 1973. O gênero de corrida cresceu muito nos primeiros anos, principalmente pelo fato de ser um gênero que condiz muito com a plataforma fliperama, que era a principal plataforma alvo na época. Com o avanço da tecnologia, os jogos de corrida começaram a ficar cada vez mais realistas, de forma que a simulação do funcionamento da física dos veículos se tornou cada vez mais próxima do que acontece no mundo real, resultando até na criação do sub-gênero de simuladores de corrida. Um exemplo de título desse gênero é o jogo *Project CARS* [27] 2015.

## 2.2 Ambientes Virtuais Distribuídos

Um Ambiente Virtual Distribuído é definido como uma tecnologia criada para a interação de diversos jogadores conectados em um mundo virtual 2D/3D pela internet. Esses ambientes têm o propósito de simular experiências imersivas, complexas e altamente interativas. Esses ambientes conectam um grande número de participantes, de diferentes locais físicos e de forma

a entregar experiências sensoriais (visuais e auditivas) [5]

Para os elementos dentro de um AVD serem entendidos como reais para o jogador, ou seja, para que haja uma boa imersão do jogador no ambiente, os diversos elementos presentes dentro do AVD precisam não somente parecer reais para que a interação do jogador com o mundo criado seja natural, mas também se mover corretamente, se comportar com credibilidade e se aproximar ao máximo da complexidade que o mundo real tem [6]. Os jogos digitais naturalmente tomam como base as definições dos AVDs, uma vez que implementam a maioria, se não todas, as suas características.

Dois grandes problemas presentes dentro dos AVDs são: a quantidade de informação crescente, que precisa ser compartilhada com cada cliente para que haja uma boa consistência entre as informações; e, a forma de gerenciamento dessas informações, que precisam ser compartilhadas considerando variáveis como consistência e responsividade [10].

A principal abordagem que lida com o primeiro problema citado, de quantidade de informação a ser compartilhada, propõe o conceito de áreas de interesse. Essa abordagem dita que cada cliente tem uma área de interesse, ou seja, uma área que define quais informações são realmente interessantes para o dado cliente, e que somente essas informações necessitam um alto nível de consistência [7]. Um exemplo da definição de uma área de interesse pode ser uma área ao redor do personagem do jogador até uma certa distância. Ou seja, a área interessante para esse jogador, em que há a necessidade de compartilhamento das informações com ele, são somente as informações próximas a ele até a distância predefinida.

A principal linha de pesquisas que lida com o segundo problema citado, do gerenciamento das informações compartilhadas entre os clientes, foca principalmente nas posições dos objetos que estão compartilhados. As informações das posições dos objetos de interesse, frequentemente, estão mudando e conseqüentemente podem estar inconsistentes. Essa inconsistência causa uma perda na QoS para o jogador, o que não é desejado [8].

O uso de ambas as abordagens concomitantemente é interessante, de forma que o uso de áreas de interesse, somado a técnicas preditivas, pode manter um nível de consistência elevada a objetos que são de fato considerados interessantes. Jaya *et al* [28] mostraram isso em sua proposta, onde houve a avaliação da junção da técnica preditiva DR, uma técnica muito usada na indústria, e a definição de áreas de interesse por auras ao entorno dos personagens dos jogadores.

Em um AVD, cada objeto que é compartilhado com outros clientes e o servidor pode ser chamado de entidade. Esses objetos, ou entidades, podem ser categorizados como locais ou remotos, dependendo de onde eles estão e a quem eles pertencem [29]. As entidades que estão na aplicação do jogador, e são consideradas como em posse desse jogador (o personagem desse jogador, como exemplo), são categorizadas como entidades locais, enquanto as outras entidades que não pertencem a esse jogador (os personagens dos outros jogadores, como exemplo) são consideradas entidades remotas. Esses objetos remotos têm o propósito de replicar o com-

portamento dos seus respectivos objetos locais que estão nas aplicações de outros jogadores [14].

Há diversas propostas na literatura que resolvem o problema de gerenciamento de informações. As principais características que influenciam tais propostas são a consistência relacionada à veracidade da informação nas entidades remotas, atraso na rede, responsividade, consumo de banda de rede, entre outras [9], [11] e [14]. Nas pesquisas dos autores abordadas neste trabalho, somente um autor considerou a presença de latência e a influência desta na técnica proposta.

### **2.3 As propostas da literatura**

As técnicas propostas na literatura podem ser categorizadas, de uma forma geral, entre aquelas que priorizam consistência das entidades remotas e aquelas que priorizam responsividade para o jogador em controle da entidade local [10].

Alguns gêneros de jogos têm maior sensibilidade para uma das categorias. Jogos de RTS, têm maior sensibilidade para consistência, enquanto jogos de FPS, têm maior sensibilidade para responsividade. Apesar de tal tendência, quando o jogo se torna um Esport, a demanda dos jogadores, por causa da competitividade do jogo, gera grande sensibilidade para ambas as características de consistência e responsividade, independente do gênero [30].

A consistência, dentro de jogos online, é definida pela veracidade da informação presente em objetos remotos, relacionada à verdadeira informação presente nos objetos locais. Essa característica é essencial para a validação de ações e para o processamento de eventos decisivos no servidor. Uma situação exemplo é quando o Jogador 1 (J1) atira e mata o Jogador 2 (J2) em um jogo FPS. Se há inconsistências entre informações compartilhadas entre os clientes, pode acontecer que o J2, que está morto nesse momento, não tenha essa informação atualizada em seu cliente, o que possibilita ele a atirar em J1. Isso resulta em um jogador morto atirando e possivelmente matando o jogador que o havia matado, o que quebra completamente a imersão do jogo e causa problemas na QoS. Gerenciar informações distribuídas, de forma consistente, é um dos principais desafios de um AVD [31].

### **2.4 Técnicas que priorizam a consistência**

A gestão das informações compartilhadas, de maneira consistente, é um dos principais objetivos para os autores que lidam com técnicas preditivas. Com esse foco, diversas abordagens priorizam a consistência da técnica, mesmo que em detrimento da responsividade [9].

Levine [32], com o intuito de propor um algoritmo anti-trapaça, propôs o protocolo *LockStep*. Segundo esse autor, uma maneira simples e comum de trapacear em jogos online é enviar mensagens de posições com tempos errados, de forma que o trapaceador ganhe algumas

centenas de milissegundos de vantagem. Em sua proposta, para impossibilitar esse caso, o protocolo *LockStep* faz com que todos os clientes e servidores somente avancem para o próximo quadro do jogo quando todos os clientes informarem e receberem as informações novas, que serão atualizadas de forma que todos avancem juntos para o próximo quadro. O ponto negativo desta técnica, apontado pelo próprio autor, é que necessariamente o atraso da responsividade para todos os jogadores se torna igual à maior latência<sup>1</sup> presente entre os jogadores e o servidor [32].

Seguindo a mesma linha da pesquisa de Levine, Lee [33] propôs o algoritmo chamado *Pipe-lined Lockstep*. A principal finalidade dessa proposta é manter a característica anti-trapaça da proposta de Levine, porém, melhorando o desempenho da técnica. Kharitonov [9] cita, porém, que o ponto negativo da técnica é que a técnica tem desvantagens de atraso e oscilações (*jitter*) na rede.

A técnica (DR) [34], por sua vez, torna-se proeminente por focar na responsividade mantendo bons níveis de consistência. Essa técnica será detalhada e aprofundada adiante na seção 2.5. Apesar de clássica, a técnica DR tem o seguinte problema, como descrito por Mauve [35]: ela consegue manter boa consistência e diminuir o custo de banda, enviando menos mensagens de atualização para os objetos remotos, porém, isso pode causar momentos nos quais a técnica DR considera que as informações estão consistentes mesmo estando erradas. Mauve [35], pensando nesse problema, propôs o conceito de Lag Local (*Local-lag*). Esse conceito tem como base a diminuição deliberada da responsividade para que haja a diminuição do número e duração de momentos de inconsistências. O autor discute que somente o *Local-lag* não consegue prevenir as ocasiões de inconsistências. Portanto, também propõe em seu trabalho o *time warp*, que corrige esses momentos restantes de inconsistências.

Zhang [36] propôs a técnica *Globally Synchronized Dead Reckoning with local lag* (GS-DR-LL), que junta a ideia de Mauve, de lag local, à técnica de Aggarwal [14], a *Globally Synchronizing Dead Reckoning* (GS-DR), que prioriza a responsividade. Com a sincronização dos *clocks* locais em conjunto com o lag local, há uma diminuição de inconsistências.

Os jogos do gênero FPS têm maior sensibilidade a latências [37]. Com esse argumento, Cronin [37] propôs a técnica *Trailing State Synchronization* (TSS). A técnica TSS é parecida com a técnica *time warp* de Mauve [35], considerando que é otimista, onde espera-se que a predição será correta e, caso haja um erro, haverá um retorno para o estado correto (*rollback*). De acordo com Kharitonov [9], o benefício da técnica TSS, em comparação à técnica *time warp*, é que ela não sofre com altos custos de memória e de processamento.

Apresentando outra abordagem, Shen [38] propôs a técnica *Progressive Slowdown*, que tem como base o conceito do método *Slowdown*. Esse conceito base define que um dado objeto remoto em movimento deve diminuir sua velocidade, de acordo com a latência presente no

<sup>1</sup> Latência ou *lag*, se refere ao atraso que afeta o envio de mensagens de um ponto a outro.

AVD, ao ser verificado que pode haver uma colisão no fim de sua trajetória, com o fim de que haja tempo suficiente para que a mensagem de atualização chegue. Shen comenta que para atrasos grandes na rede essa técnica não funciona. Para abrange tais latências, o autor propôs a técnica mencionada anteriormente *Progressive Slowdown*. Nesta nova proposta, o objeto remoto diminui sua velocidade progressivamente até o ponto de colisão, de forma que esteja lenta o suficiente antes da colisão para o recebimento da mensagem de atualização.

## 2.5 Técnicas que priorizam a responsividade

A responsividade do jogador é relativa ao atraso da reação do jogo com as entradas de dados que o jogador insere no jogo. Uma grande responsividade significa um menor atraso no processamento de entradas de dados, enquanto uma menor responsividade significa um maior atraso no processamento desses dados. Alguns gêneros de jogos têm necessidade de maior responsividade, como o gênero FPS [10]. Em tal gênero, principalmente quando o jogo ganha a característica de Esports, os desenvolvedores, tanto de softwares quanto de hardwares, vão a limites em prol da otimização a fim de entregar uma responsividade com o menor atraso possível. Um grande desafio de técnicas que têm como foco a responsividade, é manter bons níveis de consistência.

A principal técnica que foca a responsividade dos jogadores, e que é capaz de manter bons níveis de consistência, é a chamada Dead Reckoning (DR) [34]. O algoritmo foi inicialmente criado para a predição da posição atual de navios e aeronaves baseado nas informações iniciais de posição e movimento e o tempo passado. A técnica se tornou um modelo clássico para alta responsividade e consegue manter bons níveis de consistência [39].

A técnica DR tem a característica de ser preditiva e otimista, sendo usada com o propósito de diminuir o número de mensagens de atualizações necessárias para se manter uma boa consistência nos objetos remotos, apesar de suas limitações. Pensando em suas principais características, diversos autores propuseram melhorias sobre a técnica clássica, tanto para melhorar alguma característica quanto para abordar algum problema presente nela.

A primeira melhoria feita sobre ela foi a técnica proposta por Aggarwal *et al* [14]. Esta técnica propõe um modelo de baixa fidelidade, sendo classificado como um algoritmo otimista, em que o objeto remoto usa informações sobre velocidade de direção para predizer a localização do objeto local até que uma mensagem de atualização de posição chegue. A característica otimista significa que a predição espera estar correta, mesmo que isso cause problemas de inconsistência. Segundo Kharitonov [9], Aggarwal propõe a inserção de marcas temporais nas mensagens para aumentar a precisão da técnica sob atrasos na rede, sendo a única presente neste trabalho que considera a latência como um fator que influencia nessa precisão.

A Figura 1 ilustra a trajetória do objeto remoto usando a técnica DR para predizer a correta trajetória do objeto local. O DR alcança a proposta de diminuição no envio de mensagens

de atualização com o modelo preditivo, onde o objeto remoto utiliza tal modelo para predição e o objeto local precisa, então, somente manter uma taxa de atualização fixa suficiente, que é justificado pela ideia de que entidades raramente mudam suas direções e velocidades, para manter uma boa precisão do modelo no objeto remoto.

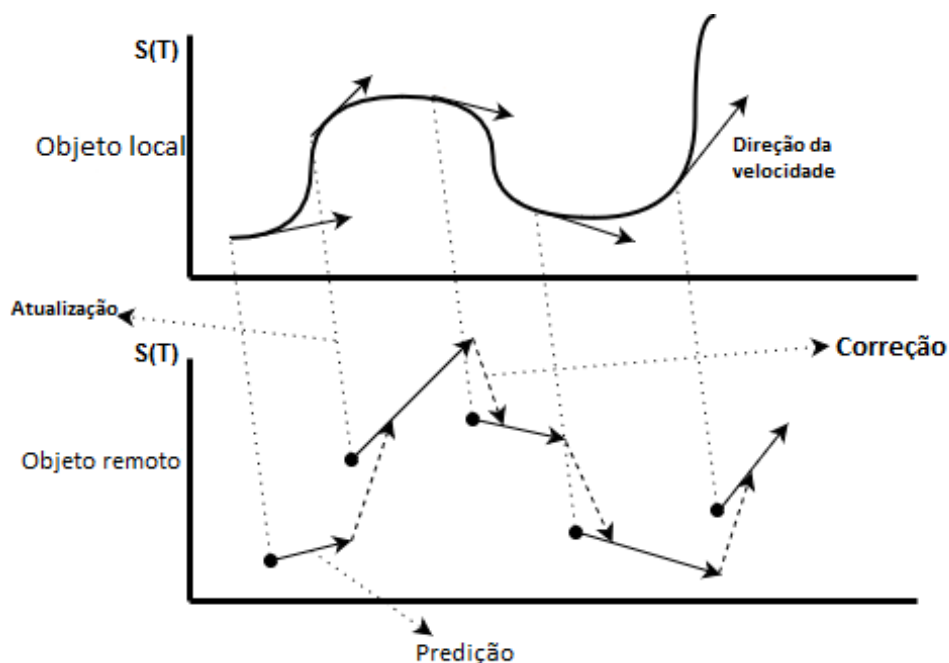


Figura 1 – Comportamento de um objeto local e seu respectivo objeto remoto usando a técnica DR para prever a trajetória correta, adaptado de Kharitonov [9].

Quando um erro na atual posição do objeto remoto é detectado, baseado na comparação com a posição correta adquirida pela mensagem de atualização, técnicas de convergência podem ser usadas para corrigir a trajetória errada para a trajetória correta do objeto local. Tais técnicas de convergência têm o propósito de ser boas respostas visuais para uma correção da trajetória errada para a trajetória correta, ao contrário de somente atualizar a posição do objeto remoto com a informação da posição correta. Tal solução simplista quebraria a imersão do jogo e daria uma sensação ruim ao jogador, de forma que causa problemas de QoS [8].

A principal solução para o problema de convergência é a interpolação entre a trajetória errada do objeto remoto e a trajetória correta do objeto local. Com isso, o objeto remoto suavemente corrige sua trajetória até a trajetória correta em uma quantidade  $T$  predeterminada de segundos.

A Figura 2 ilustra a técnica de convergência de interpolação aplicada no objeto remoto. O primeiro passo da interpolação é determinar qual será o tempo total de interpolação da trajetória incorreta até a trajetória correta (parâmetro  $T$ ). Após esse passo, o modelo preditivo é usado para prever a próxima posição, baseada nas variáveis e estado de movimento incorretos presentes no objeto remoto, que é representado pela seta verde *movimento remoto*. Antes de



atualizar a próxima posição, o modelo preditivo é usado para prever a próxima posição com base nas informações corretas de atualização que chegaram da entidade local, que é representado pela seta vermelha *movimento local*. Baseado no tempo que passou desde o começo do movimento de correção até o tempo T, a próxima posição real do objeto remoto é definida pela interpolação entre a posição da trajetória errada e a posição da trajetória correta, de forma que ao fim do tempo T, o objeto remoto esteja na trajetória correta [4].

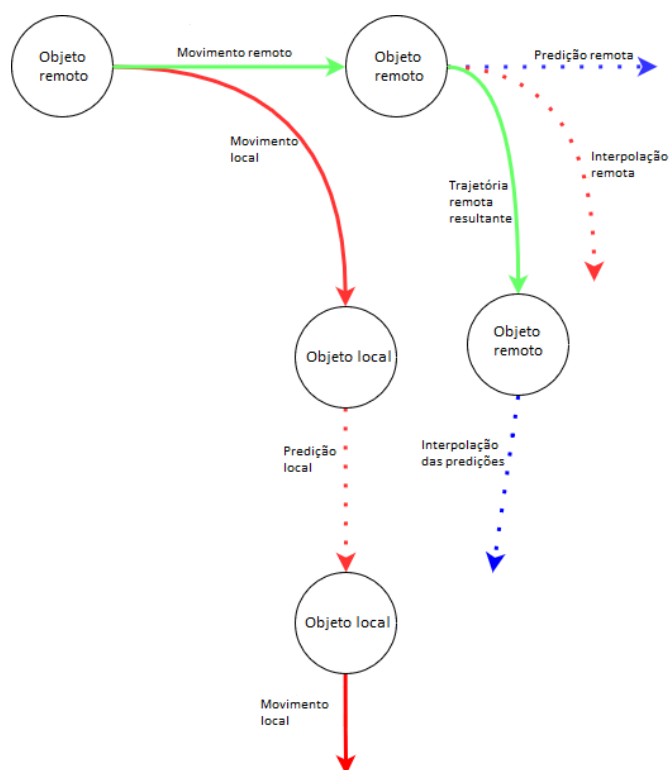


Figura 2 – Correção da trajetória do objeto remoto via interpolação.

Algumas características da técnica DR são mais modificadas e melhoradas nas propostas encontradas na literatura. Uma característica usualmente modificada, pelos autores abordados nesse trabalho, é a condição para se mandar mensagens preemptivas, quando presente. No modelo clássico a proposta é predeterminar uma taxa de atualização fixa, sem condição para envio preemptivo de mensagens. Outro modelo usado no DR é a simulação de um objeto com baixa fidelidade no lado local. A principal ideia é replicar o comportamento do objeto remoto no lado local, de forma que utiliza-se da comparação entre a posição do objeto local e a posição do objeto com baixa fidelidade, resultando no envio de uma mensagem de atualização baseada num limite de distância fixo [9]. O modelo mais utilizado é o primeiro.

Visando a característica de melhorar a condição para enviar mensagens preemptivas, Cai [40] propôs um algoritmo adaptativo chamado *Auto-Adaptive Dead Reckoning* (ADR). A principal ideia proposta pelo autor é diminuir a taxa de atualização com base na distância, dentro do ambiente virtual, entre as entidades remotas e a janela de visão do jogador local. A justificativa é que entidades remotas que estão com uma distância maior da janela de visão

do jogador podem ter uma maior inconsistência sem ferir QoS, permitindo também um menor consumo de banda de rede. Da mesma forma que entidades remotas com maior distância têm menos atualizações de posições, entidades remotas mais próximas da janela de visão do jogador precisam ter maior taxa de atualização para manter maior consistência. Em sua proposta, o autor não considerou a presença de latência e como esta influencia na performance do seu algoritmo.

Outra característica modificada na técnica DR são as expressões de predição. No algoritmo clássico, a expressão básica de velocidade linear constante é aplicada. Na literatura, diversos autores propuseram expressões mais complexas para aumentar a precisão das predições.

Lee *et al* [41], em conjunto com Cai *et al* [40], exploraram essa outra característica do algoritmo DR. A principal proposta foi implementar uma diferente expressão de predição para ser usada pelo anteriormente proposto algoritmo ADR. Enquanto a adaptabilidade diminui a taxa de atualização e o consumo de rede, diferentes expressões de predição propõem uma melhor predição com menos erros, aumentando a consistência da técnica. A proposta foca principalmente nos movimentos abruptos causados pelas correções previamente implementadas na técnica ADR. Novamente, os autores avaliaram o algoritmo proposto comparando sua precisão com a precisão de outros algoritmos que representavam o estado da arte, porém, sem avaliar a performance da técnica na presença de diferentes valores de latências.

Duncan *et al* [42] focaram nas características de adaptabilidade, propondo o algoritmo PreReckoning. Duncan é o primeiro a propor uma abordagem adaptativa que usa informações de detecção de movimento para adaptar a taxa de atualização. A técnica implementa essa abordagem checando a todo momento a aceleração do objeto local. Se uma mudança na aceleração é detectada, uma mensagem preemptiva é mandada para o objeto remoto, que por sua vez atualiza seus próprios valores. A principal ideia usada pelos autores na técnica é que, se uma mudança de aceleração ocorre no objeto local, o objeto remoto está fadado a ter uma predição errada, causando erros e inconsistências. Portanto, uma mensagem preemptiva é enviada para que a entidade remota tenha uma reação mais rápida às mudanças locais. A performance da técnica proposta foi avaliada somente em termos de precisão, sem a consideração de latência.

As duas características mais focadas por autores, apresentadas previamente, são inter-relacionadas, onde uma taxa de atualização baixa leva a um maior erro no modelo preditivo e uma taxa de atualização muito grande causa um uso ineficiente de banda de rede, apesar das maiores precisões [9].

Um algoritmo, com base no DR considerado ideal, precisa balancear a taxa de atualização sem consumir muita banda de rede e sem levar o modelo de predição a ter muitos erros. A técnica deveria manter bons níveis de consistência sem descartar a responsividade e precisão nas expressões. Kharitonov *et al* [9], baseado nas melhorias sobre a técnica DR achadas na literatura, e analisando seus pontos positivos e negativos, propuseram uma técnica que modifica tanto a condição para envio de mensagens preemptivas quanto as expressões preditivas.

Kharitonov chamou seu algoritmo de *Motion-Aware Adaptive Dead Reckoning* (MAADR). Essa técnica é considerada ciente de movimento, o que significa que ela utiliza um modelo mais elaborado para a predição da trajetória correta dos objetos locais em seus objetos remotos, de forma que esse mesmo modelo complexo é encontrado no objeto local, o qual é usado para identificar mudança de padrões de movimento. O modelo de predição usa informações de velocidade, aceleração e movimentos circulares para fazer suas predições. A técnica é adaptativa porque ao usar tal modelo preditivo no objeto local, e ao identificar mudanças nos padrões de movimento, uma mensagem de atualização de posição preemptiva é enviada para minimizar erros causados por tais mudanças.

A Figura 3 mostra a relação entre a entidade local e a entidade remota, onde ambos implementam a técnica MAADR [9]. Há dois tipos de cenários em que uma mensagem de atualização dos padrões de movimento é enviada. No primeiro cenário, a mudança de padrões de movimento é identificada pela entidade local e uma mensagem de mudança de padrão de movimento é enviada de forma preemptiva. A entidade remota, recebendo esse tipo de mensagem, atualiza suas informações de padrões de movimento e faz suas predições adequadamente. No segundo cenário, o tempo limite para envio de uma mensagem de atualização é atingido e uma mensagem de atualização de posição é enviada. A entidade remota prediz a atual posição para checar se está correta e continua sua execução de acordo com o resultado. Caso a predição indique que sua trajetória atual está errada, a entidade inicia uma rotina de correção e, caso esteja correta, somente continua suas predições normalmente.

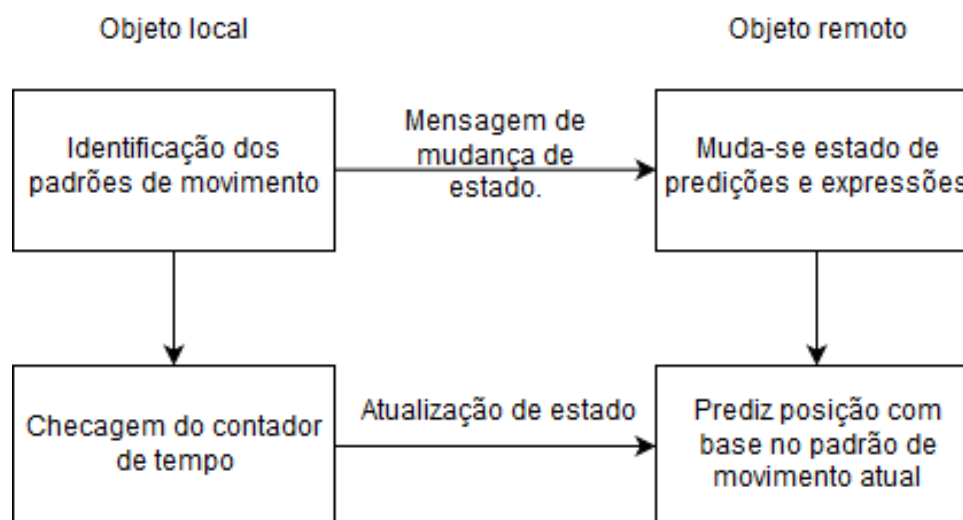


Figura 3 – Relações entre objeto local e objeto remoto dentro da técnica MAADR, adaptado de Kharitonov [9].

Kharitonov conclui seu trabalho com os resultados da sua técnica proposta. Tais resultados mostram que a técnica MAADR tem uma melhor performance quando comparada com a técnica DR. O autor avaliou diversos valores de taxas de atualização, com objetivo de mensurar o melhor balanço entre consumo de banda de rede e precisão. A conclusão para a melhor taxa

de atualização para técnica MAADR foi de 2 Hz, com 0,5 segundo de limite de taxa de atualização. Em sua análise, porém, o autor não considera a latência como variável para ser avaliada, que é um fator chave para a QoS [8].

### 3 MATERIAIS E MÉTODOS

Ambientes virtuais distribuídos demandam grande desempenho dos processadores principais, RAM e GPU, especialmente em jogos. Além de tal processamento, o componente de rede tem que compartilhar as informações entre clientes e servidor de forma consistente e responsiva. Para realizar a avaliação proposta neste trabalho, e por ser uma ferramenta profissional usada por vários desenvolvedores de jogos, animações altamente gráficas, entre outros produtos de mídia, o motor gráfico Unity3D [43] foi escolhido. O motor gráfico citado implementa o *framework* de rede, chamado UNet [44], que tem a capacidade de atrasar mensagens recebidas pelo jogo na camada de aplicação.

Colocando em prática as técnicas MAADR [9] e DR [14], 6 cenários foram criados para melhor avaliar as suas performances sob médias e altas latências. Nesses cenários, o erro para cada técnica foi medido usando a distância euclidiana entre a atual posição do objeto remoto e a correta posição do objeto local em cada quadro fixo (*Fixed Frame*). Um quadro fixo, dentro do Unity3D, acontece a cada 0,02 segundos e é usado para processar a física em todos os objetos dentro do ambiente virtual.

#### 3.1 Arquitetura de rede

Como mencionado anteriormente, o *framework* de rede do Unity3D, chamado UNet [44], foi usado para modelar as conexões dos jogadores dentro do jogo com uma arquitetura servidor-cliente. Nesta arquitetura, os clientes são responsáveis por mandar as ações dos seus personagens para o servidor e por renderizar o ambiente virtual com base no estado atual do mesmo. O servidor, por sua vez, é responsável por calcular o resultado de toda física relevante aos jogadores e os resultados das ações dos jogadores, para assim, enviar a informação atualizada para todos os clientes.

A principal forma de abordar a presença de latência em ambientes virtuais, ao compartilhar informações que precisam de consistência e são relativamente velhas (quanto maior a latência, mais velha a mensagem será quando finalmente chegar nas entidades remotas), é colocando marcas temporais (*timestamps*) nas mensagens, com o tempo exato atual no objeto local, de forma que no objeto remoto haja a predição da posição correta atual a partir da posição correta velha, como proposto por Aggarwal [14]. Essa abordagem é usada nos algoritmos MAADR e DR para a avaliação presente neste trabalho.

Ao tentar criar a melhor técnica preditiva, deve existir um balanço entre a precisão e o consumo de banda de rede. Quanto maior for a taxa de envio de atualizações de mensagens, maior será a precisão e maior será o consumo de rede, o que pode fazer a abordagem se tornar inviável para aplicação em casos reais, por causa do alto consumo de rede. Partindo disso, a

avaliação proposta neste trabalho tem como objetivo mensurar a capacidade das técnicas DR e MAADR em manter a precisão de suas previsões sob condições de altas latências. As principais características variadas são a latência e os padrões de movimento, ou seja, as expressões da previsão. Já que a condição para envio das mensagens de atualização em nenhuma das técnicas considera o presente atraso na rede para enviar mensagens preemptivas, o consumo de rede é mantido constante para cada par de cenário e algoritmo ao longo de todas as latências testadas. Portanto, a avaliação é focada na precisão da previsão das trajetórias de cada técnica.

### 3.2 Implementações do DR e do MAADR

O uso da técnica DR leva em consideração dois problemas: a escolha da forma de atualização das entidades remotas, e o modelo preditivo com suas expressões usado nos objetos. Ambos os problemas estão inter-relacionados.

Para a técnica DR a taxa de atualização é fixa em 2 Hz (0,5 segundo de limite), de forma que não há condições preemptivas que a modificam. A única expressão preditiva presente nessa técnica é a expressão de velocidade linear constante. A rotina de correção usada com a técnica DR é a interpolação da trajetória errada em direção à trajetória correta em um determinado tempo  $T$ .

O algoritmo 1 mostra a técnica DR presente no objeto local, onde  $Ol$  representa as informações de movimento do objeto local. A técnica DR presente no objeto local verifica se o tempo limite envia uma mensagem de atualização caso o tempo limite tenha sido alcançado.

---

#### Algoritmo 1: DR LOCAL

---

**Entrada:**  $Ol$

```

1 início
2   se  $ContadorTempo \leq TempoLimite$  então
3     EnviarMensagemDeAtualização( $Ol$ )
4     ContadorTempo = 0
5 fim
```

---

O algoritmo 2 mostra a implementação da técnica DR presente em objetos remotos, onde  $Ol$ ,  $Or$  e  $T$  representam as informações de movimento do objeto local, as informações de movimento do objeto remoto e o tempo determinado usado na rotina de correção, respectivamente. Primeiro verifica-se se houve um recebimento de uma mensagem de atualização. Caso sim, checka-se se a trajetória atual do objeto remoto está correta usando a fórmula de previsão da técnica DR. Se a trajetória atual do objeto remoto está errada, inicia-se uma rotina de correção que interpola a trajetória incorreta em direção à trajetória correta. Se não houve o recebimento de mensagem de atualização, ou a trajetória do objeto remoto está correta, continua-se a mover o objeto remoto de acordo com as previsões da técnica DR. O algoritmo 2 é executado em cada

quadro fixo (0,02 segundo).

---

**Algoritmo 2: DR REMOTO**

---

**Entrada:**  $Ol, Or, T$

```

1 início
2   se MensagemRecebida( $Ol$ ) então
3      $Pc = \text{PreverPosiçãoCorretaAtual}(Ol)$ 
4      $Traj\acute{e}t\acute{o}riaErrada = \text{ChecaErroTrajet\acute{o}ria}(Pc, Or)$ 
5     se  $Traj\acute{e}t\acute{o}riaErrada == Verdadeiro$  então
6       IniciarRotinaDeCorreção( $Ol, Or, T$ )
7     senão
8        $P = \text{PreverPr\acute{o}ximaPosi\c{c}\~{a}oDR}(Or)$ 
9       AplicarNovaPosiç\~{a}o( $P$ )
10    fim
11  senão
12     $P = \text{PreverPr\acute{o}ximaPosi\c{c}\~{a}oDR}(Or)$ 
13    AplicarNovaPosiç\~{a}o( $P$ )
14  fim
15 fim
```

---

A taxa de envio de mensagens de atualização da técnica MAADR não é somente definida pela taxa fixa de atualização, mas também pelo envio preemptivo. Tal envio é condicionado pelo modelo de movimento complexo, que também é adotado como modelo no objeto local que identifica mudanças significativas e envia mensagens preemptivas. No modelo de movimento são definidos estados de padrões de movimento, em que cada estado de padrão de movimento tem sua respectiva expressão de movimento.

Como a taxa de atualização de 2 Hz é a taxa com um bom balanço de custo de banda de rede e consistência, encontrada por Kharitonov [9], este é o valor usado na implementação de ambas as técnicas. A técnica de convergência também é a interpolação das trajetórias.

O algoritmo 3 define o comportamento da técnica MAADR presente nos objetos remotos, onde  $Ol$ ,  $Or$  e  $T$  são as informações de movimento do objeto local, as informações de movimento do objeto remoto e o tempo predefinido da rotina de correção, respectivamente. A ideia geral do algoritmo é bem parecida com a do DR remoto, de forma que as grandes diferenças estão na atualização dos padrões de movimentos, onde atualiza-se o tipo de movimento que está sendo usado nas predições e as predições em si, dentro da função de predição

*PreverProximaPosicaoMAADR()*.

---

**Algoritmo 3: MAADR REMOTO**

---

**Entrada:** *Ol, Or, T*

```

1 início
2   se MensagemRecebida(Ol) então
3     AtualizarPadrõesDeMovimentos(Ol)
4     Pc = PreverPosiçãoCorretaAtual(Ol)
5     TrajetóriaErrada = ChecaErroTrajetória(Ol, Or)
6     se TrajetóriaErrada == Verdadeiro então
7       IniciarRotinaDeCorreção(Ol, Or, T)
8     senão
9       P = PreverPróximaPosiçãoMAADR(Or)
10      AplicarNovaPosição(P)
11    fim
12  senão
13    P = PreverPróximaPosiçãoMAADR(Or)
14    AplicarNovaPosição(P)
15  fim
16 fim
```

---

O algoritmo 4 define o comportamento da técnica MAADR no objeto local, onde *Ol* e *PadroesVelhos* representam as informações do objeto local e os padrões de movimento do quadro anterior, respectivamente. O começo do algoritmo mostrado ilustra a identificação de uma mudança nos padrões de movimento do objeto local no momento atual, comparada com os padrões de movimento do quadro anterior, de forma que se há uma mudança significativa, há um envio preemptivo.

---

**Algoritmo 4: MAADR LOCAL**

---

**Entrada:** *Ol, PadroesVelhos*

```

1 início
2   PadroesDeMovimento = LerPadrõesDeMovimento0(Ol)
3   se PadroesVehos != PadroesDeMovimento então
4     EnviarMensagemDeAtualização(Ol)
5     ContadorTempo = 0
6   senão se ContadorTempo <= TempoLimite então
7     EnviarMensagemDeAtualização(Ol)
8     ContadorTempo = 0
9 fim
```

---

No algoritmo MAADR, três padrões de movimento são definidos. O primeiro é o movimento estacionário, que identifica uma ausência de movimento e possibilita o envio de mensa-



gens menores. O segundo é o movimento retilíneo uniforme, onde há a presença de velocidade constante. E o terceiro é o movimento uniformemente acelerado, onde há a presença de uma aceleração constante. Para melhor controle do movimento dos objetos, os padrões de movimento foram divididos em dois módulos: o primeiro é o chamado *Linear Motion Kernel* (LMK) e o segundo módulo é o *Angular Motion Kernel* (AMK). Ambos usam as mesmas expressões de movimento, porém, com a mudança de variáveis lineares para angulares. A predição com dois módulos é a mesma que com um módulo, ou seja, o uso de dois módulos no algoritmo não interfere na predição nem nos resultados do experimento feito. O único benefício da divisão dos padrões de movimento em dois é a maior facilidade do controle das variáveis.

A Figura 4 mostra dois módulos de movimento usados na implementação para a avaliação da técnica MAADR.

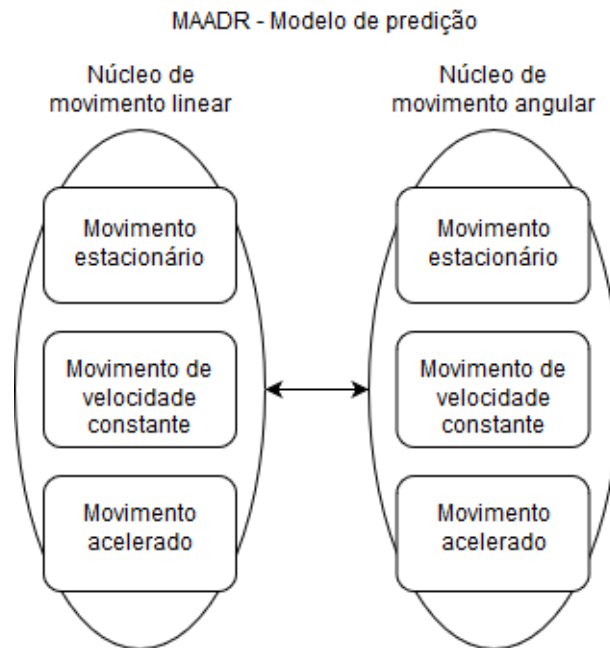


Figura 4 – Padrões de movimento implementados em dois módulos, adaptado de Kharitonov [9] para controle facilitado.

Para cada padrão de movimento, há uma expressão relativa a ele. As equações 3.1, 3.2 e 3.3 mostram, respectivamente, as expressões para os seguintes padrões de movimento: movimento estacionário, movimento retilíneo uniforme e movimento uniformemente acelerado.

$$S_f = S_i, \quad (3.1)$$

em que  $S_f$  é a posição final e  $S_i$  é a posição inicial.

$$S = S_0 + \Delta t \times v, \quad (3.2)$$

em que  $S$  é a posição final,  $S_0$  é a posição inicial,  $\Delta t$  é a variação de tempo e  $v$  é a velocidade constante.

$$S = S_0 + \Delta t \times v_0 + \frac{a \times \Delta t^2}{2}, \quad (3.3)$$

em que  $S$  é a posição final,  $S_0$  é a posição inicial,  $\Delta t$  variação de tempo,  $v_0$  é a velocidade inicial e  $a$  é a aceleração constante.

No movimento em que o objeto remoto recebe uma mensagem de atualização e verifica que um erro ocorreu, a rotina de correção é iniciada para interpolar a trajetória errada que o objeto remoto está percorrendo com a trajetória correta baseada na predição a partir das informações corretas e trajetória correta do objeto local. As equações 3.4, 3.5 e 3.6 são usadas na rotina de convergência com interpolação.

$$S_j = S_{0j} + \Delta t \times v_0 + \frac{a \times \Delta t^2}{2}, \quad (3.4)$$

em que  $S_j$  é a posição correta final do objeto local,  $S_{0j}$  é a posição correta inicial do objeto local,  $\Delta t$  é a variação de tempo,  $v_0$  é a velocidade inicial do objeto local  $a$  é a aceleração do objeto local.

$$S_f = S_{0f} + \Delta t \times v_0 + \frac{a \times \Delta t^2}{2}, \quad (3.5)$$

em que  $S_f$  é a posição final errada do objeto remoto,  $S_{0f}$  é a posição inicial errada do objeto remoto,  $\Delta t$  é a variação do tempo,  $v_0$  é a velocidade inicial e  $a$  é a aceleração presente.

$$I_p = S_f + (S_j - S_f) \times t_i, \quad (3.6)$$

em que  $I_p$  é a posição interpolada final,  $S_f$  e  $S_j$  são as posições errada e correta finais, respectivamente, para as equações 3.5 e 3.4 e  $t_i$  representa a proporção da trajetória errada para a correta, variando de 0 a 1, de forma que quando  $t_i$  chega ao valor 1 a trajetória do objeto remoto está completamente interpolada.

## 4 ANÁLISE E RESULTADOS

Para a avaliação das técnicas abordadas, cinco cenários foram propostos. Os cenários foram pensados para avaliar ambas as técnicas, usando desde padrões de movimento simples até complexos. Para se analisar a significância das comparações feitas entre as técnicas foram usadas três análises estatísticas.

### 4.1 Considerações gerais da análise dos resultados

Para manter consistência e precisão entre as avaliações, um sistema que lê as entradas de dados dos jogadores de um arquivo de texto foi implementado. Todos os cenários têm 4 segundos de duração e mantêm uma velocidade linear com magnitude que varia de 4 a 5 metros por segundo, a fim de não causar grandes variações nos erros por conta da variável de velocidade linear. O primeiro cenário tem uma média de velocidade linear de 4,25 metros por segundo, enquanto os demais cenários têm uma média de 4,5 metros por segundo.

O primeiro cenário é uma simples linha reta, mantendo uma velocidade linear constante nos dois primeiros segundos e variada nos dois últimos. O segundo cenário avalia a presença de uma velocidade linear e angular constantes, resultando em um movimento circular. O terceiro cenário aumenta o estresse nas técnicas por ter uma aceleração linear constante que alterna entre dois valores e uma velocidade angular constante, criando um movimento circular variado. O quarto cenário estressa ambos os algoritmos com uma aceleração linear constante que alterna entre dois valores em paralelo com mudanças abruptas na velocidade angular, resultando em um movimento de zigue-zague. E, por fim, o quinto cenário propõe um movimento pseudo aleatório, resultando em um caso com grande número de mudanças abruptas de movimento.

A taxa de atualização é estabelecida como 2 Hz, como descrito anteriormente. O intervalo entre 0 e 300 de latência simulada, em milissegundos, foi escolhido para abordar tanto a mínima latência simulada possível quanto o valor de 300 milissegundos que representa uma alta latência, em que a QoS fica precária e a perda de jogadores é iminente [45]. Os seguintes valores de latência simulada foram usados para as avaliações em cada cenário: 0, 50, 100, 150, 200, 250 e 300 milissegundos. O intervalo fixo de 50 milissegundos possibilita uma análise profunda das consequências da presença de latência, sem uma inundação de informações redundantes.

Na proposta original de Kharitonov, não houve uma análise estatística para mostrar se há uma diferença significativa entre as amostras da técnica MAADR e a técnica comparada originalmente, mesmo sem considerar a presença de latência. No presente trabalho há a avaliação de duas técnicas (MAADR e DR), sob 7 diferentes valores de latências (0, 50, 100, 150, 200, 250 e 300) em 5 cenários diferentes (linha reta, movimento circular, movimento circular variado, zigue-zague e movimento aleatório), em que técnicas, latências e cenários podem ser

interpretados como fatores em uma análise da variância no esquema fatorial [46]. Ao realizar tal análise, porém, identifica-se que os erros do modelo não cumprem o pré-requisito de terem uma distribuição normal.

Considerando essa anormalidade, testes não paramétricos foram escolhidos para avaliar a significância da diferença entre as técnicas para cada valor de latência, em cada cenário, e a significância do crescimento do erro da técnica DR em relação à técnica MAADR. Para identificar se há uma diferença significativa entre os conjuntos de amostras MAADR e DR para cada valor de latência, o teste U de Mann-Whitney (MW) [47] foi usado. O MW é um teste não paramétrico, onde não há a suposição inicial de que as amostras de dados têm uma distribuição normal, possibilitando verificar se a diferença entre as medianas de dois conjuntos de amostras é significativa.

Dada a significância da diferença entre os erros da técnica MAADR com a técnica DR, temos que um dos objetivos desse trabalho é identificar e analisar a diferença entre as duas técnicas, para cada cenário, conforme a latência aumenta. Na implementação dessa avaliação, para cada cenário e para cada valor de latência, foi gerado um conjunto de amostras novas contendo a diferença dos valores dos erros da técnica DR com os erros da técnica MAADR. Para identificar se há diferença significativa entre os conjuntos das diferenças entre as técnicas, o teste Kruskal-Wallis (KW) foi usado [48].

O KW é um teste não paramétrico, no qual não há a suposição inicial de que as amostras dos dados têm uma distribuição normal, e é aplicável a 3 ou mais conjuntos de amostras. A hipótese nula, para o teste KW, diz que todos os conjuntos de amostras testados têm a mesma distribuição, de forma que a hipótese alternativa diz que pelo menos um dos conjuntos de amostras tem uma distribuição diferente de pelo menos um outro conjunto de amostras, possibilitando concluir que há uma diferença significativa entre os conjuntos avaliados. A análise estatística KW identifica que pelo menos um conjunto difere significativamente de outro, porém, não expõe a relação entre os conjuntos. A análise de Dunn [49] então, possibilita analisar se a diferença entre os conjuntos é significativa, par a par, utilizando os mesmos ranques que são usados na análise KW, sendo uma análise *post-hoc* (realizado posteriormente) comum ao KW. Portanto, para os cenários onde a conclusão da técnica KW foi de rejeitar a hipótese nula, a análise de Dunn foi usada. Em todos os testes estatísticos houve o uso do valor de 95% de confiança e conseqüentemente o valor de 0,05 de significância. Para implementar e executar os testes estatísticos mencionados, a linguagem e ambiente de desenvolvimento R foi usada [50].

## 4.2 Cenário 1: Linha com Aceleração Constante

O primeiro cenário é o mais simples, mas bem comum de se encontrar em diversos gêneros de jogos. É composto de um período de velocidade linear constante, de 0 até 2 segundos e, então, seguido por um período com aceleração linear constante, primeiramente com o valor

de  $1 \text{ m/s}^2$  de aceleração do 2o segundo ao 3o segundo, seguido por um período de aceleração linear constante com o valor de  $-1 \text{ m/s}^2$ . Isso resulta em um movimento com média de  $4,25 \text{ m/s}$ , variando entre  $4$  e  $5 \text{ m/s}$  de velocidade linear e sendo o único cenário com média de velocidade linear diferente de  $4,5 \text{ m/s}$ . Esse padrão de movimento pode ser encontrado não somente em jogos de corrida, onde o veículo de um jogador deve se mover em uma linha reta, com ou sem aceleração, mas também em gêneros como *First Person Shooter* (FPS), onde o jogador deve se mover em uma linha reta para uma localização desejada. No gênero de jogos *Real Time Strategy* (RTS) também é encontrado esse padrão de movimento, no qual o jogador envia sua unidade para uma posição que somente precisa de um movimento linear para ser alcançado. A principal proposta desse cenário é avaliar não somente a habilidade de ambas as técnicas serem precisas neste caso simples, mas também medir o quanto a precisão piora com a presença de latência mesmo que seja um cenário simples. Esse caso foi inspirado no primeiro caso de movimento linear usado na avaliação de Aggarwal [14].

A Figura 5 ilustra a trajetória feita por um objeto remoto sem latência simulada. O movimento é vertical, de baixo para cima. Há um total de duas marcas na figura que representam as mudanças de padrões de movimento, seja nos módulos de movimento linear ou de movimento angular. A primeira marca em azul representa a velocidade inicial em  $4 \text{ m/s}$ , de forma que as seguintes marcas em verde representam a aceleração linear com os valores de  $1$  e  $-1 \text{ m/s}$  estando presentes, respectivamente.



Figura 5 – Cenário 1: movimento linear simples.

A Figura 6 mostra o gráfico das velocidades para o cenário 1. A velocidade angular é mantida em  $0 \text{ rad/s}$  por toda o cenário, enquanto a velocidade linear tem um período de aceleração como descrito anteriormente.

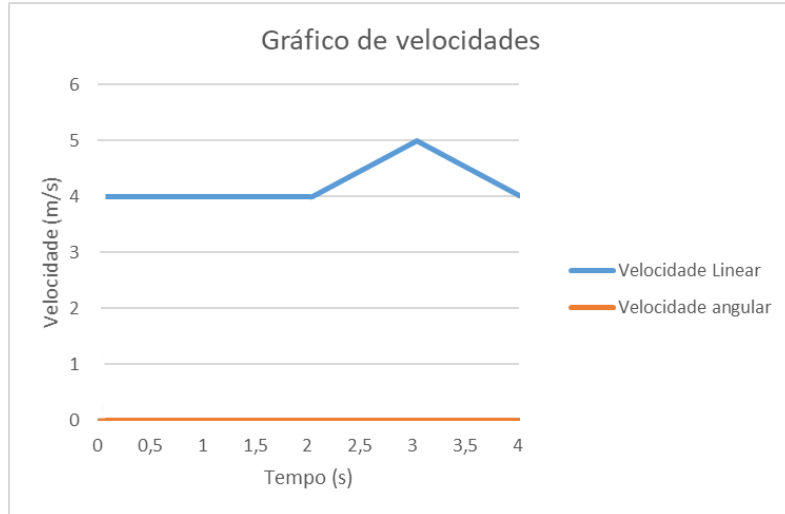


Figura 6 – Perfis de velocidade para o cenário 1: Movimento linear

A Figura 7 mostra a média dos erros de cada técnica para cada valor de latência no cenário 1. Essa figura mostra que o aumento do erro de ambas as técnicas conforme a latência aumenta, no qual o erro da técnica DR aumenta com maior magnitude do que a técnica MAADR. A área da técnica MAADR se mantém a todo momento menor do que a técnica DR, o que mostra que a técnica MAADR resulta em um menor erro.

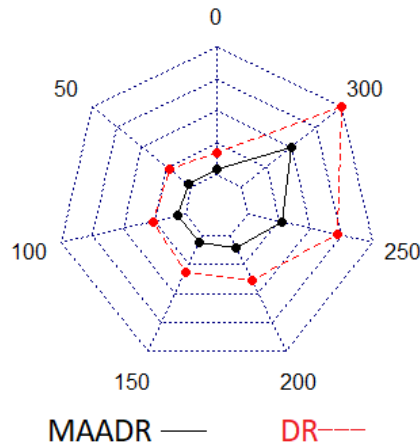


Figura 7 – Gráfico de radar com a média dos erros de ambas as técnicas para cada valor de latência para o cenário 1.

A Figura 8 ilustra um gráfico com as médias dos erros e desvios padrões para cada técnica ao longo das latências avaliadas presentes no eixo X. As marcações dos desvios padrões estão presentes somente para cima ou para baixo, para que esses traços não se sobreponham quando há um desvio padrão alto para ambas as técnicas.

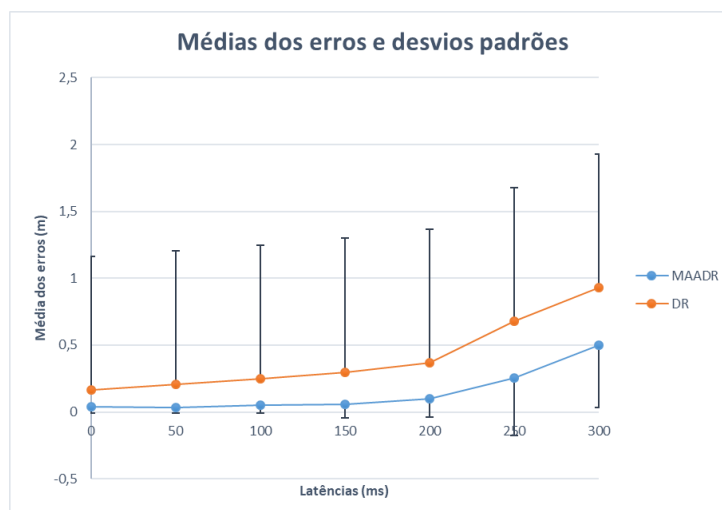


Figura 8 – Cenário 1: Médias dos erros e desvios padrões ao longo das latências.

A visualização da Figura 8 mostra indicativos de que a técnica DR atinge ambos erros e desvios padrões significativamente maiores do que a técnica MAADR, de forma que tem um aumento maior conforme a latência aumenta, mesmo que em um cenário simples. A técnica MW [47] então foi usada para analisar a diferença da técnica DR e MAADR para cada valor de latência.

A tabela 1 mostra os p-valores do teste de Mann-Whitney para cada valor de latência.

Tabela 1 – P-Valores do teste MW entre as duas técnicas para cada latência no cenário 1.

	0 ms	50 ms	100 ms	150 ms	200 ms	250 ms	300 ms
P-Valor	0,2299	7,199e-05	0,136	1,12e-05	4,97e-08	3,255e-10	3,263e-11

Temos que para os valores de latência simulada 0 e 100 ms o p-valor na tabela 1, de 0,2299 e 0,136 respectivamente, aceita-se a hipótese nula, no qual os dois conjuntos dos erros da técnica MAADR e DR são estatisticamente iguais. Já os p-valores para as latências simuladas de 50, 150, 200, 250 e 300 ms são menores que o valor crítico de 0,05, portanto, a hipótese nula é rejeitada e temos que há uma diferença significativa entre os conjuntos avaliados.

A igualdade estatística entre os conjuntos dos erros de ambas as técnicas com 0 ms de latência simulada para este cenário era esperado, considerando que como é simples, a variação do erro não é suficiente para causar uma diferença significativa. Já em 100 ms, há um resultado inesperado, principalmente porque com 50 ms de latência, a diferença é significativa. Isso pode ser causado pelo tipo de movimento e os momentos exatos onde há mudanças nos padrões de movimento desse cenário. Porém, isso mostra indícios de que, para movimentos muito simples, não há uma diferença significativa entre ambas as técnicas sob latências.

Conjuntos das diferenças dos erros da técnica DR e os erros da técnica MAADR foram então criados, lat0 até lat300, variando de 50 em 50, relativo a cada valor de latência. Tais

conjuntos foram usados para analisar se há um aumento significativo entre os erros das duas técnicas conforme aumenta a latência, inicialmente com o teste KW [48]. Com o p-valor igual a  $3,424e-14$  para o teste KW, temos que a hipótese nula é rejeitada, possibilitando concluir que há pelo menos um conjunto avaliado que é significativamente diferente de algum outro. Para se avaliar qual ou quais conjuntos têm uma diferença significativa entre si, foi aplicado o teste estatístico de Dunn.

A Figura 9 mostra as médias de cada conjunto de diferenças e os grupos gerados a partir da análise estatística de Dunn, onde as diferenças entre cada par de grupos foram usadas para gerar subgrupos de equivalência. Os conjuntos lats são identificados na figura pelas suas respectivas latências, no eixo X. Ao todo há cinco subgrupos formados pela análise estatística: 'a', 'b', 'bc', 'cd' e 'd'. Se um conjunto está presente em um grupo 'a', ele é significativamente diferente dos outros identificados com uma letra diferente, ou que não contenham a letra que o identifica. Isso significa que o conjunto lat300 deste cenário é estatisticamente igual ao conjunto lat250 e diferente aos demais conjuntos.

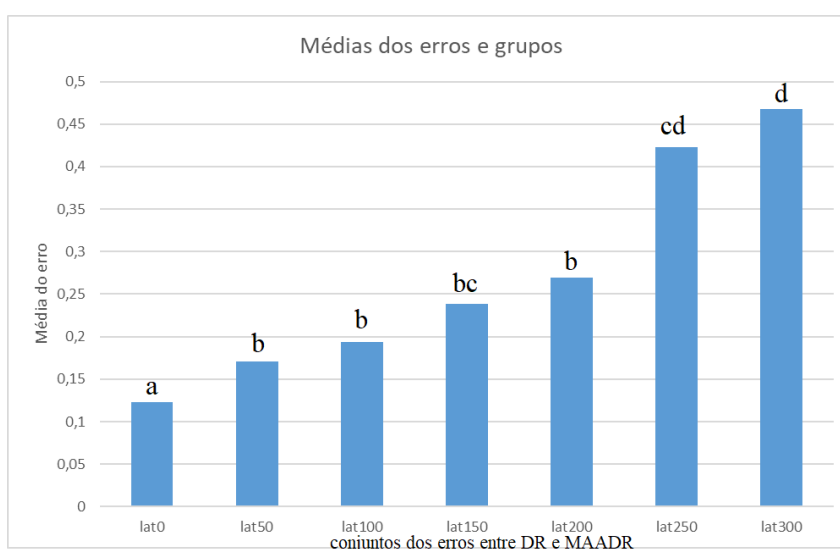


Figura 9 – Médias das diferenças dos erros e grupos para o cenário 1.

O conjunto lat200 é significativamente diferente dos conjuntos lat0, lat250, lat300, porém, como a letra que representa o grupo do conjunto lat200 é 'b', o conjunto lat200 não é significativamente diferente dos conjuntos de latência 150, 100 e 50. Da mesma forma, como a letra que identifica o conjunto de latência 300 é 'd', e está presente no nome 'cd', temos que a latência de 300 ms simulados não é significativamente diferente das latências de 0, 50, 100, 150 e 200.

A análise de Dunn, em conjunto com a visualização da Figura 8, mostra que há um aumento significativo do erro entre as técnicas DR e MAADR quando a latência aumenta de 0 ms para 50 ms de latência simulada, porém de 50 até 200 ms não há aumento significativo do erro. Somente com uma latência de 250 temos uma diferença no erro significativamente



diferente do intervalo de latência entre 50 e 200. A partir de 250 ms de latência, a análise de Dunn identifica uma diferença significativa no aumento da diferença do erro, ou seja, houve um crescimento maior do erro para a técnica DR do que para a técnica MAADR, que pode ser visualizado também na Figura 8.

Para o caso do cenário 1, que é mais simples, o aumento na latência precisa ser alto para resultar em uma diferença significativa na precisão das técnicas, porém, mostra que existe uma diferença entre as duas técnicas em termos de crescimento dos seus respectivos erros conforme a latência aumenta.

### 4.3 Cenário 2: Movimento Circular

O cenário 2 foi feito para avaliar a presença da velocidade linear e angular paralelamente, representando um movimento circular. Esse começa com a presença de uma velocidade linear de  $4,5m/s$  e uma velocidade angular de  $3,14 rad/s$ , ambos mantidos até o fim do cenário. O principal gênero de jogos que contempla esse tipo de padrão de movimento são os de corrida, onde em diversas curvas esse padrão de movimento pode ser encontrado. Em outros gêneros esse movimento pode aparecer, porém, menos frequentemente. Esse cenário foi proposto para avaliar ambas as técnicas sob caminhos circulares fechados, como o primeiro cenário testado por Kharitonov [9], e a influência da latência sobre as técnicas nesse tipo de comportamento.

A Figura 10 ilustra a trajetória feita pelo objeto remoto, seguindo de baixo para cima. Há duas marcas que representam mudanças nos padrões de movimento do módulo linear ou do módulo angular, assim como no cenário anterior. As marcas se encontram uma sobre a outra porque representam mudanças nos padrões de movimento que acontecem no mesmo instante. Nesse instante inicial, a velocidade linear tem magnitude de  $4,5 m/s$  e a velocidade angular tem magnitude de  $3,14 rad/s$ .

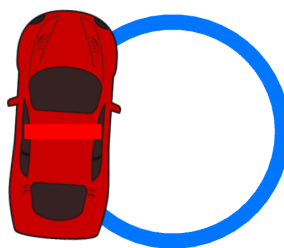


Figura 10 – Cenário 2: Movimento Circular.

A Figura 11 mostra o gráfico de velocidades para o cenário 2, no qual há uma velocidade linear e angular inicial que se mantém até o fim.

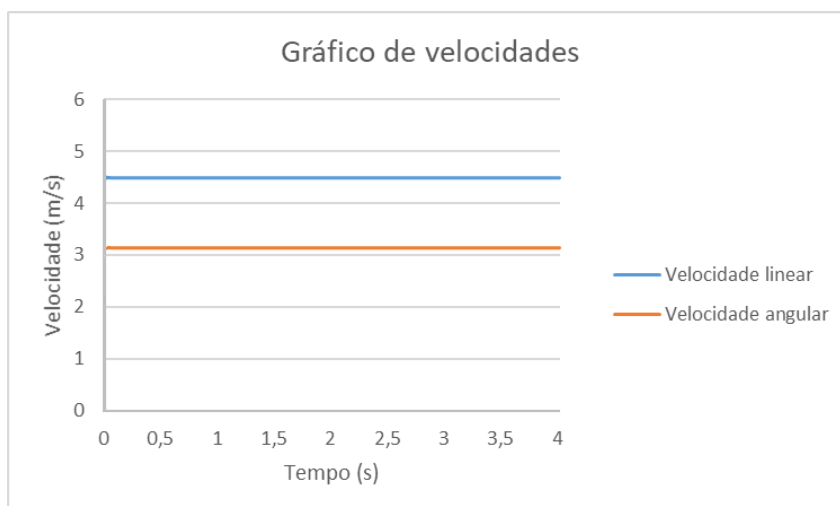


Figura 11 – Perfis de velocidade para o cenário 2: Movimento Circular.

A Figura 12 mostra um gráfico de radar com a média dos erros de cada técnica sob cada nível de latência para o cenário 2. Observa-se que o cenário de movimento circular é o que mais afeta a técnica DR, e, como a técnica MAADR têm o movimento circular em sua modelagem, a diferença do erro entre as duas é alta. O aumento do erro entre as técnicas tende a ter um aumento significativo conforme a técnica DR tem um aumento grande, que precisa ser provado.

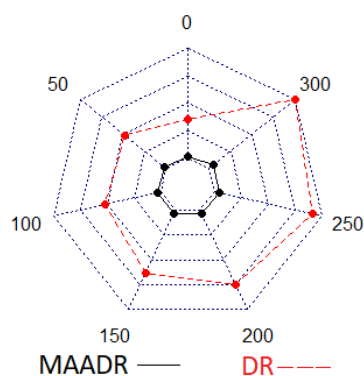


Figura 12 – Gráfico de radar com o erro de ambas as técnicas para cada valor de latência para o cenário 2.

Na Figura 13, a apresentação da médias dos erros para ambas as técnicas sob as latências mostradas no eixo X pode ser vista. Como mencionado anteriormente, movimentos circulares tendem a causar grande erro resultante da técnica DR, e quase nada de erro com a técnica MAADR, pela sua própria modelagem. O erro da técnica DR e o aumento do mesmo é exacerbado pelo aumento da latência, que pode ser visto na figura em questão.

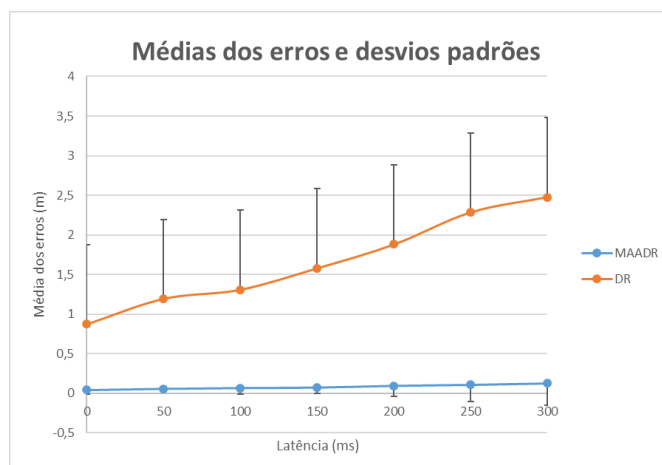


Figura 13 – Cenário 2: Média dos erros e desvios padrões ao longo das latências para ambas as técnicas.

A menor precisão da técnica DR, comparada à maior precisão da técnica MAADR em movimento circulares, foi previamente mostrado por Kharitonov [9] quando inicialmente propôs a sua técnica. Portanto, o resultado pior da técnica DR não é sem precedentes. Porém, a grande proporção com que a técnica DR aumenta o erro, quando submetida a grandes latências, se comparada com a técnica MAADR, mostra a importância dessa avaliação previamente não feita. Para avaliar, para cada valor de latência, se a diferença entre ambas as técnicas é significativa para o cenário 2, o teste MW foi usado.

A Tabela 2 mostra o p-valor do teste MW entre as duas técnicas, para cada valor de latência. Temos que, como a diferença entre os conjuntos das duas técnicas para cada valor de latência é muito grande, o p-valor sempre foi menor do que o menor valor que o método usado calcula (de  $2,2e-16$ ), dentro do ambiente R. E como o p-valor para cada latência é muito próximo de 0, temos que para todos os casos do cenário 2 as duas técnicas são diferentes de forma significativa.

Tabela 2 – P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 2.

	0 ms	50 ms	100 ms	150 ms	200 ms	250 ms	300 ms
P-Valor	< $2,2e-16$	< $2,2e-16$	< $2,2e-16$	< $2,2e-16$	< $2,2e-16$	< $2,2e-16$	< $2,2e-16$

Como a técnica MAADR tem em seu modelo o movimento circular, enquanto a técnica DR não tem, os resultados visualizados na figura 13 e na tabela 2 eram esperados. Além da significância entre o erro de ambas as técnicas, necessita-se avaliar se o crescimento do erro de ambas as técnicas é significativo para o cenário 2.

Para isso, assim como para o cenário 1, foram criados os conjuntos lat0 até lat300, variando de 50 em 50, que representam os conjuntos das diferenças dos erros da técnica DR com a técnica MAADR para cada valor de latência para o cenário 2. Para avaliar se há uma diferença

significativa entre pelo menos dois conjuntos lat's, o teste KW foi usado. Como o p-valor resultante do teste KW é igual a  $2,2e-16$ , que é um valor muito pequeno, rejeita-se a hipótese nula, concluindo-se que pelo menos um dos conjuntos testados tem uma diferença significativa com pelo menos um dos outros conjuntos avaliados. Para identificar quais conjuntos têm diferença significativa entre si, o teste de Dunn foi usado.

A Figura 14 mostra as médias das diferenças dos erros de cada conjunto e os grupos concluídos a partir da análise estatística de Dunn.

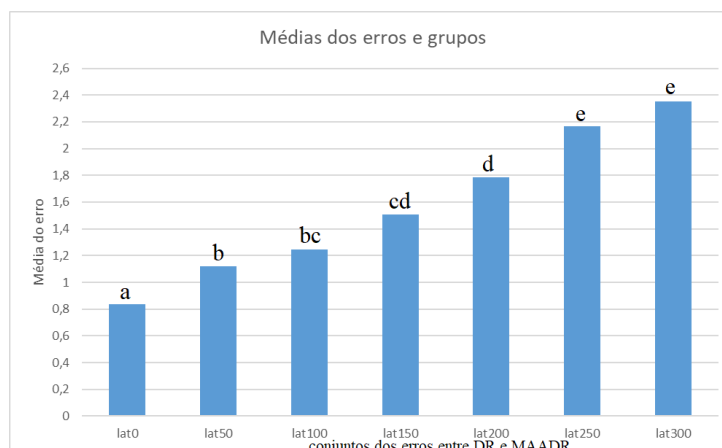


Figura 14 – Médias das diferenças dos erros e grupos para o cenário 2.

Como esse cenário 2 causa grande erro para a técnica DR, evidenciado e exacerbado pelo aumento da latência, enquanto não acontece o mesmo para a técnica MAADR, temos que o aumento da latência faz com que os conjuntos tenham uma diferença significativa sem necessitar um aumento de latência como ocorreu no cenário 1. O único conjunto que se diferencia de forma significativa é o conjunto lat0. Os demais conjuntos são sempre estatisticamente iguais com pelo menos um outro conjunto ou no máximo 2. Os conjuntos lat50, 100 e 150 formam um subgrupo de igualdade estatística, enquanto os conjuntos lat100, 150 e 200 formam outro. Por fim os conjuntos de lat 250 e 300 são estatisticamente iguais, porém diferentes dos demais. Isso mostra que, para movimentos curvilíneos não acelerados, há diferença de desempenho entre as técnicas, principalmente quando há aumento na latência.

#### 4.4 Cenário 3: Movimento Circular Variado

O cenário 3 foi criado para aumentar a complexidade do movimento, mas mantendo a ideia de um movimento circular, adicionando a presença de aceleração linear durante a curva, o que resulta em movimento circular variado. O movimento começa com valor de  $4 \text{ m/s}$  de velocidade linear e  $4,18 \text{ rad/s}$  de velocidade angular. A aceleração linear começa com um valor de  $1 \text{ m/s}^2$  e alterna seu valor para o seu oposto a cada 1 segundo, resultando em um movimento com média de velocidade linear de  $4,5 \text{ m/s}$ . Especialmente por causa da velocidade angular,

presente no padrão de movimento do jogador nesse cenário, o principal gênero que contém esse tipo de movimento é o de jogos de corrida, onde a presença de uma variação na aceleração linear durante curvas é provável de acontecer. Em jogos FPS esse padrão é menos provável de acontecer, especialmente porque a velocidade angular do jogador é diretamente relacionada à direção da mira do jogador, que por sua vez normalmente é controlada por periféricos diversificados, e dificilmente representa uma velocidade angular constante.

A Figura 15 mostra a trajetória feita pelo objeto remoto. A marca vermelha representa o início do movimento, com os valores iniciais de velocidade linear e angular, enquanto as demais marcas verdes representam as alterações no valor da aceleração linear para o seu oposto.

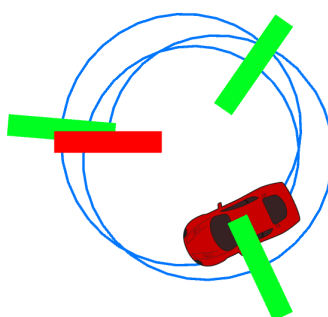


Figura 15 – Cenário 3: Movimento Espiral.

A Figura 16 ilustra o comportamento dos perfis de velocidade para o cenário 3. O gráfico mostra a presença de aceleração linear por todo o movimento circular, o que resulta no movimento circular variado. Em toda a trajetória do cenário 2 a velocidade angular é constante.

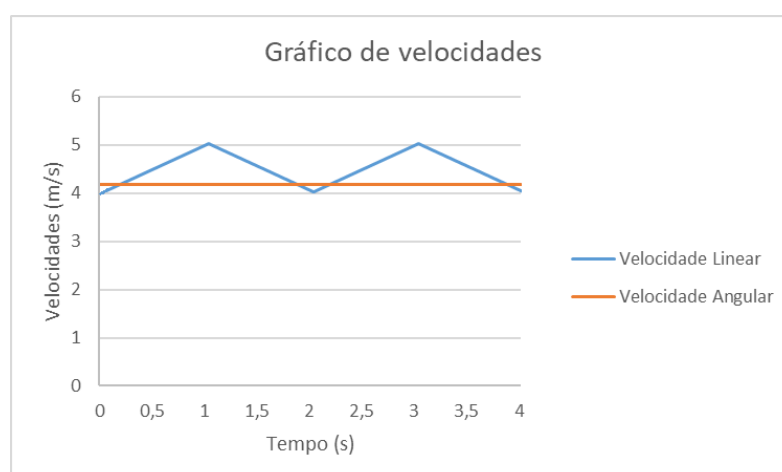


Figura 16 – Perfis de velocidade para o cenário 3: Movimento Circular Variado.

A Figura 17 mostra a média dos erros de cada técnica para cada valor de latência em um gráfico de radar, podendo-se visualizar o aumento do erro conforme o valor da latência aumenta. Como o cenário 3 representa um movimento circular variado, mais complexo que o

movimento circular presente no cenário 2, este é o cenário que mais afeta a técnica DR em relação ao seu erro. A área menor da técnica MAADR para cada valor de latência mostra que a técnica MAADR resulta em um menor erro.

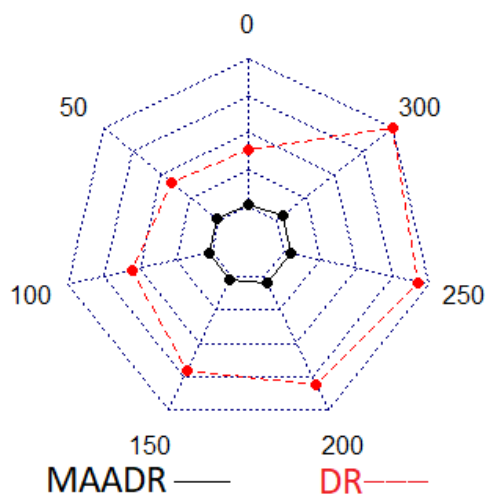


Figura 17 – Gráfico de radar com o erro de ambas as técnicas para cada valor de latência para o cenário 3.

A Figura 18 apresenta as médias dos erros para ambas as técnicas, sob as latências avaliadas do eixo X. O movimento circular variado mais complexo, com uma aceleração linear e uma constante velocidade angular, mostra que a técnica DR sofre tanto com o movimento que dificulta sua predição, quanto com o aumento da latência.

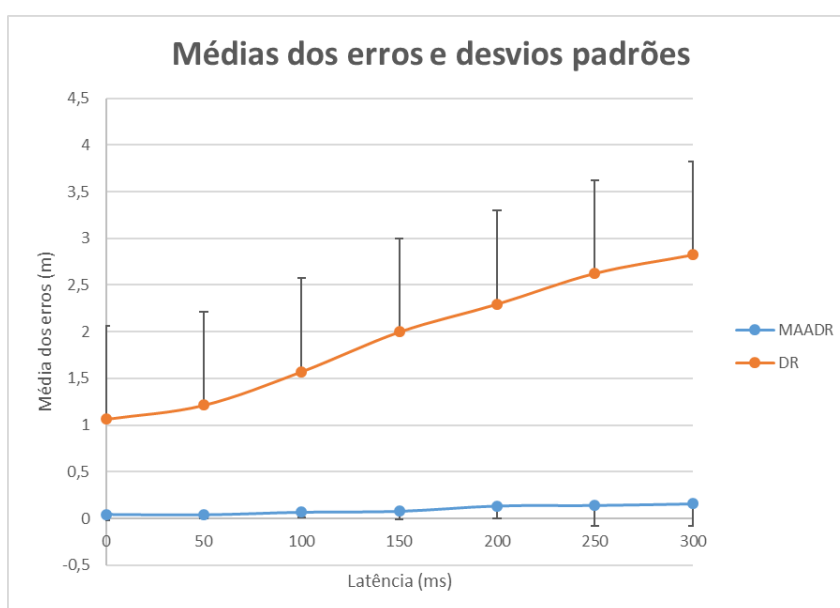


Figura 18 – Cenário 3: Médias dos erros e desvios padrões através das latências.

Para avaliar se a diferença entre o erro das duas técnicas é significativa para cada valor de latência, o teste MW foi usado.

A Tabela 3 mostra o p-valor resultante do teste MW entre cada técnica e para cada valor de latência no cenário 3.

Tabela 3 – P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 3.

	0 ms	50 ms	100 ms	150 ms	200 ms	250 ms	300 ms
P-Valor	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16

De forma parecida com o cenário 2, temos que a diferença entre os conjuntos dos erros de cada técnica é muito grande, fazendo com que o p-valor do teste MW seja menor do que  $2,2e-16$  para todos os valores de latência, significando que os conjuntos de erros das duas técnicas são significativamente diferentes. Mesmo que a diferença entre o erro das duas técnicas seja visível, torna-se necessário analisá-las estatisticamente.

Assim como para os cenários anteriores, foram criados os conjuntos das diferenças dos erros da técnica DR com a técnica MAADR para cada valor de latência de 0 até 300, resultando nos conjuntos lat0 até lat300 (variando de 50 em 50). O teste KW foi usado para analisar e comparar a distribuição de cada conjunto lat.

O p-valor resultante do teste KW foi de  $< 2,2e-16$ . Por ser um valor muito próximo de 0, rejeita-se a hipótese nula, possibilitando concluir que há pelo menos um conjunto avaliado que é significativamente diferente de algum outro conjunto. Para identificar quais conjuntos são significativamente diferentes no cenário 3, o teste de Dunn foi utilizado, assim como para os cenários anteriores.

A Figura 19 mostra as médias das diferenças dos erros de cada conjunto e os grupos concluídos a partir da análise estatística de Dunn.

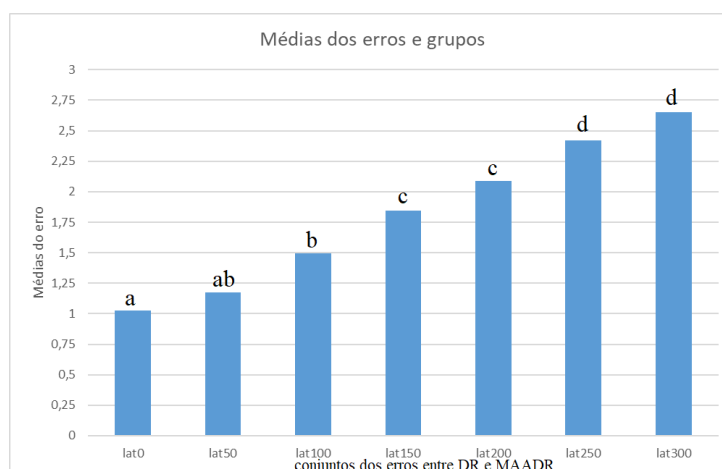


Figura 19 – Médias das diferenças dos erros e grupos para o cenário 3.

O cenário 3 representa um movimento circular mais complexo que o do cenário 2, portanto, causa maiores erros e maior aumento nos erros, o que pode ser visto no gráfico da Figura

18. Esse movimento tem um padrão parecido com o cenário 2, onde os conjuntos são estatisticamente iguais a um ou no máximo dois outros conjuntos. Temos que os pares de conjuntos 0 e 50; 50 e 100; 150 e 200; 250 e 300 formam subgrupos de conjuntos que são estatisticamente iguais entre si. Considerando a diferença significativa e visualizando a figura 18, conclui-se que o erro da técnica DR aumenta de forma significativa comparada à técnica MAADR conforme a latência aumenta para o cenário 3, de forma parecida com o cenário 2.

#### 4.5 Cenário 4: Movimento zigue-zague.

O cenário 4 foi criado para estressar as técnicas com uma mistura de elementos complexos presentes em cenários passados. Uma aceleração linear constante está presente por todo o escopo do cenário, com a presença de mudanças abruptas na velocidade angular de um valor para o seu oposto, resultando em um movimento de zigue-zague. As mudanças abruptas nos padrões de movimento, presentes nesse cenário, podem ser vistas em diversos gêneros de jogos, como os de corrida, onde aparecem principalmente nos momentos em que o jogador se depara com uma pista que demanda um movimento de zigue-zague. Outros gêneros também têm esse padrão presente, porém, com intervalos menos definidos entre as mudanças abruptas de movimento. Em jogos com o gênero FPS, o jogador pode escolher fazer esse padrão de movimento ao tentar desviar ou enganar o outro jogador quanto a sua posição e causar mais erros nos tiros por parte do seu oponente.

Esse cenário foi inspirada no segundo e terceiro cenários criado por Kharitonov [9], onde há uma avaliação de ambas as técnicas sob mudanças abruptas de direção e como elas se adaptam às constantes mudanças de padrões de movimento.

A Figura 20 mostra a trajetória feita pelo objeto remoto no cenário 4. A trajetória flui da esquerda para a direita, resultando em um movimento de zigue-zague. A primeira marca vermelha denota o início do movimento, onde há a presença de uma velocidade linear e angular e uma aceleração linear inicial. As demais marcas vermelhas indicam os momentos em que a velocidade angular e a aceleração linear se alteram para os seus opostos.

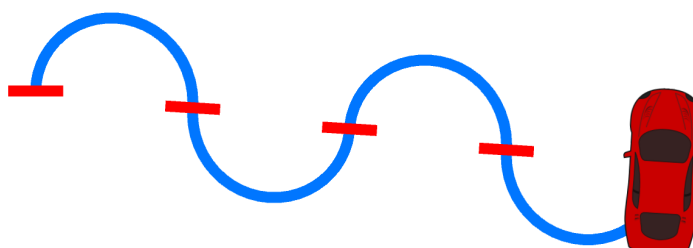


Figura 20 – Cenário 4: Movimento Zigue-zague.



Os resultados podem ser vistos na Figura 21, em que o gráfico mostra os valores para as velocidades angular e linear através das cenários.

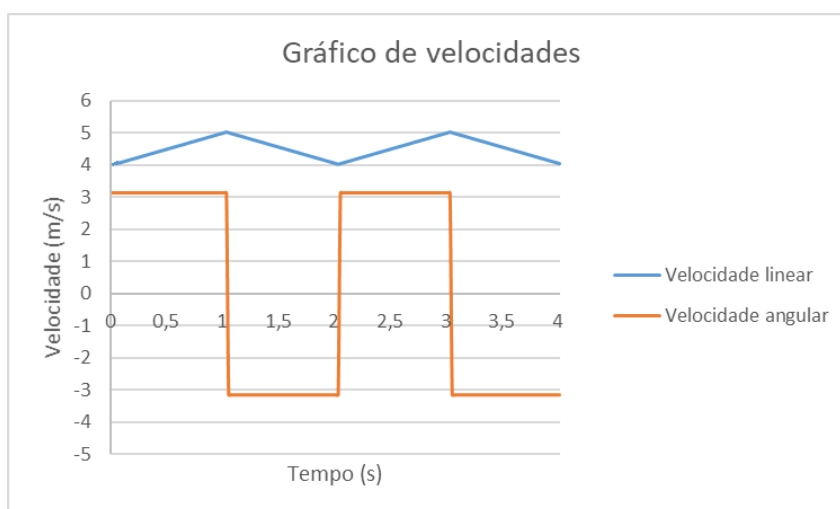


Figura 21 – Perfis de velocidade para o cenário 4: Movimento de Zigue-zague.

A Figura 22 mostra as médias dos erros de cada técnica, para cada valor de latência, na forma de um gráfico de radar. Diferentemente dos cenários anteriores, o tipo de movimento zigue-zague causa erros nas predições tanto da técnica DR quanto da técnica MAADR, porém, afetando a última de uma forma menos severa. Essa influência sobre as técnicas pode ser visualizada na figura abordada, onde a menor área da técnica MAADR resulta em um menor erro.

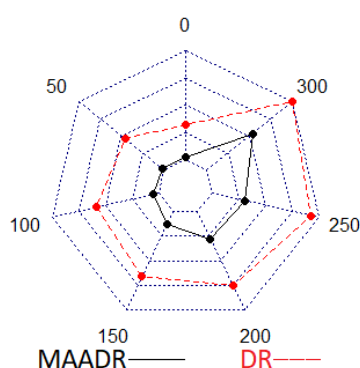


Figura 22 – Gráfico de radar com a média dos erros de ambas as técnicas para cada valor de latência no cenário 4.

A Figura 23 mostra a média dos erros para cada técnica, MAADR e DR, ao longo das latências presentes no eixo X para o cenário 4. Para esse cenário, a ideia de ter a presença de uma aceleração linear e um movimento circular é expandida para criar mudanças abruptas de direção, resultando no movimento de zigue-zague e assim, num movimento complexo para ambas as técnicas, de forma que o gráfico abordado mostra isso.

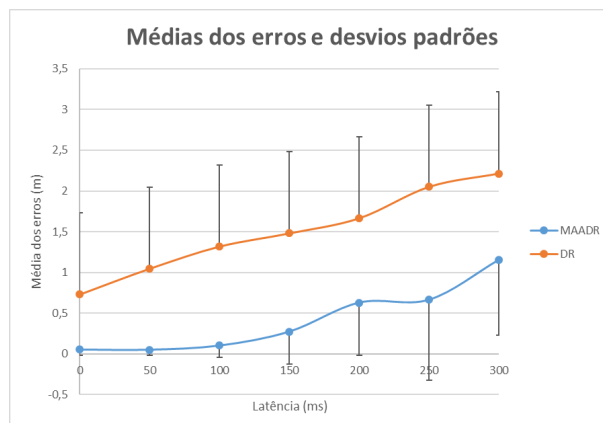


Figura 23 – Cenário 4: Médias dos erros e desvios padrões através das latências.

A complexidade do cenário reflete os resultados encontrados, em que é possível visualizar que ambas as técnicas têm suas precisões diminuídas conforme a latência aumenta. Diferentemente dos cenários anteriores, a técnica MAADR também sofre com uma magnitude maior. Mesmo com esse aumento no erro, a visualização dos resultados da Figura 23 indicam que a técnica DR ainda tem uma proporção de aumento de erro grande comparada à magnitude do aumento do erro da técnica DR. Para avaliar a significância da diferença do erro entre ambas as técnicas para cada valor de latência, o teste estatístico MW foi usado.

A Tabela 4 mostra o p-valor resultante do teste estatístico MW aplicado para cada par de técnica e latência.

Tabela 4 – P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 4.

	0 ms	50 ms	100 ms	150 ms	200 ms	250 ms	300 ms
P-Valor	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16	< 2,2e-16

Mesmo com o aumento da complexidade do movimento, o erro da técnica DR é alto o suficiente para ter uma diferença significativa relativa ao erro da técnica MAADR. O resultado do teste MW corrobora com essa afirmativa, de forma que em todos os valores de latência o p-valor teve resultado < 2,2e-16, evidenciando que há uma diferença significativa entre as técnicas em cada uma das comparações.

Foram criados, assim como em cenários anteriores, os conjuntos das diferenças dos erros entre a técnica DR e MAADR para cada valor de latência lat0 até lat300 (variando de 50 em 50). O teste estatístico KW foi então usado para verificar se há uma diferença significativa entre os conjuntos lats, com objetivo de avaliar se há um aumento significativo entre o erro das técnicas conforme a latência aumenta.

Como o p-valor resultante do teste KW, de 1,029e-13, é um valor muito pequeno, próximo de 0, temos que a hipótese nula é rejeitada. Isso possibilita concluir que há pelo menos um

conjunto lat que difere significativamente de pelo menos um outro conjunto lat. Para analisar a significância da diferença entre os conjuntos de cada latência, o teste de Dunn foi usado.

A Figura 24 mostra as médias das diferenças dos erros de cada conjunto e os grupos concluídos a partir da análise estatística de Dunn.

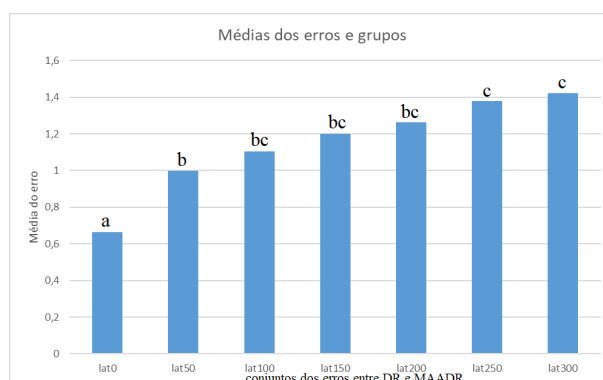


Figura 24 – Médias das diferenças dos erros e grupos para o cenário 4.

Sendo um cenário mais complexo, com mudanças abruptas de movimento, temos que não somente a técnica DR sofre com o aumento da latência, mas também a técnica MAADR. Os grupos gerados a partir do teste de Dunn mostram, sob latência 0, que o conjunto da diferença dos erros é significativamente diferente dos erros de todas as demais latências, enquanto os demais conjuntos de erros não têm uma diferença significativa com pelo menos um outro conjunto de erros.

Com as diversas relações de diferenças não significativas, que podem ser vistas com os grupos de lat50 até lat300, pode-se entender que há uma diferença significativa quando as técnicas são postas sob uma latência pequena de 50 ms, ao comparar com os seus desempenhos sem latência simulada. Porém, ao termos uma crescente latência, vemos que a diferença entre os níveis de latência não se tornam tão significativos. A Figura 24 com os grupos gerados a partir da análise de Dunn e a Figura 23, indicam que a técnica DR, em um cenário complexo com movimento abruptos, não tem um aumento do seu erro tão significativo quando comparado com a diferença dos erros dos cenários 2 e 3. Porém o aumento da diferença dos erros ainda é significativo. Um ponto de comparação entre as duas técnicas importante que também precisa ser considerado é a própria magnitude do erro entre as técnicas, que no caso desse cenário é significativa, como foi mostrado com o teste MW para cada par de técnicas para cada valor de latência.

#### 4.6 Cenário 5: Movimento aleatório

O cenário final 5, inspirado pela avaliação de Aggarwal [14], implementa movimentos aleatórios a fim de estressar ambas as técnicas e ter um comportamento mais próximo de um

jogador real, no qual é difícil prever os movimentos. A aleatoriedade desse cenário foi obtida com o uso do gerador pseudo aleatório presente na *Application Programming Interface* (API) do Unity. Esse gerador foi usado para gerar uma série de entradas de dados e foi usado e configurado para gerar uma velocidade linear entre 4 e 5  $m/s$ , velocidade angular entre -4,18 e 4,18  $rad/s$ , aceleração angular entre -1,57 e 1,57  $rad/s^2$  e o tempo de intervalo entre as entradas de dados entre 0,1 e 1 segundo. A aceleração linear manteve-se variando entre 1 e -1, a fim de manter uma média de 4,5  $m/s$  ao longo do cenário. O resultado do uso do gerador pseudo aleatório nessa configuração foi a presença de diversas mudanças abruptas nos padrões de movimento, resultando num cenário com mais ruídos. Por causa da configuração dos parâmetros que foram usados, o principal gênero em que esse cenário se encaixa é o de corrida. Porém, esse comportamento também pode ser encontrado em jogos de FPS por causa do movimento do jogador que constantemente muda com eventos abruptos.

A Figura 25 mostra a trajetória feita pelo objeto remoto no cenário 5, que flui de baixo para cima configurando um movimento aleatório.

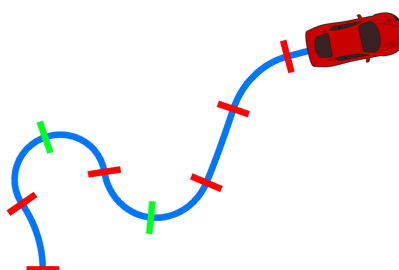


Figura 25 – Cenário 5: Movimento aleatório.

Os perfis de velocidade para esse cenário podem ser vistos na Figura 26, em que um gráfico mostra os valores para as velocidades linear e angular ao longo do cenário.

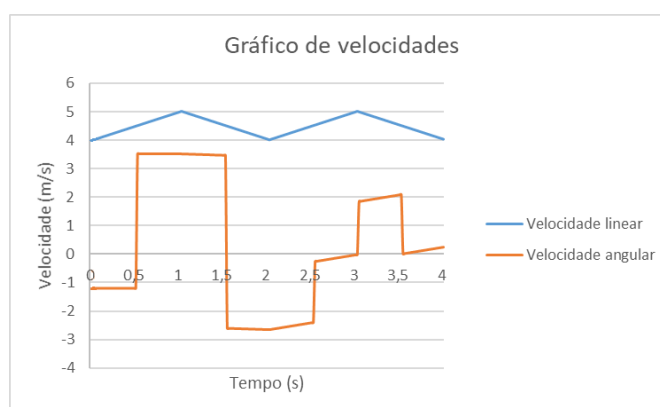


Figura 26 – Perfis de velocidade para o cenário 5: Movimento aleatório.

A Figura 27 mostra um gráfico de radar com as médias dos erros de ambas as técnicas

para cada valor de latência. Mais próximo do cenário 4 que dos demais cenários, com o gráfico de radar é possível ver que as mudanças abruptas de movimento fazem com que ambas as técnicas sejam afetadas em sua precisão conforme a latência aumenta. A área menor que a técnica MAADR apresenta resulta em um erro menor.

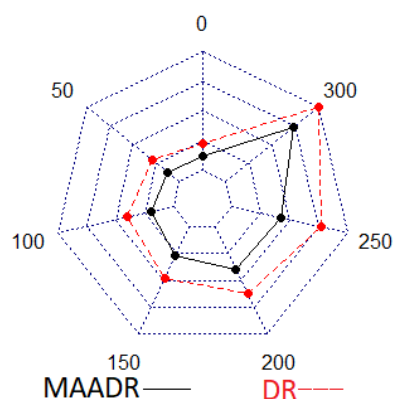


Figura 27 – Gráfico de radar com o erro de ambas as técnicas para cada valor de latência para o cenário 5.

A Figura 28 mostra a média dos erros e desvios padrões de cada técnica, MAADR e DR, ao longo do eixo X para o cenário 5. Nesse cenário a exploração de uma situação aleatória com mudanças abruptas dos padrões de movimento foi dada, resultando não somente em um cenário complexo, mas também em uma avaliação mais próxima de um caso real.

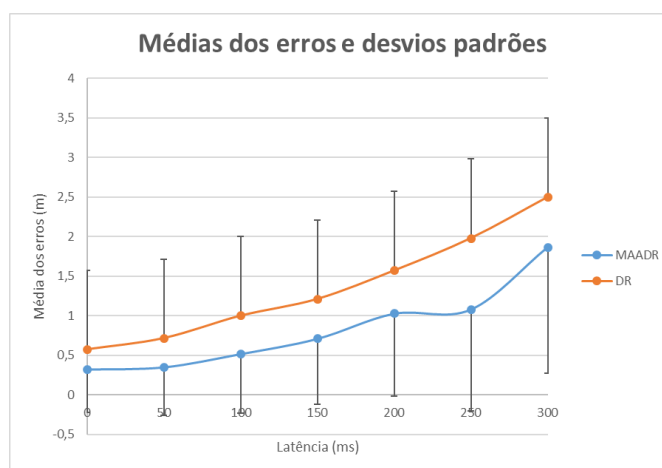


Figura 28 – Cenário 5: Médias dos erros e desvios padrões através das latências.

A característica abrupta desse cenário é presente nos resultados finais. De forma geral, a técnica MAADR apresentou melhor precisão geral do que a técnica DR, como esperado. A tendência de maior magnitude do erro da técnica DR, comparada ao erro da técnica MAADR, pode ser vista. A presença de grande número de mudanças abruptas afetou os picos dos erros não somente da técnica DR, mas também da MAADR, fazendo com que os desvios padrões

do MAADR tenham magnitude muito mais próxima dos desvios da técnica DR, se comparado com outros cenários.

Para avaliar a significância da diferença entre os erros de ambas as técnicas, para cada valor de latência, o teste estatístico MW foi usado para cada par. A Tabela 5 mostra os p-valores resultantes do teste estatístico MW para as técnicas MAADR e DR para cada valor de latência.

Tabela 5 – P-Valores do teste Mann-Whitney entre as duas técnicas para cada latência no cenário 5.

	0 ms	50 ms	100 ms	150 ms	200 ms	250 ms	300 ms
P-Valor	< 2,2e-16	< 2,2e-16	< 2,2e-16	1,32e-12	3,028e-09	9,259e-14	2,523e-08

Todos os p-valores de resultados presentes na Tabela 5 são muito próximos a 0, significando que rejeita-se a hipótese nula e conclui-se que há uma diferença significativa entre cada par de técnicas para cada latência.

Para então avaliar se a diferença do erro entre as técnicas tem uma diferença significativa, foram criados os conjuntos das diferenças dos erros entre as técnicas DR e MAADR para cada valor de latência, lat0 até lat300 (variando de 50 em 50), assim como foi feito nos cenários anteriores. O teste estatístico KW foi usado para avaliar se há uma diferença significativa entre os conjuntos das diferenças dos erros.

O p-valor gerado pelo teste KW, de 4,438e-05, é um valor pequeno. Isso conclui em rejeitar a hipótese nula e significa que há pelo menos um conjunto avaliado que é diferente de forma significativa de pelo menos um outro conjunto de erros testado. O teste de Dunn então foi aplicado para identificar quais conjuntos são diferentes de forma significativa de quais outros conjuntos.

A Figura 29 mostra as médias das diferenças dos erros de cada conjunto e os grupos concluídos a partir da análise estatística de Dunn.

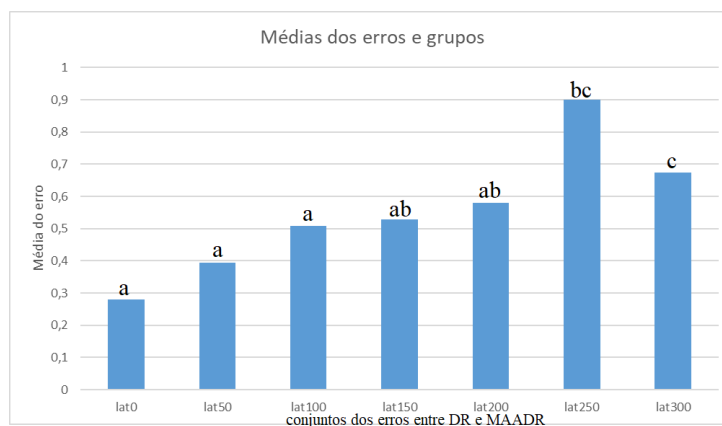


Figura 29 – Médias das diferenças dos erros e grupos para o cenário 5.

O cenário 5 segue os mesmos padrões vistos no cenário 4, por ser um cenário com um movimento complexo. Esse movimento complexo causa aumento no erro de ambas as técnicas conforme a latência aumenta. O agrupamento feito pelo teste de Dunn mostra que os conjuntos lat0, lat50, lat100, lat150 e lat200 não têm uma diferença significativa entre si, de forma que com a visualização da Figura 28, nota-se que mesmo que a técnica DR tenha magnitude de erro maior, o aumento do seu erro conforme a latência aumenta é estatisticamente igual ao aumento do erro da técnica MAADR.

Pode-se formar, também, um segundo subgrupo de conjuntos que não têm diferenças significativas entre si, contendo lat150, lat200 e lat250. Um terceiro subgrupo de conjuntos sem diferença significativa pode ser formado com os conjuntos lat250 e lat300. O aumento da latência faz uma diferença significativa surgir na comparação entre as duas técnicas, com a técnica DR aumentando o seu erro com maior taxa do que a técnica MAADR, porém, de forma mais lenta nesse cenário mais complexo.

## 4.7 Discussão

Cada cenário foi proposto com a intenção de avaliar movimentos simples e complexos que podem ser feitos por jogadores reais, de forma que foram inspirados em padrões de movimentos reais e avaliações de autores chave da literatura.

A Figura 30 mostra o gráfico de radar das médias dos erros de ambas as técnicas para cada cenário, de 1 a 5, sob 0 ms de latência simulada. O gráfico mostra quanto o tipo de movimento abordado em cada cenário afeta cada algoritmo, independente de latência.

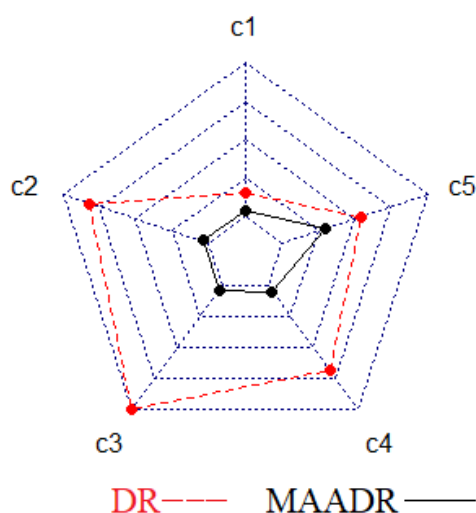


Figura 30 – Gráfico de radar com as médias dos erros de cada técnica para cada Cenário sob 0 ms de latência simulada.

A Tabela 6 mostra os valores das médias dos erros de cada ponto do gráfico de radar.

Tabela 6 – Tabela da média dos erros de cada técnica em cada cenário

	cenário 1	cenário 2	cenário 3	cenário 4	cenário 5
MAADR	0,04218	0,0419	0,0426	0,0513	0,3258
DR	0,165135	0,8756	1,0662	0,7322	0,5760

Nota-se que a técnica DR tem grande perda de precisão no segundo e terceiro cenários, que abordam os movimentos circulares, como pôde ser visto na seção dos respectivos cenários. Essa perda é compreensível, considerando que a predição da técnica DR é sempre em linha reta, fazendo com que um movimento circular o afete de forma severa. Observando a média para o cenário 4 temos que, mesmo com uma complexidade maior, a média do erro é menor, visto que uma predição em linha reta é mais próxima ao resultado real final do movimento em zigue-zague. Como a técnica MAADR tem em sua modelagem parte dos movimentos complexos abordados, há um crescimento do erro, sem os picos encontrados com a técnica DR, conforme o cenário se torna mais complexo até chegar no movimento aleatório, que é menos previsível.

Em todos os cenários, a técnica MAADR teve uma melhor precisão do que a técnica DR. A principal avaliação e resultados desse trabalho é o comportamento de ambas as técnicas sob médias e altas latências. Os resultados estatísticos e visuais mostram que ambas as técnicas perdem precisão conforme a latência aumenta, porém, a técnica DR perde precisão com uma taxa muito maior do que a MAADR nos cenários com movimentos mais simples e mais intrínsecos à modelagem da técnica MAADR.

O primeiro cenário é composto por um movimento de complexidade muito baixa, portanto, a técnica DR se mantém com alta precisão, tendo apenas algumas quedas nos momentos de aceleração ou desaceleração. A técnica MAADR, como modela a presença de aceleração e envia mensagens preemptivas, têm ganho de precisão perceptível e visível no gráfico das médias dos erros do cenário 1, comprovada também nos testes estatísticos feitos para avaliar as técnicas.

O segundo e o terceiro cenários têm movimentos parecidos, onde o cenário 2 propõe um movimento circular constante e o cenário 3 propõe um movimento circular variado com a presença de aceleração. Neste tipo de movimento, a presença de velocidade angular e de aceleração linear constam na modelagem da técnica MAADR e fazem o objeto local se mover em direções contrárias às predições da técnica DR. Portanto, a diferença entre a precisão das técnicas é perceptível e mostrada pelos testes estatísticos, que indicam diferença significativa entre as técnicas a cada aumento da latência. Isso evidencia que em cenários com movimentos lineares ou circulares com pouca variação ou poucas modificações abruptas nos padrões de movimento, a técnica MAADR tem precisão maior do que a técnica DR e se adapta melhor a condições de latência adversas, tendo não somente menor magnitude do erro, mas diferença no crescimento do erro significativamente menor do que a técnica DR.

O quarto e o quinto cenários têm, também, movimentos parecidos, nos quais ambos



os cenários apresentam movimentos complexos para avaliar ambas as técnicas. O cenário 4 apresenta uma situação de zigue-zague com mudanças abruptas nos movimentos, enquanto o cenário 5 apresenta um movimento aleatório também com mudanças abruptas nos movimentos. Com a presença de movimentos mais complexos e abruptos, temos que a precisão de ambas as técnicas tendem a diminuir, de forma que a visualização dos gráficos, somados com os resultados dos testes estatísticos, mostram que há diferença significativa entre as diferenças dos erros das técnicas MAADR e DR, ou seja, a técnica DR tem aumento no seu erro com uma taxa significativa maior do que a MAADR, porém, de forma mais lenta do que nos cenários 1, 2 e 3, precisando de maior variação na latência para obter significância no seu aumento. Mesmo com o aumento da diferença do erro entre as técnicas com uma taxa menor, a diferença na magnitude do erro entre as técnicas se mantém presente.

Henderson *et al* [45] concluiu que, com uma latência de 100 milissegundos, o atraso nas ações é perceptível para alguns jogadores; para uma latência de 200 milissegundos, praticamente todos os jogadores percebem o atraso e; para um atraso de 300 milissegundos, a perda de QoS é inevitável. A técnica DR é muito usada por causa da sua simplicidade e eficácia suficiente mas, em casos de médias e altas latências, uma maior magnitude do erro da técnica DR pode fazer com que jogadores percam suficiente QoS e decidam não permanecer no AVD. O uso da técnica MAADR para esses casos, nos quais médias e altas latências são esperadas, é extremamente recomendado para manter níveis adequados de QoS.

## 5 CONCLUSÃO

A principal contribuição desse trabalho é a análise experimental e estatística da performance das técnicas DR e MAADR sob relevantes intervalos de latência, que contempla cenários com alta responsividade e maior consistência (0 milissegundos de atraso) até altos níveis de latência (300 milissegundos, no qual a perda de interesse dos jogadores é inevitável), aplicada em jogos multijogador. Os resultados obtidos mostram que a técnica DR perde uma grande magnitude de sua performance comparada a técnica MAADR, de forma que o aumento do erro da técnica DR é significativamente maior do que da técnica MAADR, que consegue manter bons níveis de QoS sob médias e altas latências. Sob baixas latências, a diferença de erro entre as duas técnicas é menor, porém, os testes Mann-Whitney em cada cenário mostram que existe uma diferença significativa entre ambas as técnicas, mesmo sob 0 de latência, que não deve ser ignorada. Porém, em médias e altas latências (100 a 300 milissegundos), a taxa com que a técnica DR aumenta o seu erro, a significância da diferença entre o aumento do erro nas as técnicas e a magnitude do erro, de uma forma geral, faz a informação compartilhada menos consistente e aumenta consideravelmente a perda de QoS, principalmente no caso do uso da técnica DR. Por essas razões, o uso da técnica MAADR nos cenários em que médias e altas latências estão presentes é altamente recomendada, assim como nos casos onde é de interesse evitar perdas na QoS dos jogadores. A adoção da técnica MAADR no lugar da DR proveria aumento da QoS e do interesse com o jogo.

## REFERÊNCIAS

- [1] SCHUYTEMA, P. *Design de games: uma abordagem prática*. [S.l.]: Cengage Learning, 2008.
- [2] 2016-2020 Global games market. <https://newzoo.com/insights/articles/new-gaming-boom-newzoo-ups-its-2017-global-games-market-estimate-to-116-0bn-growing-to-143-5bn-in-2020/>. Accessed: 2018. Disponível em: <<https://newzoo.com/insights/articles/new-gaming-boom-newzoo-ups-its-2017-global-games-market-estimate-to-116-0bn-growing-to-143-5bn-in-2020/>>.
- [3] ESPORTS: A Brief History. <http://adanai.com/esports/>. Accessed: 2018. Disponível em: <<http://adanai.com/esports/>>.
- [4] SAVERY, C.; GRAHAM, T. C. N. Timelines: simplifying the programming of lag compensation for the next generation of networked games. *Multimedia Systems*, v. 19, n. 3, p. 271–287, 2013. ISSN 1432-1882. Disponível em: <<http://dx.doi.org/10.1007/s00530-012-0271-3>>.
- [5] CHEN, Y.; LIU, E. S. A path-assisted dead reckoning algorithm for distributed virtual environments. In: IEEE. *2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. [S.l.], 2015. p. 108–111.
- [6] STYTZ, M. R. Distributed virtual environments. *IEEE Computer Graphics and Applications*, IEEE, v. 16, n. 3, p. 19–31, 1996.
- [7] BOULANGER, J.-S.; KIENZLE, J.; VERBRUGGE, C. Comparing interest management algorithms for massively multiplayer games. In: ACM. *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*. [S.l.], 2006. p. 6.
- [8] CHEN, P.; ZARKI, M. E. Perceptual view inconsistency: an objective evaluation framework for online game quality of experience (qoe). In: IEEE. *Network and Systems Support for Games (NetGames), 2011 10th Annual Workshop on*. [S.l.], 2011. p. 1–6.
- [9] KHARITONOV, V. Y. Motion-aware adaptive dead reckoning algorithm for collaborative virtual environments. In: ACM. *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. [S.l.], 2012. p. 255–261.
- [10] GAO, C.; SHEN, H.; BABAR, M. A. Concealing jitter in multi-player online games through predictive behaviour modeling. In: IEEE. *Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on*. [S.l.], 2016. p. 62–67.
- [11] LEE, K. et al. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In: ACM. *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. [S.l.], 2015. p. 151–165.
- [12] QUAKE Engine code review : Prediction. <http://fabiansanglard.net/quakeSource/quakeSourcePrediction.php>. Accessed: 2018. Disponível em: <<http://fabiansanglard.net/quakeSource/quakeSourcePrediction.php>>.

- [13] LATENCY Compensating Methods in Client/Server In-game Protocol Design and Optimization. [https://developer.valvesoftware.com/wiki/Latency\\_Compensating\\_Methods\\_in\\_Client/Server\\_In-game\\_Protocol\\_Design\\_and\\_Optimization](https://developer.valvesoftware.com/wiki/Latency_Compensating_Methods_in_Client/Server_In-game_Protocol_Design_and_Optimization). Accessed: 2018. Disponível em: <[https://developer.valvesoftware.com/wiki/Latency\Compensating\Methods\in\\\_Client/Server\\\_In-game\\\_Protocol\\\_Design\\\_and\\\_Optimization](https://developer.valvesoftware.com/wiki/Latency\Compensating\Methods\in\_Client/Server\_In-game\_Protocol\_Design\_and\_Optimization)>.
- [14] AGGARWAL, S. et al. Accuracy in dead-reckoning based distributed multi-player games. In: ACM. *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*. [S.l.], 2004. p. 161–165.
- [15] HOWARD, E. et al. Cascading impact of lag on quality of experience in cooperative multiplayer games. In: *Proceedings of the 13th Annual Workshop on Network and Systems Support for Games*. Piscataway, NJ, USA: IEEE Press, 2014. (NetGames '14), p. 13:1–13:6. Disponível em: <<http://dl.acm.org/citation.cfm?id=2755535.2755551>>.
- [16] WOLF, M. J. Genre and the video game. *The medium of the video game*, Austin: University of Texas Press, p. 113–134, 2001.
- [17] INKPEN, K. M. Drag-and-drop versus point-and-click mouse interaction styles for children. *ACM Transactions on Computer-Human Interaction (TOCHI)*, ACM, v. 8, n. 1, p. 1–33, 2001.
- [18] EGENFELDT-NIELSEN, S.; SMITH, J. H.; TOSCA, S. P. *Understanding video games: The essential introduction*. [S.l.]: Routledge, 2013.
- [19] STARCRAFT. <https://starcraft.com>. Accessed: 2018. Disponível em: <<https://starcraft.com>>.
- [20] WARCRAFT 3: Reign Of Chaos. <http://us.blizzard.com/pt-br/games/war3/>. Accessed: 2018. Disponível em: <<http://us.blizzard.com/pt-br/games/war3/>>.
- [21] DOTA 2. <http://br.dota2.com/>. Accessed: 2018. Disponível em: <<http://br.dota2.com/>>.
- [22] DUNGEONS And Dragons. <http://dnd.wizards.com/>. Accessed: 2018. Disponível em: <<http://dnd.wizards.com/>>.
- [23] MATTHEUS, D. T. *Rogue - Video Game*. Chapel Hill, NC, USA: Ventana Press, Inc., 2012. ISBN 6138576403, 9786138576402.
- [24] BALDUR'S Gate. <https://www.baldursgate.com/>. Accessed: 2018. Disponível em: <<https://www.baldursgate.com/>>.
- [25] FALLOUT. <https://fallout.bethesda.net/>. Accessed: 2018. Disponível em: <<https://fallout.bethesda.net/>>.
- [26] ARSENAULT, D. Action. Routledge, 2014.
- [27] PROJECTCARS. <https://www.projectcarsgame.com/?lang=pt-br>. Accessed: 2018. Disponível em: <<https://www.projectcarsgame.com/?lang=pt-br>>.
- [28] JAYA, I.; LIU, E. S.; CHEN, Y. Combining interest management and dead reckoning: a hybrid approach for efficient data distribution in multiplayer online games. In: IEEE. *Distributed Simulation and Real Time Applications (DS-RT), 2016 IEEE/ACM 20th International Symposium on*. [S.l.], 2016. p. 92–99.

- [29] BLAU, B. et al. Networked virtual environments. In: ACM. *Proceedings of the 1992 symposium on Interactive 3D graphics*. [S.l.], 1992. p. 157–160.
- [30] WAGNER, M. G. On the scientific relevance of esports. In: *International conference on internet computing*. [S.l.: s.n.], 2006. p. 437–442.
- [31] SINGHAL, S.; ZYDA, M. *Networked virtual environments: design and implementation*. [S.l.]: ACM Press/Addison-Wesley Publishing Co., 1999.
- [32] LEVINE, N. E. B. B. N. Cheat-proof payout for centralized and distributed online games. Citeseer.
- [33] LEE, H. et al. Synchronization and cheat-proofing protocol for real-time multiplayer games. In: *Proceedings of 1st Workshop on Entertainment Computing, Makuhari, Japan*. [S.l.: s.n.], 2002.
- [34] MACEDONIA, M.; ZYDA, M. *Networked Virtual Environments Design and Implementation*. [S.l.]: Addison Wesley, 1999.
- [35] MAUVE, M. et al. Local-lag and timewarp: providing consistency for replicated continuous applications. *IEEE Transactions on Multimedia*, v. 6, n. 1, p. 47–57, Feb 2004. ISSN 1520-9210.
- [36] ZHANG, Y.; CHEN, L.; CHEN, G. Globally synchronized dead-reckoning with local lag for continuous distributed multiplayer games. In: ACM. *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*. [S.l.], 2006. p. 7.
- [37] CRONIN, E. et al. An efficient synchronization mechanism for mirrored game architectures. In: ACM. *Proceedings of the 1st workshop on Network and system support for games*. [S.l.], 2002. p. 67–73.
- [38] SHEN, H.; ZHOU, S. Achieving critical consistency through progressive slowdown in highly interactive multi-player online games. In: IEEE. *Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on*. [S.l.], 2013. p. 332–337.
- [39] HECKER, C.; SIMPSON, Z. Dead reckoning a.k.a. motion prediction. *Game Developer*, v. 8, n. 2, p. 10, 2001.
- [40] CAI, W.; LEE, F. B.; CHEN, L. An auto-adaptive dead reckoning algorithm for distributed interactive simulation. In: IEEE. *Parallel and Distributed Simulation, 1999. Proceedings. Thirteenth Workshop on*. [S.l.], 1999. p. 82–89.
- [41] STEPHEN, B.; CHEN, J. T. L. Adaptive dead reckoning algorithms for distributed interactive simulation. *School Of Computer Engineering Nanyang Technological University. Singapore*, v. 639798.
- [42] DUNCAN, T. P.; GRAČANIN, D. Algorithms and analyses: pre-reckoning algorithm for distributed virtual environments. In: WINTER SIMULATION CONFERENCE. *Proceedings of the 35th conference on Winter simulation: driving innovation*. [S.l.], 2003. p. 1086–1093.
- [43] UNITY game engine. <https://docs.unity3d.com/Manual/index.html>. Accessed: 2018. Disponível em: <<https://docs.unity3d.com/Manual/index.html>>.

- [44] NETWORKING framework for unity. <https://docs.unity3d.com/Manual/UNet.html>. Accessed: 2018. Disponível em: <<https://docs.unity3d.com/Manual/UNet.html>>.
- [45] HENDERSON, T.; BHATTI, S. Networked games: A qos-sensitive application for qos-insensitive users? In: *Proceedings of the ACM SIGCOMM Workshop on Revisiting IP QoS: What Have We Learned, Why Do We Care?* New York, NY, USA: ACM, 2003. (RIPQoS '03), p. 141–147. ISBN 1-58113-748-6. Disponível em: <<http://doi.acm.org/10.1145/944592.944601>>.
- [46] YATES, F. The design and analysis of factorial experiments. *Imp. Bur. Soil Sci. Horpendon. Tech. Comm*, v. 35, 1937.
- [47] MANN, H. B.; WHITNEY, D. R. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, JSTOR, p. 50–60, 1947.
- [48] KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, Taylor & Francis Group, v. 47, n. 260, p. 583–621, 1952.
- [49] DUNN, O. J. Multiple comparisons among means. *Journal of the American statistical association*, Taylor & Francis Group, v. 56, n. 293, p. 52–64, 1961.
- [50] RIPLEY, B. D. The r project in statistical computing. *MSOR Connections. The newsletter of the LTSN Maths, Stats & OR Network*, v. 1, n. 1, p. 23–25, 2001.

## TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

Publicação principal do trabalho.

1. Luis Fernando Kawabata de Almeida, Dr. Alan Salvany Felinto, **Evaluation of the Motion-Aware Adaptive Dead Reckoning Technique Under Different Network Latencies Applied in Multiplayer Games**, 2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), 02-2019, IEEE, 10, 978-1-5386-9605-7 (Qualis CC 2018, B2)