

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Engenharia Elétrica

Avaliação de Soluções de Inteligência Artificial
Preditiva para um Cenário de Internet Tátil

Cleuves Cajé de Carvalho

Cleuves Cajé de Carvalho

Avaliação de Soluções de Inteligência Artificial Preditiva para um Cenário de Internet Tátil

Dissertação de Mestrado apresentada à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Angelo Perkusich, D. Sc
Danilo Freire de S. Santos, D. Sc.
Orientadores

Campina Grande, Paraíba, Brasil
©Cleuves Cajé de Carvalho, 24 de setembro de 2020

C331a Carvalho, Cleuves Cajé de.
Avaliação de soluções de inteligência artificial preditiva para um cenário de internet tátil / Cleuves Cajé de Carvalho. - Campina Grande, 2020.
91f. : il. Color.

Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2020.
"Orientação: Prof. Dr. Angelo Perkusich, Prof. Dr. Danilo Freire de Souza Santos".
Referências.

1. Inteligência Artificial. 2. Internet Tátil. 3. Computação de Borda.
I. Perkusich, Angelo. II. Santos, Danilo Freire de Souza. III. Título.

CDU 004.8(043)

Agradecimentos

Agradeço primeiramente a Deus por ter me dado saúde e força para superar todas as adversidades, permitindo assim a realização deste trabalho.

Agradeço a minha família, em especial aos meus pais Antônio Cajé e Maria do Socorro e ao meu irmão Cleiton Cajé, por estarem sempre ao meu lado e não medirem esforços para que eu chegasse até essa etapa da minha vida.

Agradeço aos professores Angelo Perkusich e Danilo Santos pela orientação, confiança e apoio durante a realização desse trabalho.

Agradeço ao Programa de Pós-Graduação em Engenharia Elétrica da UFCG pelo suporte administrativo e também a CAPES pelo suporte financeiro para o desenvolvimento desse trabalho.

Por fim, agradeço aos meus colegas de mestrado Fernanda Famá, Gabriel Daltro, Elvys Raposo e Bruna Sales por todos os momentos de aprendizado, incentivo e descontração.

Resumo

A internet está evoluindo para permitir interações físicas com objetos reais ou virtuais de maneira remota. Esse novo paradigma é conhecido como Internet Tátil, que requer uma latência fim-a-fim de 1 ms. Com o objetivo de diminuir a percepção do atraso em um sistema de internet tátil, abordagens de inteligência artificial preditiva devem ser investigadas. Neste trabalho propõe-se o desenvolvimento de um sistema para simulação e avaliação de abordagens de inteligência artificial preditiva e controle em um cenário de Internet Tátil. Os componentes bases do sistema são apresentados, e detalhes a respeito do fluxo de controle e projeto das aplicações são fornecidas. Além disso, é realizada a modelagem e controle remoto do manipulador adotado como robô escravo. São propostas duas abordagens de controle para o sistema desenvolvido: controle Proporcional-Derivativo (PD) e controle Preditivo Baseado em Modelo, *Model Predictive Control* (MPC). Resultados de simulações experimentais das abordagens de controle (no sistema desenvolvido) são apresentados, validando e enfatizando o desempenho do sistema. O controlador PD apresentou melhor desempenho, visto a sintonia dos seus parâmetros por algoritmo genético. Abordagens de predição são investigadas e uma abordagem de predição com uma rede neural artificial baseada em modelo autorregressivo com entradas exógenas, *Auto-regressive Exogenous* (ARX), é avaliada utilizando o sistema desenvolvido. Resultados adquiridos da simulação comprovam a eficácia da abordagem na predição da resposta do modelo escravo sobre atraso de comunicação no domínio da rede.

Abstract

The internet is evolving to allow physical interactions with real or virtual objects remotely. This new paradigm is known as Tactile Internet, which requires an end-to-end latency of 1 ms. In order to decrease the perception of delay in a tactile internet system, predictive artificial intelligence approaches should be investigated. This work proposes the development of a system for simulation and evaluation of predictive and control artificial intelligence approaches in a Tactile Internet scenario. The basic components of the system are presented, and details regarding the control flow and design of the applications are provided. In addition, modeling and remote control of the manipulator adopted as a slave robot is performed. Two control approaches are proposed for the developed system: Proportional-Derivative (PD) control and Model Predictive Control (MPC). Results of experimental simulations of the control approaches (in the developed system) are presented, validating and emphasizing the performance of the system. The PD controller showed better performance, given the tuning of its parameters by genetic algorithm. Prediction approaches are investigated and a prediction approach based on the Autoregressive Exogenous (ARX) artificial neural network is evaluated using the developed system. Results obtained from the simulation prove the effectiveness of the approach in predicting the response of the slave model on communication delay in the network domain.

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Motivação | 3 |
| 1.2 | Problemática | 4 |
| 1.3 | Objetivos | 4 |
| 1.4 | Organização do Texto | 4 |
| 2 | A Internet Tátil: Visão Geral | 6 |
| 2.1 | Requisitos da Internet Tátil | 7 |
| 2.1.1 | Conectividade Ultra - Responsiva | 8 |
| 2.1.2 | Conectividade Ultra - Confiável | 8 |
| 2.1.3 | Segurança e Privacidade | 9 |
| 2.2 | Desafios Técnicos para a Realização da Internet Tátil | 9 |
| 2.2.1 | Dispositivos Hápticos e Cinemáticos | 9 |
| 2.2.2 | Compressão de Dados Hápticos | 11 |
| 2.2.3 | Inteligência Artificial de Borda | 11 |
| 2.3 | Considerações Finais | 12 |
| 3 | Soluções de Inteligência Artificial Preditiva | 13 |
| 3.1 | Abordagens Baseadas em Funções Matemáticas | 14 |
| 3.2 | Abordagens Baseadas em Modelo | 16 |
| 3.3 | Considerações Finais | 22 |
| 4 | O Manipulador UR5 | 23 |
| 4.1 | Princípios de Modelagem de Manipuladores | 24 |
| 4.2 | A Cinemática do Manipulador | 27 |
| 4.2.1 | Cinemática Direta | 27 |
| 4.2.2 | Cinemática Inversa | 32 |
| 4.3 | A Dinâmica do Manipulador | 39 |
| 4.3.1 | Método de Euler-Lagrange | 39 |
| 4.4 | Considerações Finais | 41 |

| | | |
|----------|---|-----------|
| 5 | Sistema para Investigação de Abordagens de Inteligência Artificial Preditiva | 43 |
| 5.1 | Arquitetura Base | 44 |
| 5.2 | Simulação do Manipulador Robótico | 46 |
| 5.3 | Processos e Interfaces de Comunicação | 46 |
| 5.4 | Considerações Finais | 53 |
| 6 | Abordagens de Controle e Inteligência Artificial Preditiva | 54 |
| 6.1 | Identificação do Sistema | 54 |
| 6.2 | Abordagens de Controle | 58 |
| 6.3 | Abordagens de Predição | 64 |
| 6.4 | Considerações Finais | 71 |
| 7 | Simulações e Resultados | 73 |
| 7.1 | Configuração Experimental | 73 |
| 7.2 | Desempenho das Abordagens de Controle | 74 |
| 7.3 | Desempenho da Abordagem de Predição | 80 |
| 7.4 | Considerações Finais | 83 |
| 8 | Conclusões | 84 |
| 8.1 | Sugestões de Trabalhos Futuros | 86 |
| | Referências Bibliográficas | 87 |

Lista de Siglas

API *Application Programming Interface.*

AR *Autorregresivo.*

ARX *Autoregressive Exogenous.*

CN *Core Network.*

DH *Denavit- Hartenberg.*

DoF *Degree of Freedom.*

GA *Algoritmo Genético.*

H2H *Human to Human.*

H2M/R *Human to Machine/Robot.*

HSI *Human-System Interface.*

IA *Inteligencia Artificial.*

IoT *Internet of Things.*

JND *Just Noticeable Difference.*

JSON *JavaScript Object Notation.*

M2M *Machine to Machine.*

MAE *Mean Absolute Error.*

MEC *Mobile Edge Computing.*

MIMO *Multiple-Input Multiple-Output.*

MLP *Multi-Layer Perceptron.*

MMT *Model-Mediated Teleoperation.*

MOESP *Multivariable Output-Error State Space.*

MPC *Model Predictive Control.*

NFV *Network Function Virtualisation.*

NRMSE *Normalized Root Mean Square Error.*

P *Proporcional.*

PD *Proporcional-Derivativo.*

PI *Proporcional-Integral.*

PID *Proporcional-Integral-Derivativo.*

PRBS *Pseudo-Random Binary Signals.*

QoE *Quality of Experience.*

RAN *Radio Access Network.*

RLS *Recursive Least Squares.*

RMSE *Root Mean Square Error.*

RNA *Rede Neural Artificial.*

ROS *Robot Operating System.*

SDN *Software-Defined Networking.*

SID *Subspace Identification Methods.*

SIMO *Single-Input Multiple-Output.*

TC *Traffic Control.*

TCP *Transmission Control Protocol.*

UDP *User Datagram Protocol.*

URLLC *Ultra-Reliable and Low-Latency Communication.*

Lista de Símbolos

| | |
|--|---|
| $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ | Matriz de Coriolis e de aceleração centrípeta, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ |
| α_i | Ângulo da articulação i , parâmetro de DH |
| β | Fator de ponderação |
| $\dot{\mathbf{q}}(t)$ | Vetor de velocidades das juntas, $\dot{\mathbf{q}}(t) \in \mathbb{R}^n$ |
| $\dot{\mathbf{q}}_m$ | vetor de velocidade das juntas no lado mestre, $\dot{\mathbf{q}}_m \in \mathbb{R}^n$ |
| $\dot{\mathbf{q}}_s$ | Vetor de velocidade das juntas no lado escravo, $\dot{\mathbf{q}}_s \in \mathbb{R}^n$ |
| $\boldsymbol{\omega}_i$ | Vetor de velocidade angular da articulação i , $\boldsymbol{\omega}_i \in \mathbb{R}^3$ |
| $\boldsymbol{\tau}$ | Vetor de torques de entrada, $\boldsymbol{\tau} \in \mathbb{R}^n$ |
| \mathbf{A}_i | Matriz de transformação homogênea que fornece a posição e orientação da coordenada i em relação a coordenada $i - 1$, $\mathbf{A}_i \in \mathbb{R}^{4 \times 4}$ |
| $\mathbf{g}(\mathbf{q})$ | Vetor de gravidade, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ |
| \mathbf{g}_o | Vetor de gravidade da coordenada inercial $o_0x_0y_0z_0$, $\mathbf{g}_o \in \mathbb{R}^3$ |
| \mathbf{I}_i | Tensor de inércia da coordenada i ($o_ix_iz_i$), $\mathbf{I}_i \in \mathbb{R}^3$ |
| $\mathbf{J}_\omega(\mathbf{q})$ | Parte angular do Jacobiano geométrico, $\mathbf{J}_\omega(\mathbf{q}) \in \mathbb{R}^{3 \times n}$ |
| $\mathbf{J}_v(\mathbf{q})$ | Parte linear do Jacobiano geométrico, $\mathbf{J}_v(\mathbf{q}) \in \mathbb{R}^{3 \times n}$ |
| $\mathbf{M}(\mathbf{q})$ | Matriz de inércia, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ |
| \mathbf{q} | Vetor de variáveis das juntas, $\mathbf{q} \in \mathbb{R}^n$ |
| \mathbf{q}_d | Vetor de posição desejada das juntas, $\mathbf{q}_d \in \mathbb{R}^n$ |
| \mathbf{q}_m | Vetor de velocidade das juntas no lado mestre, $\mathbf{q}_m \in \mathbb{R}^n$ |
| \mathbf{q}_s | Vetor de posição das juntas no lado escravo, $\mathbf{q}_s \in \mathbb{R}^n$ |
| $\mathbf{R}_x(\theta)$ | Matriz de rotação de θ graus em torno do eixo x, $\mathbf{R}_x(\theta) \in SO(3)$ |
| $\mathbf{R}_y(\theta)$ | Matriz de rotação de θ graus em torno do eixo y, $\mathbf{R}_y(\theta) \in SO(3)$ |
| $\mathbf{R}_z(\theta)$ | Matriz de rotação de θ graus em torno do eixo z, $\mathbf{R}_z(\theta) \in SO(3)$ |
| \mathbf{R} | Matriz de rotação, $\mathbf{R} \in SO(3)$, $\mathbf{R}^T = \mathbf{R}^{-1}$, $\det \mathbf{R} = 1$ |

| | |
|-------------------|--|
| \mathbf{r}_{ci} | Vetor do centro de massa da articulação i , $\mathbf{r}_{ci} \in \mathbb{R}^3$ |
| \mathbf{T} | Matriz de transformação homogênea, $\mathbf{T} \in \mathbb{R}^{4 \times 4}$ |
| \mathbf{t} | Vetor de translação, $\mathbf{t} \in \mathbb{R}^3$ |
| \mathbf{u}_τ | Vetor dos torques nas juntas, $\mathbf{u}_\tau \in \mathbb{R}^n$ |
| \mathbf{v}_i | Vetor de velocidade linear da articulação i , $\mathbf{v}_i \in \mathbb{R}^3$ |
| \mathbf{X}_d | Vetor da pose desejada do efector final no espaço da tarefa, $\mathbf{X}_d \in \mathbb{R}^6$ |
| ΔI | Intensidade perceptível de um estímulo |
| Δ | Tempo de espera no laço de controle |
| ${}^A\mathbf{p}$ | Vetor que representa a posição do ponto p no sistema de coordenadas A, ${}^A\mathbf{p} \in \mathbb{R}^3$ |
| τ_1 | atraso do sinal de comando no canal de comunicação |
| τ_2 | atraso do sinal de resposta no canal de comunicação |
| θ_i | Ângulo de rotação da junta i , parâmetro de DH |
| ε | Erro relativo percentual |
| a_i | Comprimento da articulação i , parâmetro de DH |
| c_{ijk} | Coefficientes de Christoffel |
| d_i | Deslocamento da articulação i , parâmetro de DH |
| e_{ss} | Erro em estado estacionário |
| I | Intensidade de um estímulo |
| j_i | Junta i |
| K | Energia cinética |
| k | Constante de Weber |
| K_D | Ganho derivativo |
| K_I | Ganho integral |
| K_P | Ganho proporcional |
| L | Lagrangiana |
| l_i | Articulação i |
| m | Massa de um corpo |
| M_p | Sobressinal |
| n | Número de juntas (graus de liberdade) |
| N_p | Horizonte de predição/controlado |

| | |
|---------|---------------------------------------|
| P | Energia potencial |
| q_i | Variável da junta i |
| Q_u | Matriz de ponderação da entrada |
| Q_x | Matriz de ponderação do erro |
| $SO(n)$ | Grupo Ortogonal Especial de ordem n |
| T_d | Tempo derivativo |
| T_i | Tempo integrativo |
| t_r | Tempo de subida |
| t_s | Tempo de acomodação |

Lista de Tabelas

| | | |
|-----|---|----|
| 4.1 | Parâmetros de DH do manipulador UR5. | 30 |
| 7.1 | Valores MAE, RMSE e e_{ss} obtidos para os controladores PD e MPC na junta 1 e nos eixos x, y e z do espaço cartesiano. | 80 |
| 7.2 | Valores MAE, RMSE e ε obtidos para o sistema com preditor baseado em rede neural ARX. | 83 |

Lista de Figuras

| | | |
|-----|---|----|
| 2.1 | Arquitetura funcional da Internet Tátil. | 7 |
| 2.2 | Representação da latência da Internet Tátil. | 8 |
| 3.1 | Ilustração de como a IA preditiva na borda fornece a percepção de um atraso de 1 ms, enquanto a latência devido a comunicações pode ser muito maior. | 14 |
| 3.2 | Diagrama do sistema de teleoperação bilateral com dois preditores lineares. | 15 |
| 3.3 | Representação da abordagem de teleoperação mediada por modelo (MMT). | 16 |
| 3.4 | Representação do sistema de controle com: a) preditor de Smith clássico; b) preditor de Smith alternativo. | 19 |
| 3.5 | Representação do uso de um preditor de Smith para compensar os atrasos em um canal de comunicação. | 20 |
| 3.6 | Representação da estratégia de controle preditivo com RNA recorrente para um sistema de teleoperação de posição-posição não-linear. Onde x_m é a posição do mestre, x_{sd} é a posição desejada do escravo e x_s é a posição do escravo. | 20 |
| 3.7 | representação da estratégia de controle preditivo baseado em RNA utilizando preditor de Smith para um sistema de teleoperação de posição-força. Onde x_m é a posição do mestre, x_{sd} é a posição desejada do escravo, x_s é a posição do escravo e \hat{f}_e é a força de contato estimada no escravo. | 21 |
| 3.8 | Representação da estratégia de controle preditivo baseado em RNA utilizando preditor de Smith para um sistema de teleoperação sem sinal de <i>feedback</i> . Onde x_m é a posição do mestre, x_{sd} é a posição desejada do escravo, x_s é a posição do escravo e \hat{f}_e é a força estimada no mestre. | 22 |
| 4.1 | Imagem do manipulador UR5. | 23 |
| 4.2 | Representação da relação entre os espaços cinemáticos de um manipulador. | 27 |
| 4.3 | Esquemático do UR5, incluindo suas juntas e articulações. | 28 |
| 4.4 | Representação do manipulador UR5 na sua posição inicial. | 31 |
| 4.5 | Ilustração da origem da coordenada da quinta junta. | 33 |

| | | |
|------|---|----|
| 4.6 | Ilustração da vista superior do manipulador para encontrar θ_1 por meio de sua geometria. | 35 |
| 4.7 | Ilustração da vista superior do manipulador para encontrar θ_5 por meio de sua geometria. | 36 |
| 4.8 | Ilustração de ${}^6\hat{\mathbf{Y}}_1$ expresso em coordenadas polares para encontrar θ_6 | 37 |
| 4.9 | Ilustração das juntas 2, 3 e 5 constituindo um manipulador 3R. | 38 |
| 5.1 | Diagrama da estrutura geral do sistema de Internet Tátil com computação na borda. | 44 |
| 5.2 | Diagrama do sistema para investigação de técnicas de inteligência artificial preditiva. | 45 |
| 5.3 | Interface do software de simulação de robôs CoppeliaSim. | 47 |
| 5.4 | Arquitetura de comunicação do sistema. | 47 |
| 5.5 | Diagrama do fluxo de dados entre os componentes do sistema. | 48 |
| 5.6 | Fluxo de controle da aplicação na borda mestre. | 51 |
| 5.7 | Fluxo de controle da aplicação na borda escrava. | 52 |
| 6.1 | Representação do sinal PRBS utilizado como entrada para identificação do sistema. | 56 |
| 6.2 | Diagrama da estrutura geral de controle de manipuladores robóticos. | 58 |
| 6.3 | Diagrama de controle do sistema. | 59 |
| 6.4 | Ganhos do controlador PID gerados pelo algoritmo genético. O ponto preto é a solução ótima encontrada pelo algoritmo genético. | 62 |
| 6.5 | Diagrama do sistema com controle preditivo baseado em modelo. | 63 |
| 6.6 | Diagrama do sistema com preditor de Smith. | 65 |
| 6.7 | Diagrama do sistema com preditor de Smith baseado em Rede Neural Artificial. | 65 |
| 6.8 | Diagrama do sistema com preditor de Smith baseado em Rede Neural sem realimentação. | 66 |
| 6.9 | Modelo de um neurônio artificial. | 68 |
| 6.10 | Arquitetura da rede neural artificial. | 69 |
| 6.11 | Curvas de perdas do treinamento e validação da rede neural ARX para um horizonte de predição igual a 10. | 70 |
| 6.12 | Representação da saída real e saída predita para o conjunto de teste utilizando a rede neural ARX com horizonte de predição igual a 10. | 71 |
| 6.13 | Sistema com o preditor baseado em rede neural ARX. | 71 |
| 7.1 | Diagrama da configuração experimental. | 74 |
| 7.2 | Representação da trajetória realizada pelo manipulador UR5. | 75 |

| | | |
|------|---|----|
| 7.3 | Resposta do sistema ao degrau na junta 1 com o controlador PD | 77 |
| 7.4 | Resposta ao degrau para o controlador MPC | 77 |
| 7.5 | Resposta ao degrau para o controlador PD | 78 |
| 7.6 | Resposta ao degrau para o controlador MPC | 79 |
| 7.7 | Resposta ao degrau para atrasos de 0,7 ms, 100 ms, 200 ms, 400 ms, e 500 ms no canal de comunicação. | 80 |
| 7.8 | Resposta do sistema com preditor baseado em rede neural ARX para a junta 1. | 81 |
| 7.9 | Resposta do sistema com preditor baseado em rede neural ARX para o eixo x. | 82 |
| 7.10 | Resposta do sistema com preditor baseado em rede neural ARX para o eixo y. | 82 |

Capítulo 1

Introdução

O desenvolvimento de novas tecnologias de comunicação móvel tem influenciado e moldado significativamente à sociedade nas últimas décadas. A Internet atual desempenha papel fundamental na economia, educação, saúde, logística e indústria com grande impacto no dia-a-dia das pessoas (SIMSEK; FETTWEIS; I, 2019). A comunicação móvel proporcionou conectividade para várias localidades que antes tinham acesso apenas a Internet fixa, trazendo conectividade e acesso à informação diretamente para as pessoas. A Internet tornou-se ainda mais imersiva nos anos recentes com a Internet das Coisas (*Internet of Things* - IoT), onde “coisas” virtuais ou físicas compartilham informações, normalmente por meio da comunicação sem fio (SACHS et al., 2018).

Com o grande número de dispositivos e serviços que foram criados, os dados cresceram exponencialmente e a necessidade de tornar a informação acessível em todos os lugares levou a criação da Computação na Nuvem (do inglês: *Cloud Computing*), onde as informações são colocadas em um dos inúmeros *data centers* espalhados pelo mundo, tornando a localização física das informações menos significativa (SACHS et al., 2018). Algumas aplicações, no entanto, podem exigir um tempo de resposta muito curto, como por exemplo aplicações para veículos autônomos, ou produzir uma grande quantidade de dados que a Computação na Nuvem não consegue processar com rapidez e fornecer o suporte necessário para tais aplicações. Esse paradigma levou a criação da Computação de Borda (do inglês: *Edge Computing*) que refere-se a tecnologias que permitem que a computação seja realizada na borda da rede. A borda é qualquer recurso de computação e rede entre as fontes de dado e a nuvem, e é desejável que ela esteja próxima a fonte de dados ou “coisas” pois isso trará benefícios como redução do tempo de resposta, redução da quantidade de dados trafegados para a nuvem e até a redução do consumo de energia (SHI et al., 2016).

Atualmente está sendo estudado e moldado a próxima grande inovação da Internet: a Internet Tátil. Quando partes do corpo humano como pele e cabelo entram em contato

com objetos externos ocorre à sensação tátil. A pele que reveste o corpo humano é uma membrana repleta de sensores que provocam no ser humano as sensações táteis que detectam o toque, a pressão e a vibração. A Internet Tátil busca transmitir as sensações táteis em tempo real para o controle remoto de objetos físicos ou virtuais pela Internet (MAIER; EBRAHIMZADEH; CHOWDHURY, 2018).

As comunicações tradicionais tem a capacidade de transmitir a voz e dados com certos níveis de performances o que possibilitou a criação dos sistemas de comunicação *Human to Human* (H2H) e *Machine to Machine* (M2M). Porém, a comunicação entre humano e máquina não foi abrangida no paradigma da Internet tradicional. A Internet Tátil, por sua vez, é centrada em torno da comunicação *Human to Machine/Robot* (H2M/R) (ATEYA et al., 2019), (MAIER; EBRAHIMZADEH; CHOWDHURY, 2018) que, por meio de dispositivos hápticos na borda aliado a equipamentos robóticos e redes de comunicações adequadas para as transmissões dos dados, habilita o transporte das sensações táteis em locais remotos.

Na borda, dispositivos como robôs e equipamentos táteis adequados permitirão que os humanos sintam a sensação de estarem totalmente imersos em um ambiente remotamente. Para alcançar esse objetivo, o robô *soft*¹ articulado têm que evoluir para uma nova categoria de robô, o robô tátil. Com a capacidade de receber informações táteis por meio de sensores e peles artificiais, os robôs táteis buscam se assemelhar aos seres humanos podendo também medir força e pressão. O robô tátil terá grandes vantagens em relação aos robôs *soft* devido a seu *feedback*² tátil para novas técnicas de controle inspirados em humanos, aprendizagem de manipulação e algoritmos de segurança. Aliada a computação em borda e outras tecnologias, os robôs táteis poderão tornar-se uma incorporação física central da Internet Tátil (HADDADIN; JOHANNSMEIERS; LEDEZMA, 2019).

O principal desafio para tornar a Internet Tátil possível de ser realizada é a baixa latência necessária. Isto se faz porque o tempo de reação humana ao toque é de 1 ms (FETTWEIS, 2014). Portanto, a Internet Tátil requer latência fim-a-fim de 1 ms, ou seja, a soma do atraso desde o tempo da aquisição do dado pelo sensor até a resposta do atuador (passando por todas as tecnologias de processamento e transmissão de dados) deve ser no máximo 1 ms para que a qualidade de experiência (*Quality of Experience* - QoE) seja aceitável.

A Internet Tátil tem como principais requisitos a baixa latência e os altos níveis de disponibilidade, confiabilidade e segurança. A quarta geração dos sistemas de comunicações móveis (4G) não abrange os requisitos citados anteriormente, mas o desenvolvimento da quinta geração dos sistemas de comunicação móveis (5G) dará suporte para o estabeleci-

¹Robôs que são flexíveis e buscam se mover e se adaptar ao ambiente assim como os organismos vivos

²No contexto deste trabalho, *feedback* refere-se a resposta a um estímulo ou comando de controle

mento da Internet Tátil (SIMSEK et al., 2016). Entretanto, é importante mencionar que a rede de comunicação 5G é a tecnologia que promete habilitar a Internet Tátil, porém, ela não lida com os requisitos específicos da Internet Tátil como, por exemplo, comunicação de informações hápticas. Todavia, há a sobreposição de requisitos e recursos entre a 5G e a Internet Tátil como conectividade de rede ultra-responsiva e ultra-confiável (BERG et al., 2017).

1.1 Motivação

Grandes esforços estão sendo feitos para que as redes 5G cumpram os requisitos necessários da Internet Tátil. O principal obstáculo é o “desafio de 1 ms”, que ao ser vencido, abrirá caminho para interações em tempo real em redes sem fio, permitindo que pessoas controlem objetos reais e virtuais remotamente sem criar *cybersickness*³. Essa grande inovação terá grande impacto na economia e na sociedade como um todo, revolucionando diversos segmentos como educação, esportes, entretenimento, saúde, automação industrial, mobilidade, realidade virtual ou aumentada, robótica, etc, além de impulsionar o crescimento econômico (FETTWEIS, 2014), (ANTONAKOGLU et al., 2018).

Os atrasos de processamento e codificação não são as únicas barreiras para alcançar 1 ms de latência, a velocidade da luz é um limitante que deve ser levado em conta. Embora os avanços em hardware, protocolos e outras tecnologias sejam importantes para diminuir atrasos fim-a-fim, a velocidade finita da luz definirá o limite final do atraso. Para contornar o limite da velocidade da luz, já que leis físicas não podem ser quebradas, outras técnicas mais avançadas devem ser utilizadas para mudar essa regra, como o uso de mecanismos de Inteligência Artificial (IA) na borda. Por meio de abordagens de inteligência artificial preditivos os comportamentos hápticos/táteis podem ser antecipados para fornecer uma experiência de completa imersão no ambiente remoto para o usuário, aumentando a distância máxima entre o usuário e o ambiente remoto acima da distância máxima permitida pela velocidade da luz (SIMSEK et al., 2016). Portanto, além dos esforços com a 5G, o empenho em outras tecnologias são necessárias para a concretização da Internet Tátil, entre elas pode-se destacar computação de borda, computação na nuvem, aprendizagem de máquina, *Software-Defined Networking* (SDN), *Network Function Virtualisation* (NFV), codificação de rede e IIA preditiva (BERG et al., 2017).

³Desconforto, náusea, tontura ao interagir com um ambiente virtual/remoto

1.2 Problemática

Várias tecnologias estão sendo estudadas com o intuito de diminuir a latência na rede e satisfazer assim os requisitos da Internet Tátil. No entanto, embora a latência na rede possa ser reduzida, ela é restringida pelo meio físico de transporte, como por exemplo, em um sistema ótico pela velocidade finita da luz. Apesar de não ser possível mudar essa condição, propostas baseadas em predição podem ser utilizadas afim de diminuir a percepção da latência para distâncias além daquelas impostas pelas limitações do meio físico de transporte. Portanto, identifica-se a necessidade de avaliar o potencial de abordagens baseadas em predição para diferentes cenários, especialmente para o controle de equipamentos industriais remotamente, e de se propor soluções que resultem na obtenção de melhores resultados do ponto de vista do tempo de resposta, erro de predição e estabilidade. Além disso, considerando esse novo cenário, é de interesse a criação de ambientes específicos para testes dessas abordagens preditivas e soluções de controle.

1.3 Objetivos

Dessa forma, com esse trabalho pretende-se projetar um sistema para investigação e avaliação de abordagens de inteligência artificial preditivas, aplicado a um cenário de Internet Tátil. O objetivo desse sistema é proporcionar um ambiente, com os três domínios característicos da Internet Tátil, onde a latência no domínio da rede possa ser alterada para investigar e validar as abordagens de predição em relação a diminuição da percepção do atraso provocados pela latência no domínio da rede. Para isso, os seguintes objetivos específicos foram definidos:

- Desenvolver um sistema para investigação das abordagens de predição no cenário de Internet Tátil para sistemas robóticos;
- Desenvolver uma estrutura de controle para o sistema;
- Avaliar e discutir abordagens de predição utilizando o sistema desenvolvido.

1.4 Organização do Texto

Este documento está organizado da seguinte forma: neste Capítulo foi apresentada a motivação desta pesquisa, bem como os objetivos a serem alcançados; no Capítulo 2 é apresentada uma visão geral a respeito da Internet Tátil. Está presente neste capítulo a descrição da arquitetura funcional da Internet Tátil e, também, são explanados os requisitos e os desafios técnicos para a sua realização; no Capítulo 3 são descritas soluções de

inteligência artificial preditiva que podem ser empregadas em aplicações de Internet Tátil para diminuir a percepção do atraso de comunicação e manter a estabilidade do sistema; no Capítulo 4 é apresentado o manipulador robótico. As especificações do manipulador são descritas e são abordadas a modelagem cinemática e a modelagem dinâmica; no Capítulo 5 é apresentado o sistema para investigação das abordagens de predição onde é detalhada a arquitetura base do sistema e são descritas a simulação do manipulador robótico virtual, as interfaces de comunicação e os processos do sistema; no Capítulo 6 são apresentadas as estratégias de controle e predição investigadas; no Capítulo 7 são apresentados os resultados obtidos com as simulações das abordagens de controle e predição no sistema desenvolvido; as conclusões e propostas de trabalhos futuros são discutidas no Capítulo 8.

Capítulo 2

A Internet Tátil: Visão Geral

A Internet Tátil, também conhecida como Internet de Habilidades (DOHLER et al., 2017), busca uma mudança no paradigma atual da internet, onde uma rede altamente responsiva, confiável e inteligente permitirá experiências físicas e táteis remotamente em tempo real. A Internet Tátil fornecerá um conjunto de habilidades e experiências, em vez de permitir apenas a entrega de conteúdo e acesso a informação como a Internet permitiu até então (SIMSEK; FETTWEIS; I, 2019).

A interação com objetos remotos necessita da transmissão de dados hápticos. Para isso o controle háptico deve ser habilitado pela internet. Quando se fala em controle háptico, entende-se que o sistema deve fornecer um *feedback* tátil, não somente um *feedback* visual/auditivo como nos sistemas não-hápticos. Esse sentido tátil ocorre bilateralmente, ou seja, ele é sentido tanto no ambiente onde o movimento é imposto, quanto no outro ambiente por meio de uma força de reação.

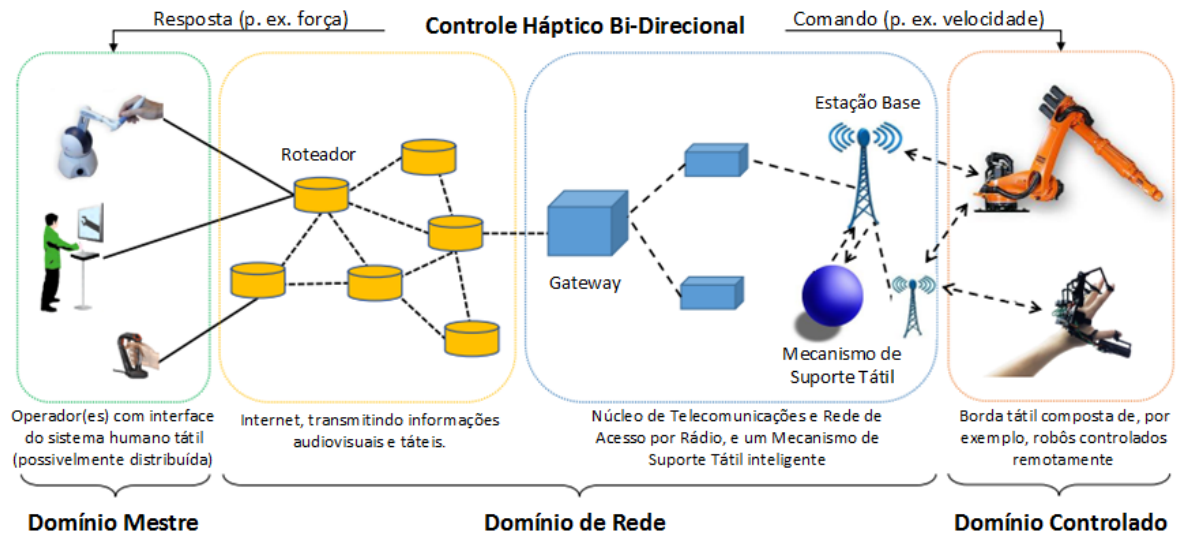
Na Figura 2.1 ilustra-se a arquitetura funcional da Internet Tátil. A arquitetura é composta por três domínios: o domínio mestre, o domínio de rede e o domínio controlado. Os dados hápticos são gerados pela Interface do sistema humano e depois esses dados passam pela rede e seguem para o domínio controlado, onde os dados hápticos são utilizados para controlar um robô ou um algoritmo de interpretação háptica, para no caso de um ambiente virtual (BERG et al., 2017).

O domínio mestre consiste de uma Interface de Sistema Humano (do inglês: *Human-System Interface* ou HSI) com um dispositivo háptico que, por meio de diversas técnicas de codificação tátil, converte as entradas do operador (humano) em dados hápticos. O HSI deve prover a resposta tátil e também disponibilizar a resposta visual e auditiva do domínio controlado.

O domínio de rede é responsável por prover a comunicação bilateral entre os domínios mestre e controlado. Formado por *gateways* e roteadores que transmitem os dados entre os domínios, o domínio de rede deve ser capaz de fornecer conectividade ultra-responsiva

e ultra-confiável (como será abordado na Seção 2.1). Espera-se que a arquitetura da comunicação 5G, composta pela *Radio Access Network* (RAN) e *Core Network* (CN), atendam as principais exigências da Internet Tátil (SIMSEK et al., 2016).

Figura 2.1: Arquitetura funcional da Internet Tátil.



Fonte: Adaptado de Simsek et al. (2016, p.4).

Por fim, o domínio controlado é composto por um robô tátil que interage com objetos no ambiente. Esses robôs são controlados remotamente pelo operador no domínio mestre, e espera-se que o operador sinta como se estivesse totalmente imerso no ambiente remoto por meio da troca de sinais de comando e *feedback* entre os domínios mestre e escravo.

2.1 Requisitos da Internet Tátil

O tempo de reação humana a sensação auditiva, sensação visual e sensação tátil é de 100 ms, 10 ms e 1ms, respectivamente. Portanto, para que haja uma interação em tempo real com um ambiente remotamente, os limites de tempo para cada sensação devem ser alcançados. Dessa forma, os requisitos da Internet Tátil estão diretamente relacionados com o tempo de reação humana para cada sensação.

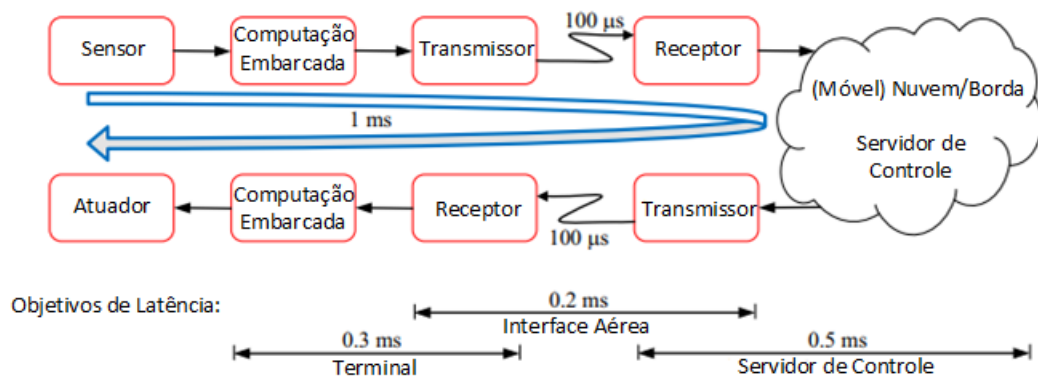
Esta seção trata dos requisitos mais importantes para a efetivação da Internet Tátil. Eles também são necessários para avaliar a QoE do usuário. Os requisitos abordados por (BERG et al., 2017) e (SIMSEK et al., 2016) são listados a seguir.

2.1.1 Conectividade Ultra - Responsiva

O principal desafio da Internet Tátil é obter uma conectividade ultra-responsiva, isso significa obter uma latência de ponta-a-ponta da ordem de 1 ms. A latência de ponta-a-ponta (atraso de ida e volta) diz respeito ao tempo gasto com a transmissão da informação do sensor pela infraestrutura de comunicação para um servidor de controle, com o processamento da informação e, também, com a retransmissão da informação pela estrutura de comunicação de volta ao atuador.

Na Figura 2.2 ilustra-se o objetivo de latência que deve ser alcançado. O sensor mede, pré-processa e fornece os dados para o sistema embarcado, que é responsável por controlar a interface aérea. A interface aérea, por meio de protocolos adequados, transmite os dados para a camada física. Esse mesmo processo acontece no lado do receptor, com uma computação de borda-nuvem, onde os dados são fornecidos a um servidor de controle. Para alcançar o objetivo de latência, pequenos pacotes devem ser transmitidos unidirecionalmente pela camada física em $100 \mu s$.

Figura 2.2: Representação da latência da Internet Tátil.



Fonte: Adaptado de Simsek et al. (2016, p.5).

Caso o atraso seja maior que 1 ms, ele será perceptível pelo usuário, o que significa que a experiência do usuário é afetada negativamente causando efeitos indesejados como *cybersickness*, com sintomas como o enjoo. Há vários fatores que podem contribuir para o atraso como o processamento, o atraso em filas, codificação e transmissão. Porém, com o avanço tecnológico e a 5G, latências de até 1ms serão alcançadas.

2.1.2 Conectividade Ultra - Confiável

Além da baixa latência, o sistema deve ser altamente confiável, pois a recepção incorreta de dados pode resultar em uma resposta incorreta do ambiente remoto, o que pode ser muito prejudicial em aplicações que demandam ações precisas. Para garantir

que os dados sejam enviados corretamente, o sistema deve ser altamente confiável com probabilidade de perdas de 0,001%. Para alcançar essa conectividade ultra-confiável, os protocolos existentes devem ser modificados ou novos protocolos devem ser criados, o que já está sendo pensado na 5G. Os protocolos *Transmission Control Protocol* (TCP) e *User Datagram Protocol* (UDP), por exemplo, levam a instabilidade de atraso, não sendo adequados para aplicações em tempo real.

2.1.3 Segurança e Privacidade

Segurança e privacidade são requisitos importantes para a internet tátil. Muitas aplicações necessitam de alto nível de segurança como é o caso das aplicações médicas. O nível de segurança que temos atualmente entra em conflito com o requisito de latência desejado, ou seja, quanto maior o nível de segurança maior será o atraso de ponta-a-ponta. Portanto, é preciso conciliar a latência com a segurança e privacidade, levando a diversos níveis de segurança em diferentes aplicações envolvendo Internet Tátil.

A segurança deve ser implementada na transmissão física com baixo esforço computacional, devido as restrições na latência. Novas técnicas de codificação devem ser desenvolvidas para que os dados sejam recebidos somente por receptores legítimos. A identificação desses receptores requer que novos métodos sejam criados, como o uso de atributos específicos de hardware (impressões digitais, por exemplo).

2.2 Desafios Técnicos para a Realização da Internet Tátil

Para cumprir os requisitos necessários da Internet Tátil alguns desafios atuais precisam ser vencidos. Desafios que foram parcialmente resolvidos e outros que ainda estão em aberto serão discutidos. Os desafios giram em torno de três áreas principais: digitalização e codificação de dados hápticos; rede ágil e protocolos de comunicação e; inteligência artificial de borda (DOHLER et al., 2017). A seguir, será feito um levantamento a respeito dos dispositivos hápticos e cinemáticos, bem como a compressão dos dados gerados por esses dispositivos. Logo após, a computação de borda e inteligência preditiva serão discutidos.

2.2.1 Dispositivos Hápticos e Cinemáticos

Os dispositivos hápticos e cinemáticos são extremamente importantes para a Internet Tátil como um todo. Primeiramente é preciso fazer a distinção entre dispositivos hápticos e cinemáticos. O primeiro diz respeito a dispositivos que capturam a informação háptica,

ou seja as sensações táteis, já o segundo refere-se a dispositivos que capturam o movimento do operador e replicam o mesmo movimento no robô remotamente operado. Na Internet Tátil os dois tipos de dispositivos devem trabalhar em conjunto para fornecer tanto as informações hápticas quanto as cinemáticas (BERG et al., 2017).

Os dispositivos hápticos dividem-se em sensores hápticos e atuadores hápticos. Os sensores hápticos são responsáveis por coletar as informações táteis, tais como pressão e força, recebido pelo robô tátil no ambiente. Os desafios relacionados ao projeto de sensores táteis são, principalmente, a otimização da resolução espacial, a sensibilidade, o tempo de varredura e posicionamento. Os atuadores hápticos são responsáveis por fornecer o *feedback* háptico no domínio mestre, ou seja, fornecer ao operador a sensação de força/pressão experimentadas pelo robô no ambiente remoto. Há dois tipos de toque dentro do *feedback* háptico, o toque cutâneo e o toque cinestésico. O toque cutâneo refere-se ao toque da pele, com sentidos como pressão, vibração, temperatura e dor, já o toque cinestésico refere-se a sensação da posição das partes vizinhas do corpo. Muitos estudos em relação ao *feedback* háptico de toque cutâneo já foram feitos e experimentados, por exemplo, luvas vibrotáteis que indicam o toque cutâneo. O grande problema das luvas vibrotáteis e dispositivos que não tenha o *feedback* de toque cinestésico é que a percepção de objetos é muito prejudicada, isso leva a sensação de estar se movendo através de objetos, em um sistema teleoperado por exemplo, o operador poderia derrubar objetos, apertá-los com muita força entre outros problemas. As implementações atuais que envolvem o *feedback* cinestésico tem a desvantagem de limitar o espaço do operador devido a uma construção aterrada, ou em casos onde não é aterrada, adiciona peso a luva o que limita o tempo de utilização do usuário em razão do esforço. Portanto, o *feedback* cinestésico ainda é um desafio para a Internet Tátil. O grande objetivo é projetar um sistema que ofereça *feedback* cutâneo e cinestésico, que não limite o espaço e, ao mesmo tempo, seja leve (BERG et al., 2017).

Os dispositivos cinemáticos incluem sensores cinemáticos e atuadores cinemáticos. Existem alguns métodos utilizados em sensores hápticos para capturar o movimento como métodos capacitivos, mecânicos, óticos e acústicos. O grande desafio é rastrear fielmente o movimento do operador, para isso, algoritmos computacionalmente intensivos precisam ser usados, o que aumentaria o tempo de varredura. Além do que, para rastrear o movimento muitos sensores devem ser utilizados, tornando a leitura de dados de cada sensor individual para pós-processamento um empecilho. O atuador cinemático, por sua vez, consiste de um mecanismo computacional que executa um algoritmo de inversão cinemática para reproduzir o movimento e um circuito de acionamento que aciona os atuadores físicos presentes no robô. No caso dos atuadores táteis, os algoritmos de inversão cinemática requerem grande esforço computacional, o que pode aumentar a latência de ponta-a-

ponta. Ainda é preciso estudar o impacto do circuito acionador na latência para casos de uso específico (BERG et al., 2017).

2.2.2 Compressão de Dados Hápticos

A aquisição, armazenamento, transmissão e exibição de informações de áudio e vídeo digitais já alcançaram um nível de qualidade bastante alto, com *codecs* de áudio e vídeo altamente eficientes. No entanto, *codecs* para dados hápticos ainda não foram amplamente explorados e agora, com a Internet Tátil, estão ganhando relevância. Os *codecs* hápticos são responsáveis por realizar a redução dos dados hápticos: informação tátil e informação cinemática (STEINBACH et al., 2018). Devem ser usados *codecs* específicos para dados hápticos para satisfazer o requisito de latência da Internet Tátil. Um método possível para redução dos dados hápticos é a compressão.

Muitos métodos exploram as limitações na percepção dos dados hápticos pelo operador humano para reduzir os dados de forma eficiente. Tais abordagens se baseiam no conceito de faixa-mortas perceptuais, que é uma zona em que as mudanças são tão pequenas que são imperceptíveis aos humanos. Quando um valor do sensor cai nessa zona, o dado não é transmitido. A relação matemática proposta por Ernst Weber, conhecida por diferença apenas perceptível (do inglês: *Just Noticeable Difference* ou JND)

$$\frac{\Delta I}{I} = k \quad (2.1)$$

é a relação entre a intensidade de um estímulo I e a intensidade perceptível de um estímulo ΔI , k é a JND ou constante de Weber (ANTONAKOGLU et al., 2018). No conceito de faixa-mortas perceptuais, os dados que serão enviados definem os limiares da percepção. Os dados que encontram-se dentro desses limites (faixa-morta) são descartados e caso o valor caia fora dessa faixa, esse dado é enviado e ele é utilizado para definir os novos limiares. Dessa forma, os dados que serão enviados são reduzidos (STEINBACH et al., 2010). Em Allin, Matsuoka e Klatzky (2002), dados mostraram um JND médio de aproximadamente 10% para a percepção de força de *feedback* no dedo indicador.

2.2.3 Inteligência Artificial de Borda

Devido a velocidade da luz, um leque de serviços e aplicações seriam limitados a 100 Km, considerando o uso de fibras ópticas. Para superar essa limitação a Internet Tátil deve suportar uma composição híbrida de informações de controle físico e informações de controle preditivo baseado em inteligência. Essas informações de controle devem estar próximas da borda, para que a latência seja reduzida. Portanto, a borda da rede deve ser

equipada com inteligência artificial preditiva, exigindo o desenvolvimento de novas técnicas de inteligência artificial para arquiteturas de borda-nuvem (SIMSEK et al., 2016).

As tecnologias de Computação de Borda são classificadas em três tipos (DOLUI; DATTA, 2017): *Fog Computing*; *Cloudnet Computing*; e *Mobile Edge Computing* (MEC). O MEC é particularmente atraente para a inteligência de borda devido a sua alta percepção de contexto e a sua proximidade com os dispositivos finais, ou seja, ele traz serviços de Computação de Nuvem para mais perto do usuário fazendo com que ele seja acessado com uma latência menor do que os serviços atuais (AIJAZ; SOORIYABANDARA, 2018).

2.3 Considerações Finais

Neste capítulo foi apresentada uma visão geral a respeito da Internet Tátil que propõe uma internet com conectividade inteligente, ultra-responsiva e ultra-confiável. A arquitetura geral da Internet tátil, compostas pelos domínio mestre, domínio de rede e domínio controlado, foi apresentada e detalhada. A internet tátil busca a transmissão remota de experiências táteis físicas e o controle em tempo real. Esse objetivo só pode ser alcançado se o sistema for altamente confiável e entregar uma latência de fim-a-fim na faixa de milissegundos. Os requisitos restritivos da Internet Tátil impõem desafios técnicos em torno da digitalização e codificação de dados hápticos, rede ágil e protocolos de comunicação e inteligência artificial de borda, que precisam ser superados para a sua consolidação.

No Capítulo 3 serão discutidas algumas soluções de inteligência artificial preditiva que aliadas a computação em borda prometem reduzir a percepção do atraso de comunicação mantendo a estabilidade do sistema.

Capítulo 3

Soluções de Inteligência Artificial Preditiva

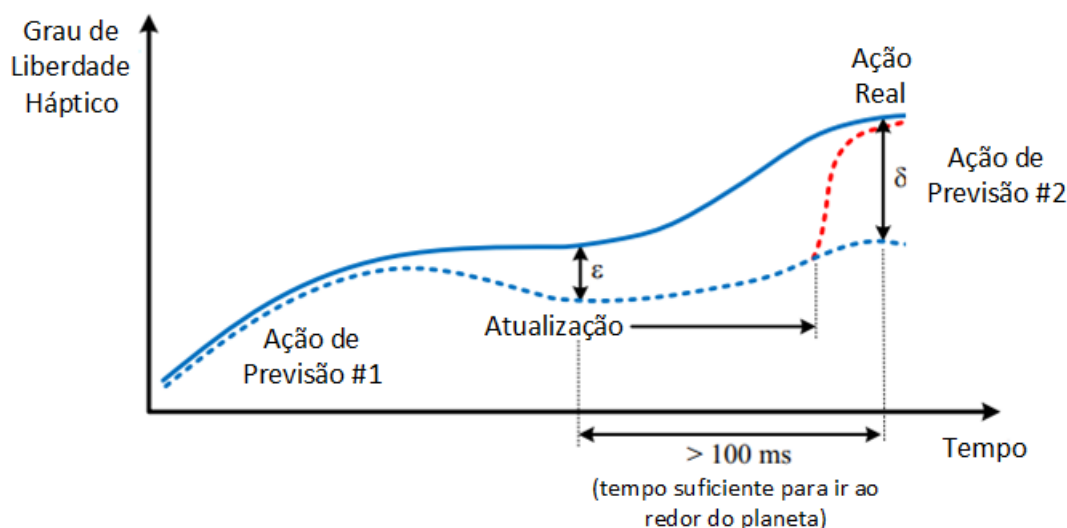
A inteligência artificial preditiva na borda juntamente com as tecnologias de rede, desempenham papel fundamental na redução da percepção do atraso na experiência tátil/háptica. Neste capítulo, soluções de inteligência artificial preditiva serão discutidos como forma de superar as limitações geográficas e físicas, garantir a estabilidade do controle e, também, transparência das aplicações de Internet Tátil.

Um domínio específico da Internet Tátil é a teleoperação que permite a imersão remota, incluindo o toque. Por meio da teleoperação os usuários experimentam uma experiência virtual otimizada. A comunicação háptica, enquanto aplicação da Internet Tátil, usa sistemas de teleoperação para permitir a troca eficiente de informação cinestésica ou táteis, fornecendo ao usuário informações auditivas e visuais sincronizadas (ANTONAKOGLU et al., 2018). Na literatura, muitas técnicas de inteligência artificial preditiva vêm sendo estudadas e implementadas em sistemas de teleoperação. As técnicas empregadas podem ser baseadas em funções matemáticas (livre de modelos) ou podem ser baseadas em modelos. As soluções baseadas em modelo convergem com maior rapidez, entretanto, as abordagens livres de modelo são mais flexíveis e podem ser generalizadas para incluir outras aplicações (AIJAZ; SOORIYABANDARA, 2018). Com algumas alterações em sua estrutura, as técnicas empregadas na teleoperação podem ser estendidas para sistemas de Internet Tátil, por meio da computação de borda e outros recursos. A seguir serão discutidos as soluções de inteligência artificial preditiva baseadas em funções matemáticas e as soluções baseadas em modelo.

3.1 Abordagens Baseadas em Funções Matemáticas

A técnica baseada em função mais básica é o preditor linear que, por meio de uma função linear formada por valores passados recebidos, é capaz de gerar uma previsão do movimento. Essa previsão é facilitada devido aos nossos padrões de movimento que são bastantes repetitivos. Conforme apresentado no Capítulo 2, uma maneira bem simples de utilizar o preditor linear é executando-o apenas no domínio mestre, para que os valores previstos sejam utilizados como *feedback* para o usuário, assim o usuário teria a sensação de uma experiência em tempo real. Outra forma é utilizar a mesma técnica em ambos os lados, ou seja, no domínio mestre e controlado. No ambiente remoto, o preditor gera as mesmas previsões de *feedback* assim como no domínio mestre, o que leva a valores preditos e reais que podem ser comparados e quando o valor divergir um certo nível, esse valor é enviado para o domínio mestre para que sejam feitas atualizações. Esse método fornece uma experiência transparente ao usuário com redução dos dados hápticos que precisam ser enviados (BERG et al., 2017). Na Figura 3.1 ilustra-se as correções que são feitas quando o valor predito se distânciava uma certa quantidade ϵ do valor real, os coeficientes são atualizados e enviados para a outra extremidade antes que ocorram danos (SIMSEK et al., 2016).

Figura 3.1: Ilustração de como a IA preditiva na borda fornece a percepção de um atraso de 1 ms, enquanto a latência devido a comunicações pode ser muito maior.

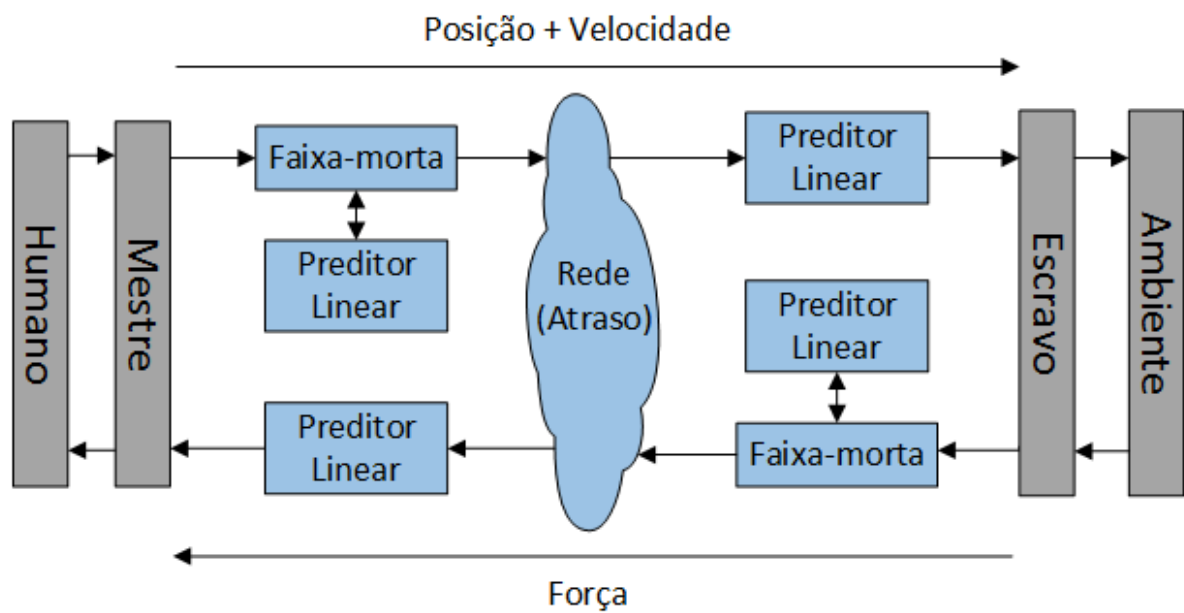


Fonte: Adaptado de Simsek et al. (2016, p.11).

Em Hinterseer et al. (2008), a configuração de um sistema de teleoperação bilateral com dois preditores lineares em ambos os lados da rede é empregada. Para reduzir ainda mais a quantidade de dados enviados, é utilizado também o princípio de faixa-morta

explanado na Seção 2.2.2 do Capítulo 2. Na Figura 3.2 ilustra-se a estrutura do sistema. Os preditores são utilizados para reduzir a taxa de pacotes sem prejudicar a experiência do usuário. As atualizações dos valores são realizadas somente se as mudanças forem perceptíveis para o usuário. Experimentos realizados pelos autores com sistemas de 1 e 3 graus de liberdade (do inglês: *Degree of Freedom* ou DoF) mostraram que a abordagem é capaz de diminuir 90% da taxa de dados iniciais sem afetar o desempenho do sistema.

Figura 3.2: Diagrama do sistema de teleoperação bilateral com dois preditores lineares.



Fonte: Adaptado de Hinterseer et al. (2008, p.4).

Um algoritmo de previsão baseado em modelo Autorregressivo (AR) é proposto em Sakr et al. (2007) para prever as informações de movimento e força em meios hápticos. O algoritmo é genérico e destina-se a aplicação háptica independente do dispositivo. O algoritmo, após inicialização e treinamento, calcula os coeficientes adaptativos do modelo e fornece a previsão dos valores, que dependendo do limiar definido pelo *Just Noticeable Difference* (JND), determina se os valores são atualizados de acordo com os dados reais ou com os valores previstos. Resultados experimentais garantiram que o método é capaz de alcançar excelente precisão com taxa de previsões de movimento e força maiores que 95%.

Em Guo et al. (2011) um modelo de previsão linear baseado em conceito de direção tangencial é utilizado com o intuito de reduzir os dados hápticos. A estratégia de compressão é dependente ainda da reconstrução da curva quadrática para melhorar a precisão do sinal. Um algoritmo de previsão de posição para interfaces hápticas é proposto em Tee et al. (2013) para compensar a latência da rede e perdas de pacotes pela rede. Por sua

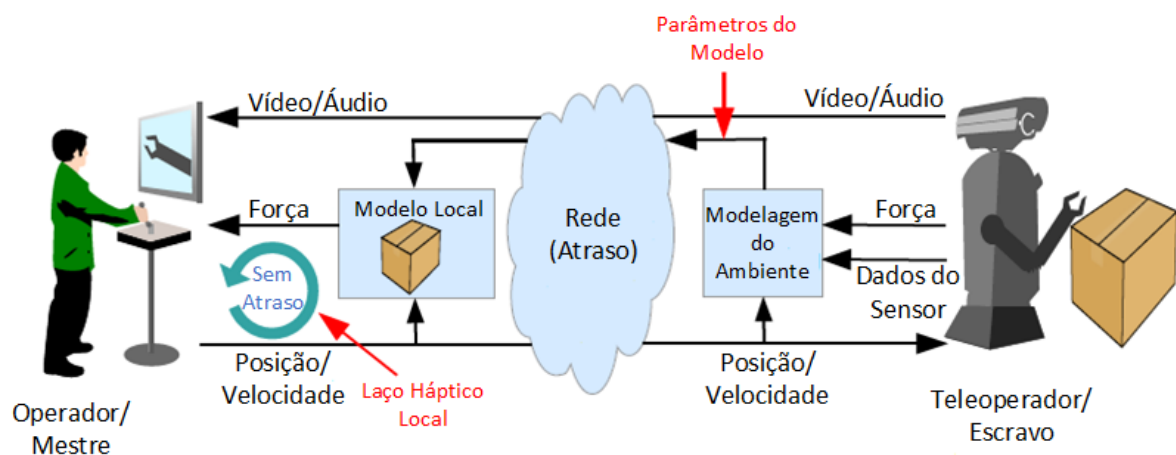
vez, em Brandi e Steinbach (2013) é investigado o uso de regressão linear para melhorar a previsão de amostras na comunicação háptica. Os resultados mostraram que a técnica utilizada melhorou a redução dos dados hápticos com menor distorção do sinal em relação aos preditores lineares de ordem zero e primeira ordem.

Recentemente, um método baseado em observador de estados com uma estrutura em cascata é utilizado em Shen et al. (2019) para previsão de movimento e compensação do atraso em um sistema de teleoperação não-linear. Em Yazdankhoo et al. (2019) é proposta uma abordagem de previsão online baseada em rede neural artificial para sistemas de teleoperação. Já em Zheng et al. (2019) uma abordagem que combina um esquema de predição livre de modelo com um método baseado em modelo é proposto para obter previsões da direção do veículo em um sistema de veículos terrestres teleoperados.

3.2 Abordagens Baseadas em Modelo

Abordagens baseadas em modelos vêm sendo amplamente utilizados em sistemas teleoperados. Na teleoperação mediada por modelo (do inglês: *Model-Mediated Teleoperation* ou MMT), a ideia é usar um modelo do ambiente no lado mestre para fornecer *feedback* estável sem atraso. A configuração geral é ilustrada na Figura 3.3. Os parâmetros do modelo que descrevem o objeto no ambiente escravo são estimados e transmitidos para o lado mestre continuamente em tempo real, ao invés de enviar os sinais hápticos (força, posição) como *feedback*. No lado mestre é mantido um modelo do objeto com base nos parâmetros recebidos e o *feedback* é calculado de acordo com esse modelo local sem que haja atraso. Se o modelo local for preciso com relação ao ambiente remoto, então o MMT pode prover teleoperação estável e transparente (ANTONAKOGLU et al., 2018).

Figura 3.3: Representação da abordagem de teleoperação mediada por modelo (MMT).



Fonte: Adaptado de Antonakoglou et al. (2018, p.15).

Uma abordagem de MMT é estudada e aplicada em um sistema com dois graus de liberdade em Willaert et al. (2012). Sensores de visão, força e posição foram empregados para exploração da geometria do ambiente. Em Uzunoglu e Dede (2017) a MMT foi estudada em um cenário de teleoperação bilateral integrada a um controlador de impedância. Resultados experimentais do estudo com dois dispositivos hápticos indicaram que a estabilidade e a passividade do sistema são garantidas sob atrasos de comunicação variáveis. Já em Song et al. (2018) um novo algoritmo de MMT é proposto para um modelo baseado em força afim de minimizar os efeitos da atualização dos parâmetros do modelo no *feedback* de força e, conseqüentemente, na estabilidade do sistema.

Em um contexto de *healthcare*, um esquema de MMT não-linear é proposto em Liu, Guo e Poignet (2018) para aplicações direcionadas a cirurgia assistida por robô. Utilizando um modelo não-linear de Hunt-Crossley com a estimativa de parâmetros de forma online por meio de técnicas de *Recursive Least Squares* (RLS), simulações mostraram que o sistema é estável, transparente e robusto em meio a variação de atraso na comunicação e também variação nos parâmetros do ambiente.

A ideia por trás do MMT é utilizar um modelo de previsão do ambiente no lado mestre para prover um sistema estável e sem atraso de *feedback*. A visão do uso de previsão do ambiente surgiu em meio ao desenvolvimento do preditor de Smith, que foi pensado originalmente para eliminar a instabilidade causada pelo atraso do sinal de *feedback*. Desde os anos 2000, o estudo de controle preditivo têm sido estudado para sistemas teleoperados com atraso, a maioria deles baseados no preditor de Smith (XU et al., 2016).

O objetivo do preditor de Smith é manter a estabilidade do sistema linear invariante ao tempo na presença de atraso. Na Figura 3.4(a) um sistema de controle baseado em preditor de Smith clássico é ilustrado e pode ser representado também em sua forma alternativa como ilustrado na Figura 3.4(b). No sistema ilustrado na Figura 3.4, $G_p(s)$ é o modelo dinâmico da planta e $G_z(s)$ representa o modelo dinâmico previsto da planta chamado Preditor. Se a variável $Y(s)$ não estiver disponível, a variável $Z(s)$ pode ser utilizada como *feedback*, porém esse modelo não pode controlar a saída $C(s)$ em razão dos distúrbios e modelos imprecisos. Dessa forma, para compensar os erros, um segundo laço de *feedback* deve ser introduzido. Assumindo que o distúrbio $D(s) = 0$, sem o preditor de Smith a função de transferência de malha fechada é dada por

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G_p(s)e^{-sT}}{1 + G_c(s)G_p(s)e^{-sT}} \quad (3.1)$$

o termo de atraso e^{-sT} no denominador indica instabilidade. O uso do preditor de Smith

modifica a função de transferência para

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G_p(s)e^{-sT}}{1 + G_c(s)G_z(s) - G_c(s)G_z(s)e^{-sT} + G_c(s)G_p(s)e^{-sT}} \quad (3.2)$$

se $G_z(s) = G_p(s)$ então

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G_p(s)e^{-sT}}{1 + G_c(s)G_z(s)} \quad (3.3)$$

eliminando o termo de atraso do denominador e, conseqüentemente, a instabilidade do sistema.

Em um sistema teleoperado o preditor de Smith pode ser utilizado, conforme o diagrama da Figura 3.5, para compensar os atrasos no canal de comunicação. A função de transferência do sistema é dada por

$$\frac{C(s)}{R(s)} = \frac{\bar{G}(s)G_p(s)e^{-sT}}{1 + \bar{G}(s)G_p(s)e^{-2sT}} \quad (3.4)$$

seja

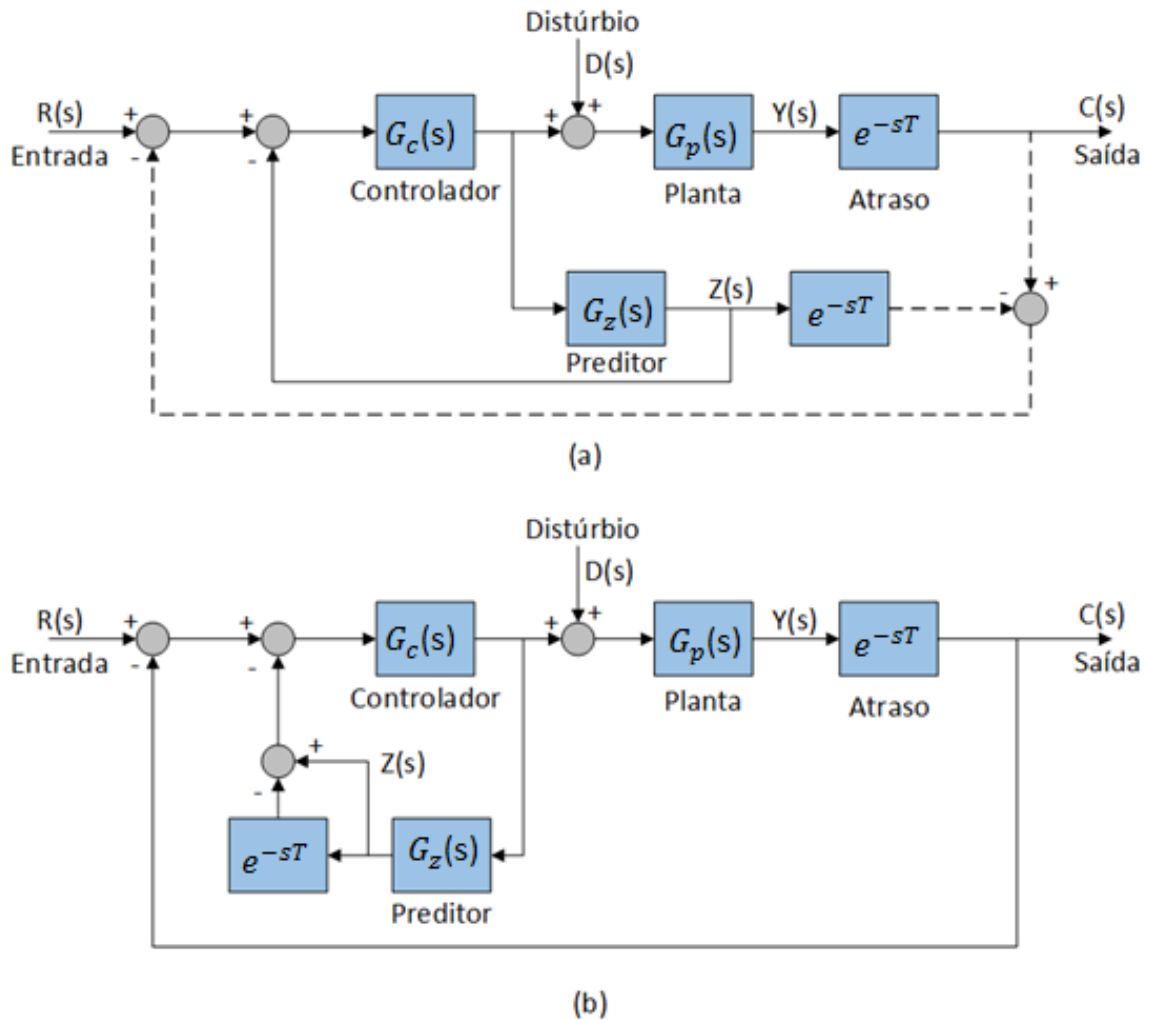
$$\bar{G}(s) = \frac{G_p(s)}{1 + G_p(s)G_z(s)(1 - e^{2sT})} \quad (3.5)$$

se $G_p(s)$ for aproximadamente igual ao modelo da planta, então $G_z(s) \cong G_p(s)$ e a função de transferência torna-se

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G_p(s)e^{-sT}}{1 + G_c(s)G_p(s)}, \quad (3.6)$$

garantindo estabilidade para o sistema em meio ao atraso de comunicação (XU et al., 2016).

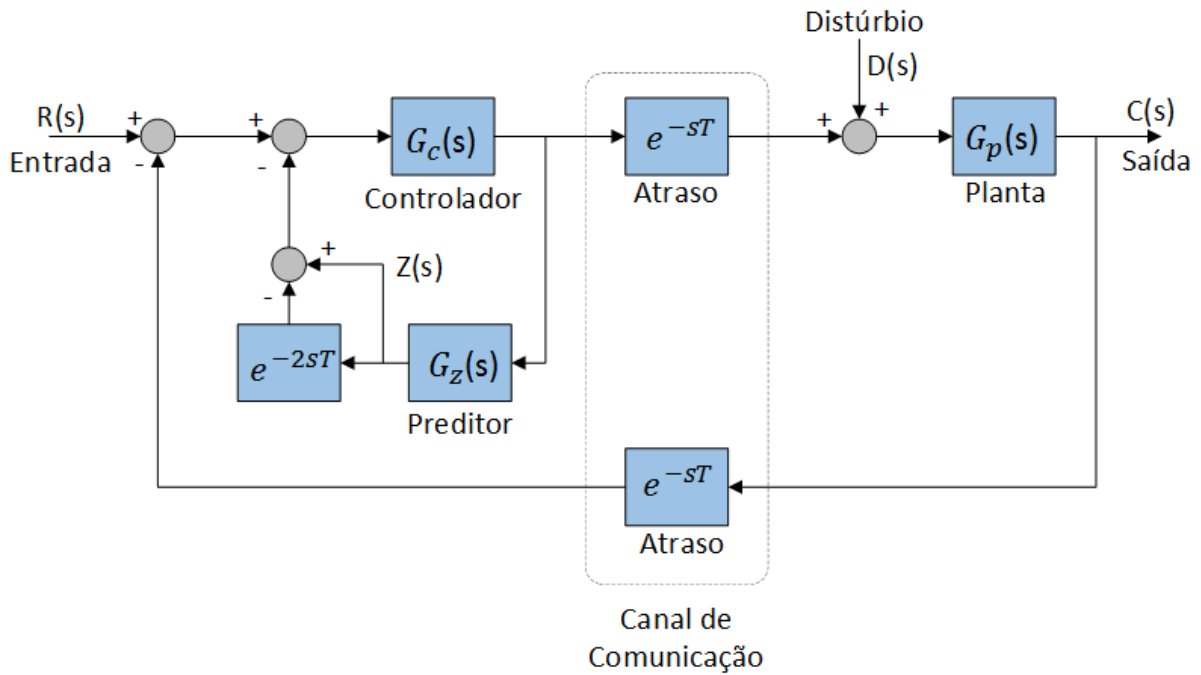
Figura 3.4: Representação do sistema de controle com: a) preditor de Smith clássico; b) preditor de Smith alternativo.



Fonte: Adaptado de Huang e Lewis (2003, p.5)

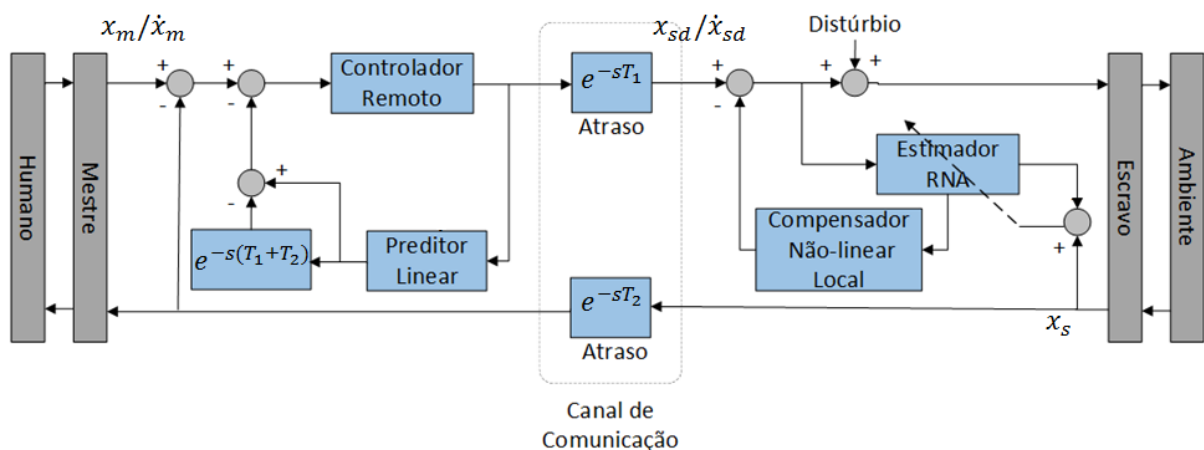
Em Huang e Lewis (2003), uma estrutura de controle preditivo com Rede Neural Artificial (RNA) recorrente é desenvolvida e analisada em um sistema de teleoperação de posição-posição com atraso constante no canal de comunicação. A estrutura de controle consiste de um estimador RNA que modela a dinâmica não-linear do ambiente, com algoritmo de ajuste de peso online. O controlador remoto, no lado mestre, consiste de um preditor de Smith modificado com preditor linear que fornece a previsão e mantém o rastreamento desejado. O preditor linear é projetado a partir do comportamento do ambiente, que assume-se conter uma parte linear que é utilizada no lado mestre para realizar as previsões do comportamento do lado escravo. Na Figura 3.6 ilustra-se o modelo desenvolvido.

Figura 3.5: Representação do uso de um preditor de Smith para compensar os atrasos em um canal de comunicação.



Fonte: Adaptado de Xu et al. (2016, p.4).

Figura 3.6: Representação da estratégia de controle preditivo com RNA recorrente para um sistema de teleoperação de posição-posição não-linear. Onde x_m é a posição do mestre, x_{sd} é a posição desejada do escravo e x_s é a posição do escravo.

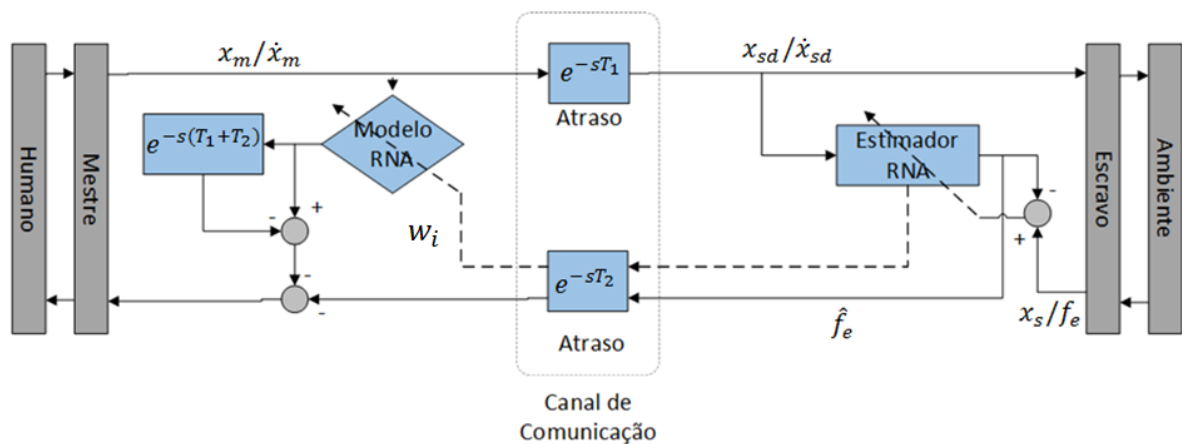


Fonte: Adaptado de Huang e Lewis (2003, p.5).

Em Smith e Hashtrudi-Zaad (2005), outra abordagem baseada em preditor de Smith é utilizada para um sistema de teleoperação de posição-força. O modelo proposto é ilustrado na Figura 3.7. A arquitetura de controle consiste de um esquema de controle preditivo

baseado em uma Rede Neural Artificial (RNA). Um estimador RNA é treinado online para modelar a dinâmica do ambiente. Os pesos w_i da RNA são transmitidos para o lado mestre para a atualização do modelo local. Embora essa atualização seja atrasada, as estimativas convergem para valores estáveis e têm poucas alterações após o período inicial de aprendizado (XU et al., 2016).

Figura 3.7: representação da estratégia de controle preditivo baseado em RNA utilizando preditor de Smith para um sistema de teleoperação de posição-força. Onde x_m é a posição do mestre, x_{sd} é a posição desejada do escravo, x_s é a posição do escravo e \hat{f}_e é a força de contato estimada no escravo.

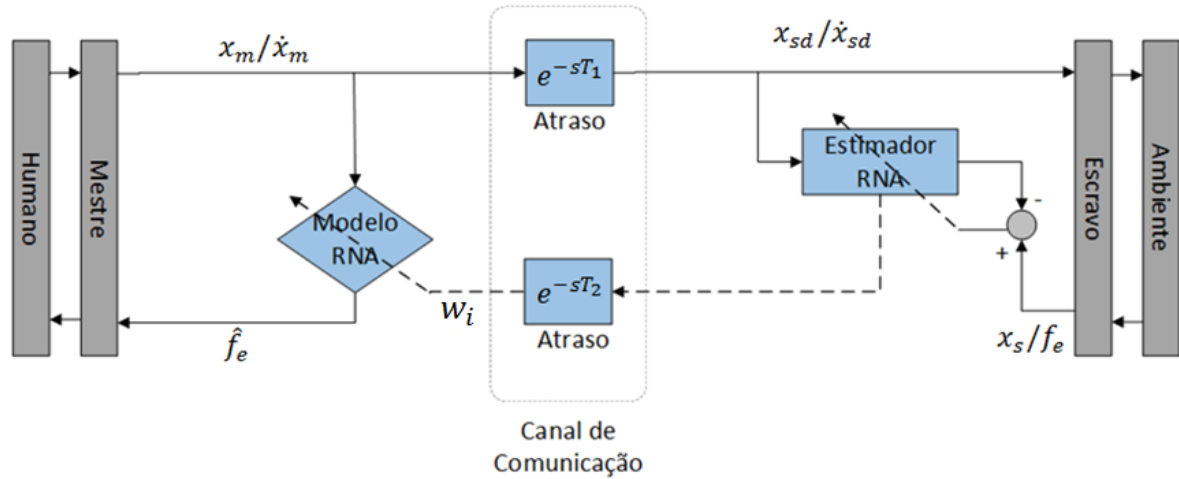


Fonte: Adaptado de Smith e Hashtrudi-Zaad (2005, p.5).

Nos esquemas anteriores, com o uso do preditor de Smith, o atraso de comunicação deve ser conhecido para eliminar o efeito do sinal de *feedback* atrasado. Além do mais, a modelagem imprecisa do ambiente no lado mestre leva a oscilações de contato e dificuldades no controle. Para resolver esses problemas Smith e Hashtrudi-Zaad (2005) desenvolveram uma arquitetura de controle baseada em RNA sem sinal de *feedback* para o lado mestre. O esquema é ilustrado na Figura 3.8. O sinal de força no lado mestre é calculado apenas por meio do modelo local, que é atualizado com as estimativas do lado escravo. Dessa forma, não é necessário conhecer o atraso do canal de comunicação embora a estabilidade do sistema depende muito da precisão do modelo ambiente e dos métodos para atualização do modelo.

A capacidade de lidar com atraso de comunicação desconhecido e com ambientes não lineares e variados, faz com que a estrutura sem sinal de *feedback* sirva de referência para as abordagens baseadas em modelo. De fato, a estrutura ilustrada na Figura 3.8 é conceitualmente idêntica a arquitetura de MMT ilustrada na Figura 3.3.

Figura 3.8: Representação da estratégia de controle preditivo baseado em RNA utilizando preditor de Smith para um sistema de teleoperação sem sinal de *feedback*. Onde x_m é a posição do mestre, x_{sd} é a posição desejada do escravo, x_s é a posição do escravo e \hat{f}_e é a força estimada no mestre.



Fonte: Adaptado de Smith e Hashtrudi-Zaad (2005, p.6).

3.3 Considerações Finais

Neste capítulo foram apresentadas abordagens de inteligência artificial avaliadas e implementadas em sistemas de teleoperação que podem ser estendidas para o domínio da Internet Tátil. As abordagens são utilizadas para diminuir a percepção do atraso de rede, garantir estabilidade do controle e transparência das aplicações. As técnicas empregadas são baseadas em funções matemáticas (livres de modelos) ou baseadas em modelos. As abordagens baseadas em modelos convergem com maior rapidez enquanto as abordagens baseadas em funções são mais abrangentes e podem ser utilizadas em diferentes aplicações.

Em um sistema de Internet Tátil, o domínio controlado consiste de um robô que interage com o ambiente e é controlado remotamente pelo operador no domínio mestre. No Capítulo 4 é apresentado o braço robótico utilizado no âmbito desta pesquisa como robô remotamente controlado: o manipulador serial UR5.

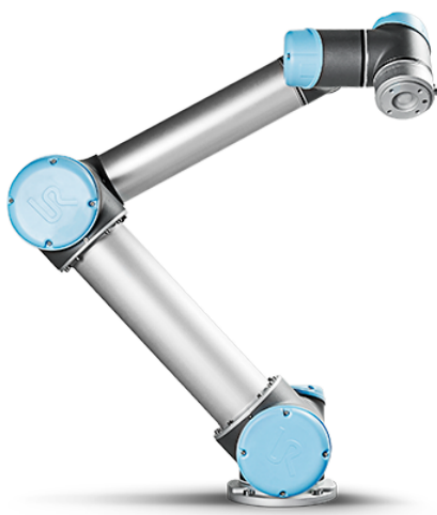
Capítulo 4

O Manipulador UR5

No domínio controlado, o manipulador serial UR5, ilustrado na Figura 4.1, é o robô remotamente controlado escolhido para compor o cenário de aplicação. Consistindo de articulações conectadas por juntas, os manipuladores seriais são braços robóticos capazes de realizar uma variedade de tarefas, por essa razão, são utilizados em muitos sistemas robóticos para, por exemplo, manuseio de materiais, fabricação e teleoperação.

Desenvolvido pela *Universal Robots*¹, o UR5 possui ao todo 6 graus de liberdade. O manipulador foi desenvolvido para ser de fácil programação, leve, flexível e seguro. O UR5 conta com uma capacidade de carga de 5 Kg, um raio de trabalho de 850 mm e uma precisão de 0,1 mm. Além disso, possui seis juntas de revolução com alcance de 360 ° e velocidade máxima de 180 °/s (ROBOTS, 2015).

Figura 4.1: Imagem do manipulador UR5.



Fonte: Kebria et al. (2016, p.2).

¹Empresa dinamarquesa fabricante de braços robóticos e acessórios. Entre outros produtos, a *Universal Robots* fabrica o UR3, UR5, UR10 e recentemente o UR16.

A modelagem cinemática e dinâmica do manipulador UR5 são descritas neste capítulo. Na Seção 4.1 são apresentados os princípios básicos da modelagem de manipuladores robóticos de acordo com (SPONG et al., 2006), (CRAIG, 2009) e (CORKE, 2017), com a descrição da representação de posição e orientação de objetos e transformações entre sistemas de coordenadas. A modelagem cinemática do manipulador é abordada na Seção 4.2, onde é discutida a cinemática direta e, também, a cinemática inversa do manipulador UR5. Em seguida, na Seção 4.3, é apresentada a modelagem dinâmica do manipulador por meio do método de Euler-Lagrange. Por fim, as considerações finais a respeito do capítulo são relatadas na Seção 4.4.

4.1 Princípios de Modelagem de Manipuladores

Os manipuladores robóticos são projetados para realizar movimentos no espaço por algum tipo de mecanismo. Portanto, é de fundamental importância representar a posição e orientação dos objetos rígidos dos manipuladores. Para isso, é necessário definir as coordenadas dos sistemas para representar a posição e orientação dos objetos e também as transformações entre esses sistemas de coordenadas.

Dado um sistema de coordenadas, a posição de qualquer ponto no espaço pode ser localizado por um vetor posição 3×1 . Um sistema de coordenadas cartesianas é definido pelos eixos x , y e z , que são três vetores unitários mutualmente ortogonais. Dado um sistema de coordenadas A , um ponto p em A é representado como um vetor ${}^A\mathbf{p}$ com elementos individuais em x , y e z :

$${}^A\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}. \quad (4.1)$$

O ponto p expresso no sistema de coordenadas A pode ser expresso em um sistema de coordenadas B , quando B tem a mesma orientação que A , por meio de uma adição vetorial. Neste caso, B difere de A por uma translação, dada pelo vetor ${}^B\mathbf{p}_A$ que localiza a origem de A com relação à B . Logo, o ponto p em B pode ser descrito por:

$${}^B\mathbf{p} = {}^A\mathbf{p} + {}^B\mathbf{p}_A \quad (4.2)$$

Para descrever a orientação de um corpo, primeiramente é necessário anexar um sistema de coordenadas ao corpo e, então, fornecer a descrição desse sistema de coordenadas em relação ao sistema de coordenadas de referência. Dado um corpo com orientação representado pela coordenada B , a descrição desse corpo na coordenada de referência A

é dada pela matriz de rotação ortonormal ${}^A_B\mathbf{R} \in SO(3)$ (grupo Ortogonal Especial de ordem três), onde

$${}^A_B\mathbf{R} = [{}^A\hat{\mathbf{X}}_B \quad {}^A\hat{\mathbf{Y}}_B \quad {}^A\hat{\mathbf{Z}}_B] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (4.3)$$

Em síntese, três vetores devem ser definidos para especificar a orientação. As escalares em 4.3 podem ser encontradas por meio da projeção dos vetores na coordenada de referência, logo:

$${}^A_B\mathbf{R} = [{}^A\hat{\mathbf{X}}_B \quad {}^A\hat{\mathbf{Y}}_B \quad {}^A\hat{\mathbf{Z}}_B] = \begin{bmatrix} \hat{\mathbf{X}}_B \cdot \hat{\mathbf{X}}_A & \hat{\mathbf{Y}}_B \cdot \hat{\mathbf{X}}_A & \hat{\mathbf{Z}}_B \cdot \hat{\mathbf{X}}_A \\ \hat{\mathbf{X}}_B \cdot \hat{\mathbf{Y}}_A & \hat{\mathbf{Y}}_B \cdot \hat{\mathbf{Y}}_A & \hat{\mathbf{Z}}_B \cdot \hat{\mathbf{Y}}_A \\ \hat{\mathbf{X}}_B \cdot \hat{\mathbf{Z}}_A & \hat{\mathbf{Y}}_B \cdot \hat{\mathbf{Z}}_A & \hat{\mathbf{Z}}_B \cdot \hat{\mathbf{Z}}_A \end{bmatrix}. \quad (4.4)$$

O produto escalar de dois vetores unitários produz o cosseno do ângulo entre eles, por essa razão, os componentes da matriz de rotação são frequentemente chamados de cossenos de direção. Rotações de θ em torno dos eixos x, y e z podem ser obtidas por meio das matrizes de rotações:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}, \quad (4.5)$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \quad (4.6)$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.7)$$

Qualquer rotação pode ser representada por não mais que três rotações em torno dos eixos de coordenadas, segundo o teorema de rotação de Euler, o que significa que qualquer rotação arbitrária pode ser decomposta em três ângulos de rotação associados aos eixos. O teorema de rotação de Euler requer três rotações sucessivas, mas não duas rotações sucessivas sobre o mesmo eixo. As rotações podem envolver rotações repetidas, mas não sucessivas, sobre um mesmo eixo (XYX, XZX, YXY, YZY, ZXZ, ou ZYZ) ou rotações em torno dos três eixos (XYZ, XZY, YZX, YXZ, ZXY, ou ZYX). Todas essas 12 sequências são chamadas de ângulos de Euler.

A combinação de translação com rotação pode ser obtida por meio de transformações homogêneas. As transformações homogêneas representam o movimento de um corpo

rígido pelas matrizes de transformações da seguinte forma:

$$\mathbf{T}_{4 \times 4} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (4.8)$$

onde $\mathbf{R} \in SO(3)$ é a matriz de rotação, $\mathbf{t} \in \mathbb{R}^3$ é o vetor de translação e $\mathbf{T} \in \mathbb{R}^{4 \times 4}$ é a matriz de transformação homogênea. Dado um corpo representado em um plano cartesiano B, para representar a rotação e translação desse corpo no plano cartesiano A, utiliza-se a transformação da seguinte forma:

$$\begin{bmatrix} {}^A p_x \\ {}^A p_y \\ {}^A p_z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_B \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^B p_x \\ {}^B p_y \\ {}^B p_z \\ 1 \end{bmatrix}. \quad (4.9)$$

A representação homogênea em 4.8 é um caso especial das coordenadas homogêneas que têm sido bastante utilizadas em computação gráfica. A transformação homogênea mais ampla tem a seguinte forma:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ \mathbf{f}_{1 \times 3} & \mathbf{s}_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotação} & \text{Translação} \\ \text{Perspectiva} & \text{Fator de escala} \end{bmatrix}. \quad (4.10)$$

Para este campo de estudo, o último vetor tem sempre a forma “[0 0 0 1]”, mas a forma mais geral em 4.10 é útil em outros campos como, por exemplo, simulação gráfica. Usando o fato que \mathbf{R} é ortogonal, então a inversa da matriz de transformação é dada por

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}_{3 \times 3}^T & -\mathbf{R}_{3 \times 3}^T \mathbf{t}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (4.11)$$

Para calcular transformações subsequentes, tem-se que multiplicar as matrizes de transformações, ou seja

$${}^i_j \mathbf{T} = {}^{i+1}_i \mathbf{T} \dots {}^{j-1}_{j-2} \mathbf{T}. \quad (4.12)$$

A parte rotacional é dada por

$${}^i_j \mathbf{R} = {}^{i+1}_i \mathbf{R} \dots {}^{j-1}_{j-2} \mathbf{R}, \quad (4.13)$$

e o vetor de translação é dado por

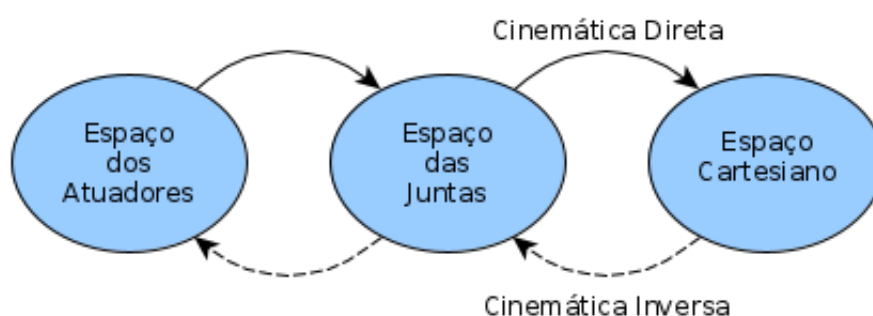
$${}^i_j \mathbf{t} = {}^{i+1}_i \mathbf{t} + {}^{i+1}_i \mathbf{R} \cdot {}^{i+2}_{i+1} \mathbf{t}. \quad (4.14)$$

4.2 A Cinemática do Manipulador

Segundo (CRAIG, 2009), a cinemática estuda o movimento de um corpo sem levar em consideração as forças que causam esse movimento. A posição, velocidade, aceleração e todas as derivadas de ordem superior da posição (em relação ao tempo ou outras variáveis) são estudadas no domínio da cinemática. Logo, as propriedades geométricas e temporais do movimento são objetos de estudo da cinemática dos manipuladores.

Existem três representações para descrever a posição e orientação de manipuladores: descrição no espaço do atuador, no espaço das juntas e no espaço cartesiano (referido também como espaço da tarefa ou espaço operacional). Em muitos manipuladores, cada junta cinemática pode ser atuada por uma combinação de atuadores, não sendo atuada diretamente por algum tipo de atuador. Dessa forma, cada robô têm uma correspondência entre a posição do atuador e a posição da junta, sendo necessário o mapeamento entre o espaço das juntas e o espaço dos atuadores em muitos manipuladores. No domínio da cinemática, o mapeamento entre o espaço das juntas e o espaço cartesiano pode ser feita pela cinemática direta, para obter a posição e orientação do efetor-final do manipulador dado os valores das variáveis das juntas, e pela cinemática inversa, que ao contrário da cinemática direta, busca obter os valores das variáveis das juntas dada a posição e orientação do efetor-final do manipulador. Na figura 4.2 ilustra-se os espaços para representação dos manipuladores e seu mapeamento.

Figura 4.2: Representação da relação entre os espaços cinemáticos de um manipulador.



Fonte: Elaborada pelo autor.

4.2.1 Cinemática Direta

Como descrito anteriormente, um manipulador é composto por uma série de articulações ligadas por juntas. As juntas podem ser do tipo simples com um grau de liberdade apenas, é o caso das juntas de revolução e das juntas prismáticas, ou mais complexas com

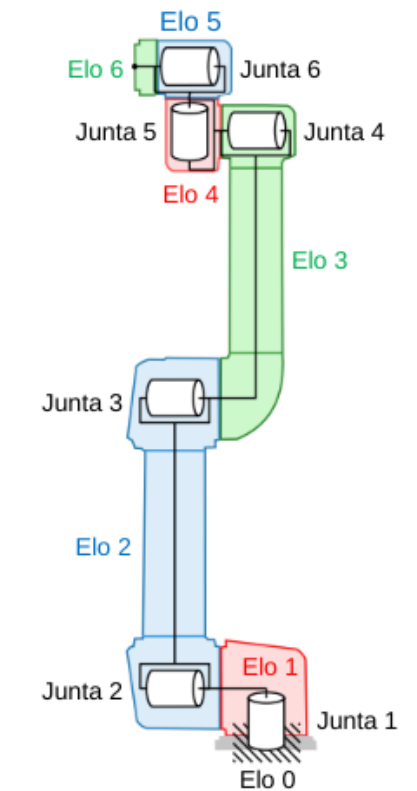
mais de um grau de liberdade, como as juntas esféricas por exemplo. Os manipuladores, favorecidos pelas considerações de projeto mecânico, geralmente possuem juntas que exibem um grau de liberdade. As juntas de revolução permitem uma rotação relativa em torno de um único eixo, já as juntas prismáticas permitem um movimento linear (extensão ou retração) em torno de um único eixo.

Desde que uma junta conecte duas articulações, um manipulador robótico com n juntas possui $n + 1$ articulações. A junta $j_i: i \in \{1, \dots, n\}$ conecta a articulação l_{i-1} a articulação $l_i: i \in \{0, \dots, n\}$. A variável da junta, denotada por q_i (sendo i a i -ésima junta), representa o ângulo de rotação da junta, para o caso de uma junta de revolução, ou o deslocamento da junta, para o caso de uma junta prismática:

$$q_i = \begin{cases} \theta_i & : \text{junta de revolução } i \\ d_i & : \text{junta prismática } i \end{cases}. \quad (4.15)$$

O manipulador robótico UR5 possui seis juntas de revolução com um grau de liberdade cada e, conseqüentemente, sete articulações. Na Figura 4.3 ilustra-se o esboço do UR5 juntamente com a localização das juntas e articulações.

Figura 4.3: Esquemático do UR5, incluindo suas juntas e articulações.



Fonte: Adaptado de Kufieta (2014, p.69)

A cinemática direta do manipulador, ou seja, a posição e orientação do efector-final do manipulador em função do ângulo de rotação das juntas, pode ser descrita pela matriz de transformação homogênea do efector-final com relação a base de referência. Como visto em 4.12, a matriz de transformação homogênea pode ser dividida em uma cadeia de transformações, uma para cada articulação. Logo, para o UR5 temos que:

$${}^0\mathbf{T} = {}^0\mathbf{T} \cdot {}^1\mathbf{T} \cdot {}^2\mathbf{T} \cdot {}^3\mathbf{T} \cdot {}^4\mathbf{T} \cdot {}^5\mathbf{T}. \quad (4.16)$$

No entanto, é possível simplificar e racionalizar a cinemática direta por meio da introdução de outras convenções, como a representação de Denavit-Hartenberg que será descrita a seguir.

Representação de Denavit-Hartenberg

Embora seja possível definir uma coordenada de referência para cada articulação e descrever a relação entre essas coordenadas por meio de transformações homogêneas, geralmente são necessários três parâmetros para definir a rotação e mais três parâmetros para definir a translação, somando assim seis parâmetros para definir o movimento rígido. Com a representação de Denavit-Hartenberg (DH), o número de parâmetros passa de seis para quatro, isso graças a escolha da origem e dos eixos de coordenadas e também devido a exploração da geometria do manipulador. Nessa representação, cada transformação homogênea é definida pela multiplicação de quatro transformações

$$\begin{aligned} \mathbf{A}_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (4.17) \\ &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

onde os quatro parâmetros θ_i , a_i , d_i , α_i são, respectivamente, o ângulo de rotação da junta, o comprimento da articulação, o deslocamento da articulação e o ângulo da articulação. A matriz \mathbf{A}_i é uma função da variável da junta, θ_i para juntas de revolução (caso do manipulador UR5) e d_i para juntas prismáticas, os outros três parâmetros são constantes.

De acordo com a representação de DH, a coordenada $o_i x_i y_i z_i$ associada com a junta

j_i e articulação l_i deve ter os seguintes atributos (GRAVD AHL, 2014):

1. O eixo z_i é o eixo de revolução da junta j_{i+1} ;
2. O eixo x_i é perpendicular aos eixos z_{i-1} e z_i ;
3. O eixo z_i cruza o eixo z_{i-1} ;
4. Todas as coordenadas são destros.

Após a configuração de todas as coordenadas, uma tabela com os parâmetros de DH deve ser estabelecida de acordo com as seguintes convenções:

- a_i - distancia ao longo de x_i da intersecção dos eixos x_i e z_{i-1} até o_i ;
- d_i - distancia ao longo de z_{i-1} de o_{i-1} até a intersecção dos eixos x_i e z_{i-1} ;
- α_i - o ângulo de z_{i-1} a z_i medido sobre x_i ;
- θ_i - o ângulo de x_{i-1} a x_i medido em torno de z_{i-1} .

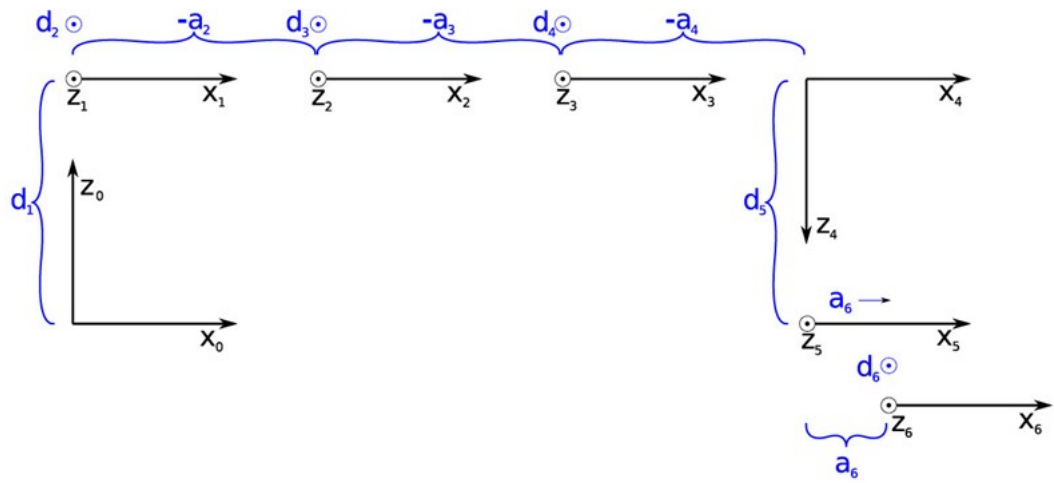
A estrutura cinemática do UR5 na posição inicial é ilustrada na Figura 4.4.

Os parâmetros de DH para o manipulador UR5, para as coordenadas das juntas da Figura 4.4, são fornecidos pela *Universal Robots*, e são apresentados na Tabela 4.1.

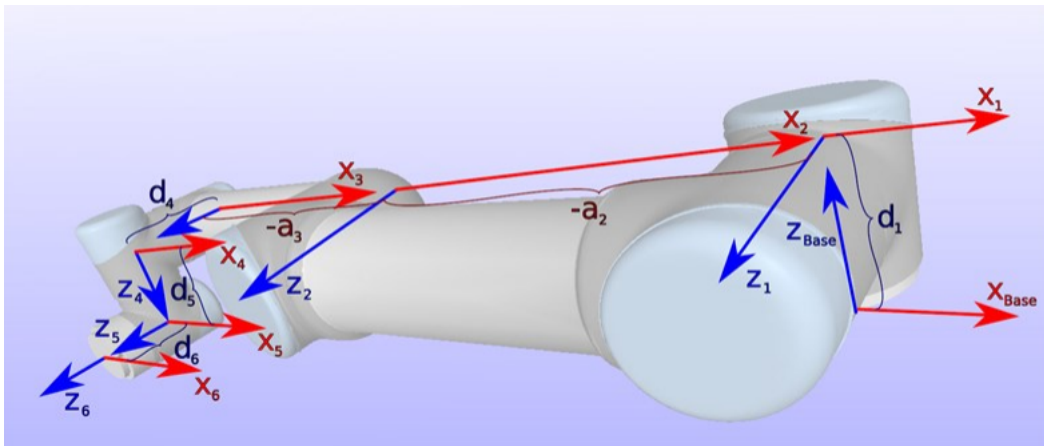
Tabela 4.1: Parâmetros de DH do manipulador UR5.

| i | θ_i (rad) | a_i (m) | d_i (m) | α_i (rad) |
|---|------------------|-----------|-----------|------------------|
| 1 | θ_1 | 0 | 0,089159 | pi/2 |
| 2 | θ_2 | -0,425 | 0 | 0 |
| 3 | θ_3 | -0,39225 | 0 | 0 |
| 4 | θ_4 | 0 | 0,10915 | pi/2 |
| 5 | θ_5 | 0 | 0,09465 | -pi/2 |
| 6 | θ_6 | 0 | 0,0823 | 0 |

Figura 4.4: Representação do manipulador UR5 na sua posição inicial.



(a) Coordenadas do manipulador UR5



(b) Vista da posição do UR5

Fonte: Robots (2020).

Para obter a cinemática direta do UR5, utiliza-se uma cadeia com as matrizes de transformações homogêneas \mathbf{A}_i para cada articulação i , de acordo com a Equação 4.16. Para isso, os parâmetros de DH do UR5, apresentados na Tabela 4.1, são utilizados de acordo com a Equação 4.17, para obter a matriz de transformação homogênea do efetor-final, ou seja:

$${}^0\mathbf{T} = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3 \cdot \mathbf{A}_4 \cdot \mathbf{A}_5 \cdot \mathbf{A}_6. \quad (4.18)$$

Com isso, é obtido a orientação e posição do efetor-final do UR5 e, conseqüentemente, o problema de cinemática direta é solucionado.

4.2.2 Cinemática Inversa

Na Seção 4.2.1 foi discutida a cinemática direta, que busca mapear o espaço das juntas no espaço cartesiano. Nesta seção, a cinemática inversa, ou seja, o mapeamento do espaço cartesiano para o espaço da junta será discutida. O problema da cinemática direta é simples de resolver, pois a matriz de transformação homogênea depende apenas dos ângulos de rotação das juntas e de uma série de parâmetros conhecidos, o que faz com que a solução seja única. O problema da cinemática inversa, por sua vez, é bem mais complicado, principalmente para um manipulador robótico com seis graus de liberdade como o UR5. Dado o valor numérico da posição e da orientação do efector-final com relação a base, ou seja, a matriz ${}^0_n\mathbf{T}$, encontrar os valores das variáveis das juntas envolve sistemas de equações não lineares onde as soluções podem não existir ou podem existir múltiplas soluções (GIER, 2015).

Para resolver o problema da cinemática inversa de um manipulador, pode-se utilizar soluções numéricas ou soluções analíticas (forma fechada). Soluções numéricas consistem em técnicas iterativas, como por exemplo a técnica iterativa de Newton-Raphson, inversa do Jacobiano, transposta do Jacobiano, pseudo-inversa do Jacobiano e método dos mínimos quadrados amortecidos (KALASARIYA; PATEL; THAKKAR,). Todos os métodos iterativos citados anteriormente são baseados no Jacobiano e, devido as iterações e otimizações, são computacionalmente mais caras e também são soluções aproximadas da solução real. As soluções analíticas, por sua vez, fazendo o uso das propriedades geométricas do manipulador, fornecem as soluções exatas resolvendo o problema de cinemática inversa de forma direta. A desvantagem de utilizar um método analítico para resolver o problema de cinemática direta é que ele não deve ser aplicado em diferentes manipuladores sem alterar a sua forma fechada.

Embora não seja uma solução que abrange outros manipuladores, uma abordagem analítica baseada na geometria do manipulador robótico UR5 é utilizado nesse trabalho, principalmente por ser um método mais rápido e preciso em relação a uma solução baseada no Jacobiano. A seguir é descrita a cinemática inversa do UR5 em sua forma fechada de acordo com (HAWKINS, 2013) e (ANDERSEN, 2018).

Cinemática Inversa para o UR5

A cinemática inversa busca encontrar os ângulos de rotações das juntas ($\theta_1 \dots \theta_6 \in [0, 2\pi)^6$) de acordo com a posição e orientação do efector-final, que é especificado pela

matriz de transformação homogênea ${}^0\mathbf{T}$ da forma:

$${}^0\mathbf{T}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = \begin{bmatrix} {}^0\hat{\mathbf{X}}_6 & {}^0\hat{\mathbf{Y}}_6 & {}^0\hat{\mathbf{Z}}_6 & {}^0\mathbf{P}_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.19)$$

$$= \begin{bmatrix} {}^0\hat{X}_{6x} & {}^0\hat{Y}_{6x} & {}^0\hat{Z}_{6x} & {}^0P_{6x} \\ {}^0\hat{X}_{6y} & {}^0\hat{Y}_{6y} & {}^0\hat{Z}_{6y} & {}^0P_{6y} \\ {}^0\hat{X}_{6z} & {}^0\hat{Y}_{6z} & {}^0\hat{Z}_{6z} & {}^0P_{6z} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.20)$$

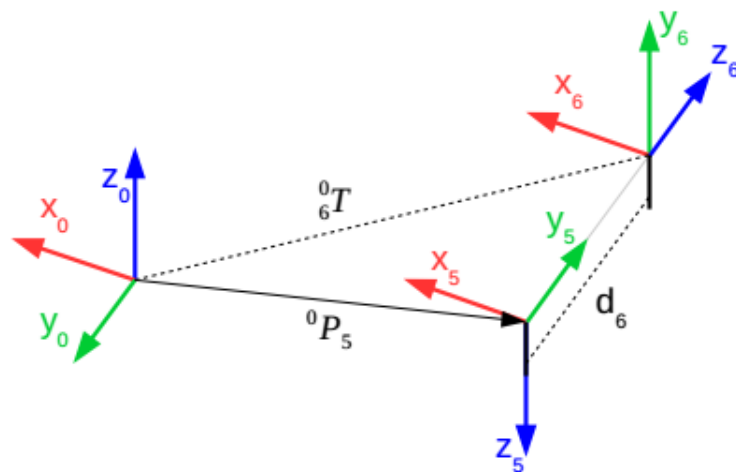
Primeiramente encontra-se θ_1 utilizando o vetor translação da coordenada da quinta junta em relação a coordenada de referência, ${}^0\mathbf{P}_5$. Para encontrar ${}^0\mathbf{P}_5$, pode-se realizar a regressão da coordenada da sexta junta para a coordenada da quinta junta ao longo de z_6 , como ilustrado na Figura 4.5.

Como ${}^0\mathbf{T}$ e d_6 são conhecidos, a translação ${}^0\mathbf{P}_5$ é dada por:

$${}^0\mathbf{P}_5 = {}^0\mathbf{P}_6 - d_6 \cdot {}^0\hat{\mathbf{Z}}_6 \quad (4.21)$$

$$= {}^0\mathbf{T} \cdot \begin{bmatrix} 0 \\ 0 \\ -d_6 \\ 1 \end{bmatrix} \quad (4.22)$$

Figura 4.5: Ilustração da origem da coordenada da quinta junta.



Fonte: Andersen (2018, p.4).

A vista superior do UR5 é utilizada para encontrar θ_1 , como ilustrado na Figura 4.6. A rotação da coordenada de referência para a coordenada da primeira junta é igual a θ_1

que é dado geometricamente por:

$$\theta_1 = \phi_1 + \left(\phi_2 + \frac{\pi}{2} \right). \quad (4.23)$$

O ângulo ϕ_1 pode ser encontrado analisando o triângulo cujo catetos são ${}^0P_{5x}$ e ${}^0P_{5y}$:

$$\phi_1 = \arctan({}^0P_{5y}, {}^0P_{5x}). \quad (4.24)$$

O ângulo ϕ_2 pode ser encontrado por meio do triângulo cujo os catetos são $|{}^0P_{5xy}|$ e d_4 da seguinte forma:

$$\cos \phi_2 = \frac{d_4}{|{}^0P_{5xy}|} \quad (4.25)$$

$$\phi_2 = \pm \arccos \left(\frac{d_4}{|{}^0P_{5xy}|} \right) \quad (4.26)$$

$$\phi_2 = \pm \arccos \left(\frac{d_4}{\sqrt{{}^0P_{5x}^2 + {}^0P_{5y}^2}} \right) \quad (4.27)$$

Dessa forma, o ângulo θ_1 é dado por:

$$\theta_1 = \arctan({}^0P_{5y}, {}^0P_{5x}) \pm \arccos \left(\frac{d_4}{\sqrt{{}^0P_{5x}^2 + {}^0P_{5y}^2}} \right) + \frac{\pi}{2}. \quad (4.28)$$

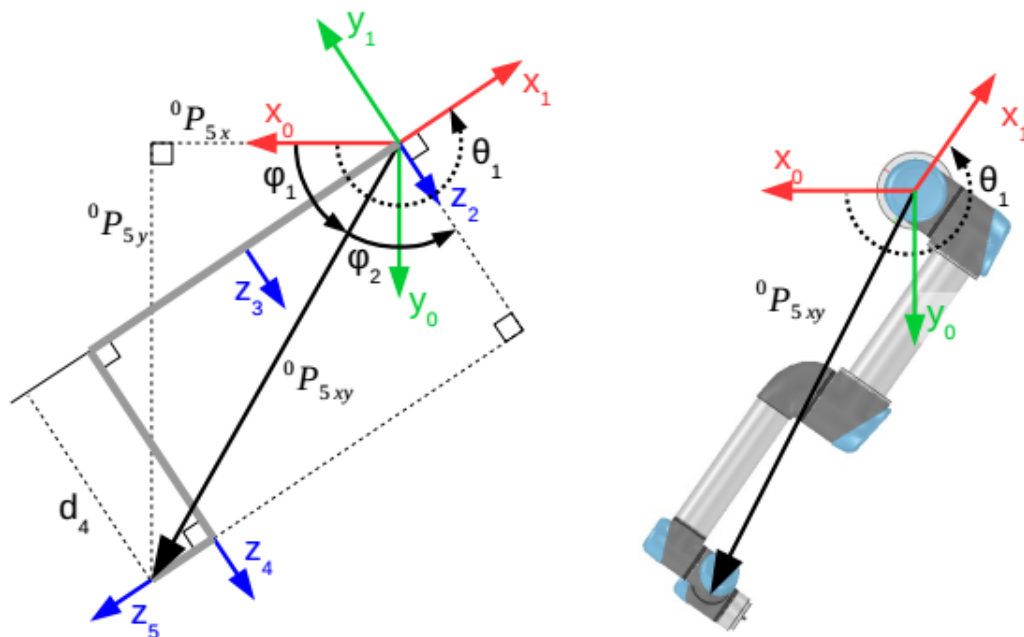
Portanto, existem duas soluções possíveis que correspondem ao ombro estar posicionado para a esquerda ou para a direita.

Para encontrar θ_5 , nota-se que ${}^1P_{6y}$ depende apenas de θ_5 . Como ilustrado na Figura 4.7, ${}^1P_{6y}$ é dado por:

$$-{}^1P_{6y} = d_4 + d_6 \cos \theta_5 \quad (4.29)$$

Outra forma de expressar ${}^1P_{6y}$ é olhando ${}^1\mathbf{P}_6$ como rotação de ${}^0\mathbf{P}_6$ em torno de z_1 , assim

Figura 4.6: Ilustração da vista superior do manipulador para encontrar θ_1 por meio de sua geometria.



(a) Geometria do UR5 para encontrar θ_1

(b) Vista superior do UR5

Fonte: Andersen (2018, p.5).

têm-se:

$$\begin{aligned}
 {}^0P_6 &= {}^0R \cdot {}^1P_6 \Leftrightarrow \\
 {}^1P_6 &= {}^0R^T \cdot {}^0P_6 \\
 \begin{bmatrix} {}^1P_{6x} \\ {}^1P_{6y} \\ {}^1P_{6z} \end{bmatrix} &= \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \cdot \begin{bmatrix} {}^0P_{6x} \\ {}^0P_{6y} \\ {}^0P_{6z} \end{bmatrix} \\
 \begin{bmatrix} {}^1P_{6x} \\ {}^1P_{6y} \\ {}^1P_{6z} \end{bmatrix} &= \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^0P_{6x} \\ {}^0P_{6y} \\ {}^0P_{6z} \end{bmatrix}
 \end{aligned}$$

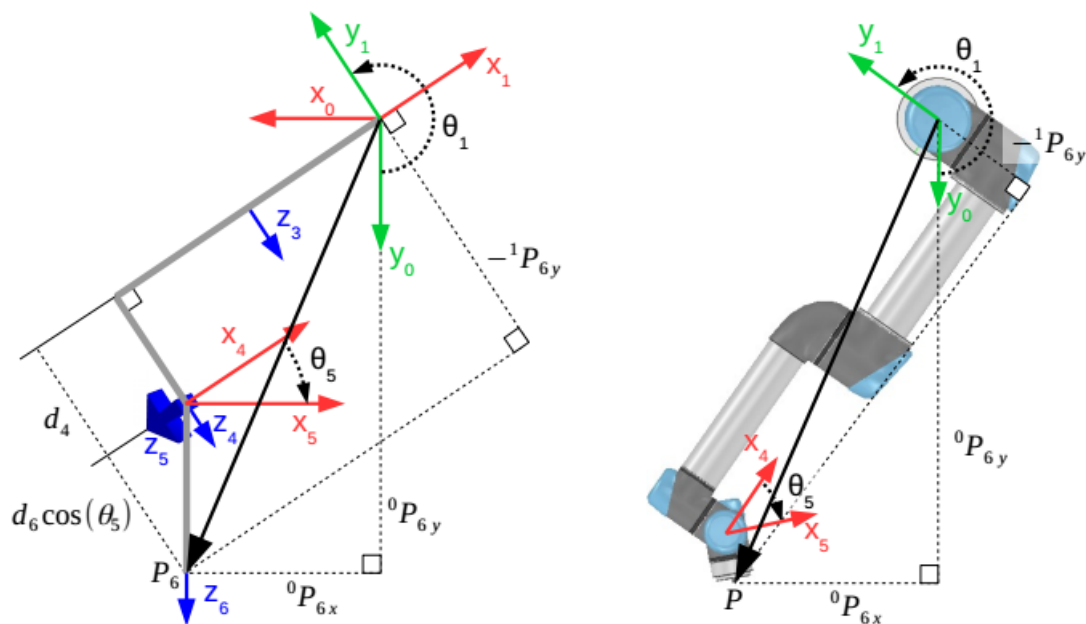
$${}^1P_{6y} = {}^0P_{6x} \cdot (-\sin \theta_1) + {}^0P_{6y} \cdot \cos \theta_1. \tag{4.30}$$

Combinando as Equações 4.29 e 4.30, pode-se eliminar o termo ${}^1P_{6y}$ e expressar θ_5 usando termos conhecidos:

$$\theta_5 = \pm \arccos \left(\frac{{}^0P_{6x} \sin \theta_1 - {}^0P_{6y} \cdot \cos \theta_1 - d_4}{d_6} \right) \tag{4.31}$$

Novamente existem duas soluções que correspondem ao pulso estando para cima ou para baixo. A soma das juntas ($\theta_2 + \theta_3 + \theta_4$) pode fazer com que o efetor-final esteja localizado na mesma posição, mas com o pulso invertido. A orientação é então corrigida por θ_6 .

Figura 4.7: Ilustração da vista superior do manipulador para encontrar θ_5 por meio de sua geometria.


 (a) Geometria do UR5 para encontrar θ_5

(b) Vista superior do UR5

Fonte: Andersen (2018, p.6).

Para determinar θ_6 , visualizamos y_1 da coordenada da sexta junta: ${}^6\hat{Y}_1$. Como ilustrado na Figura 4.8, o termo ${}^6\hat{Y}_1$ pode ser descrito em coordenadas esféricas:

$${}^6\hat{Y}_1 = \begin{bmatrix} -\sin \theta_5 \cos \theta_6 \\ \sin \theta_5 \cos \theta_6 \\ -\cos \theta_5 \end{bmatrix}. \quad (4.32)$$

Pode-se também expressar ${}^6\hat{Y}_1$ como uma rotação de θ_1 no plano $x - y$ da coordenada de referência:

$${}^6\hat{Y}_1 = \begin{bmatrix} -{}^6X_{0x} \sin \theta_1 + {}^6Y_{0x} \cos \theta_1 \\ -{}^6X_{0y} \sin \theta_1 + {}^6Y_{0y} \cos \theta_1 \\ -{}^6X_{0z} \sin \theta_1 + {}^6Y_{0z} \cos \theta_1 \end{bmatrix}. \quad (4.33)$$

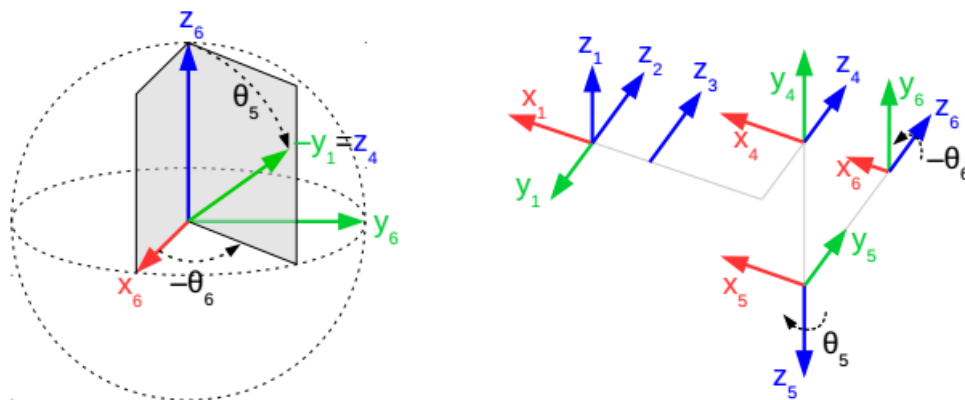
As coordenadas x e y dos vetores das Equações 4.32 e 4.33 formam um sistema que pode ser resolvido como:

$$\theta_6 = \arctan \left(\frac{-{}^6X_{0y} \sin \theta_1 + {}^6Y_{0y} \cos \theta_1}{\sin \theta_5}, \frac{{}^6X_{0x} \sin \theta_1 - {}^6Y_{0x} \cos \theta_1}{\sin \theta_5} \right) \quad (4.34)$$

Se $\sin \theta_5 = 0$ então a solução é indeterminada. Isso significa que as juntas 2, 3, 4 e 6 estão alinhadas. As juntas 2, 3 e 4 podem girar o efeitor-final em torno da sexta junta sem move-la, tornando-a redundante. Dessa forma, θ_6 pode ser definido por um valor arbitrário.

Se os dois numeradores na Equação 4.34 forem iguais a zero, então a solução também é indeterminada, mas neste caso $\sin \theta_5$ também é igual a zero e a situação anterior se repete.

Figura 4.8: Ilustração de ${}^6\hat{Y}_1$ expresso em coordenadas polares para encontrar θ_6 .



(a) Coordenadas polares expressando ${}^6\hat{Y}_1$.

(b) Vista das coordenadas relevantes para solucionar θ_6 .

Fonte: Andersen (2018, p.8).

Na Figura 4.9 ilustra-se as três juntas restantes, que configuram um manipulador 3R (três juntas de revolução). Essas juntas são todas paralelas. A transformação ${}^1_4\mathbf{T}$ é ilustrada na Figura 4.9(a) no plano $x-z$ da coordenada da primeira junta. A partir disso, o ângulo ϕ_3 pode ser encontrado por meio da lei dos cossenos:

$$\cos \phi_3 = \frac{(-a_2)^2 + (-a_3)^2 - |{}^1P_{4xz}|^2}{2(-a_2)(-a_3)} = \frac{a_2^2 + a_3^2 - |{}^1P_{4xz}|^2}{2a_2a_3}. \quad (4.35)$$

A relação entre $\cos \phi_3$ e $\cos \theta_3$ é:

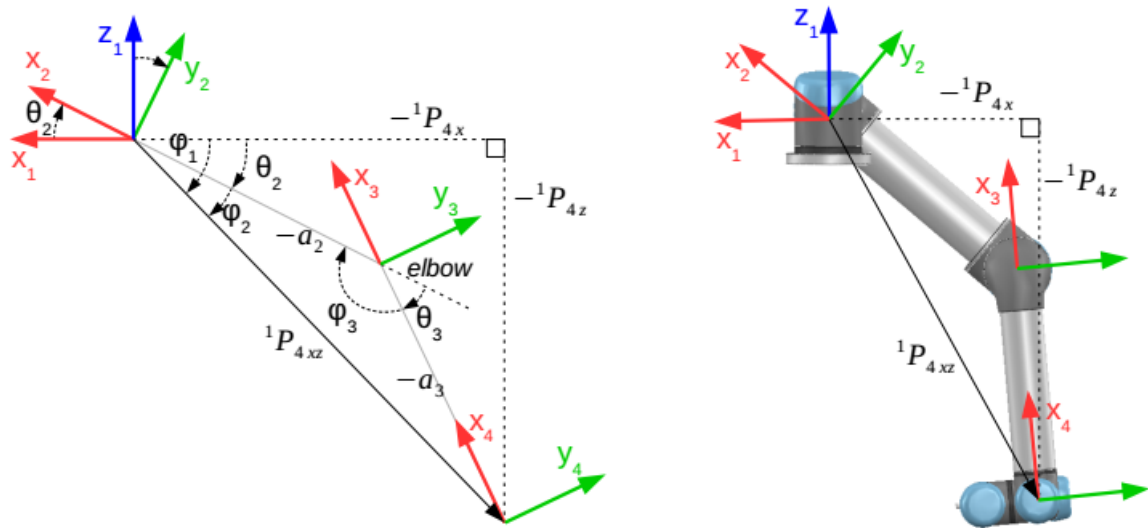
$$\cos \theta_3 = \cos (\pi - \phi_3) = -\cos \phi_3 \quad (4.36)$$

Combinando as Equações 4.35 e 4.36 têm-se:

$$\theta_3 = \pm \arccos \left(\frac{|{}^1P_{4xz}|^2 - a_2^2 - a_3^2}{2a_2a_3} \right). \quad (4.37)$$

Existem duas soluções possíveis, correspondendo ao cotovelo estando para cima ou para baixo.

Figura 4.9: Ilustração das juntas 2, 3 e 5 constituindo um manipulador 3R.



(a) Configuração das juntas 2, 3 e 5.

(b) Vista das juntas 2, 3 e 5 do UR5 formando um manipulador 3R.

Fonte: Andersen (2018, p.6).

O ângulo θ_2 é dado pela diferença entre ϕ_1 e ϕ_2 que são dados por:

$$\phi_1 = \arctan(-{}^1P_{4z}, -{}^1P_{4x}), \quad (4.38)$$

$$\phi_2 = \arcsin \left(\frac{-a_3 \sin \phi_3}{|{}^1P_{4xz}|} \right). \quad (4.39)$$

O termo $\sin \phi_3$ pode ser substituído por $\sin \theta_3$ pois $\sin \phi_3 = \sin(180^\circ - \theta_3) = \sin \theta_3$. Portanto, têm-se:

$$\theta_2 = \phi_1 - \phi_2 = \arctan(-{}^1P_{4z}, -{}^1P_{4x}) - \arcsin \left(\frac{-a_3 \sin \phi_3}{|{}^1P_{4xz}|} \right). \quad (4.40)$$

Por fim, o ângulo θ_4 é definido como o ângulo de x_3 a x_4 medido em torno de z_4 . Assim, pode-se derivar θ_4 a partir do vetor ${}^3\hat{X}_4$ da matriz de transformação 3T_4 :

$$\theta_4 = \arctan({}^3X_{4y}, {}^3X_{4x}). \quad (4.41)$$

Com isso, tem-se ao todo oito soluções possíveis para o problema de cinemática inversa: $2_{\theta_1} \times 2_{\theta_5} \times 1_{\theta_6} \times 2_{\theta_3} \times 1_{\theta_2} \times 1_{\theta_4}$.

4.3 A Dinâmica do Manipulador

O problema da cinemática descreve o movimento do manipulador desconsiderando as forças que causam esse movimento. A dinâmica do manipulador, por outro lado, descreve a relação entre as forças que atuam no corpo e seu movimento. Em termos matemáticos, essa relação é dada por equações de movimento. O modelo dinâmico completo para o manipulador é dado por:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (4.42)$$

onde $\mathbf{q} = \mathbf{q}(t) \in \mathbb{R}^n$ são as variáveis das juntas, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ é a matriz de inércia, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ é a matriz de Coriolis e de aceleração centrípeta, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ é o vetor de gravidade, $\boldsymbol{\tau} \in \mathbb{R}^n$ é o vetor de torques de entrada e n é o número de graus de liberdade (RAGAZZON, 2012).

O modelo dinâmico pode ser aplicado de forma direta, onde dado os torques de entrada ($\boldsymbol{\tau}$) são obtidos os estados do robô manipulador ($\mathbf{q}, \dot{\mathbf{q}}$), ou pode ser utilizado o modelo dinâmico inverso para encontrar os torques de entrada para controlar o manipulador dado os estados ao longo do tempo. A dinâmica direta é utilizada para realizar a simulação do manipulador, enquanto a dinâmica inversa é normalmente utilizada para projetar controladores.

Existem dois métodos para desenvolver a dinâmica de um manipulador robótico: o método de Euler-Lagrange e o método de Newton-Euler. O método de Euler-Lagrange é derivado a partir da energia mecânica do manipulador, enquanto o método de Newton-Euler utiliza as forças e torques nas juntas para derivar a modelagem dinâmica. O método de Newton-Euler é usualmente preferida por se tratar de uma abordagem recursiva, que é uma abordagem que deriva o modelo com maior rapidez. Entretanto, já foi provado que o método de Euler-Lagrange pode ser computado recursivamente com rapidez equivalente ao método de Newton-Euler (HØIFØDT, 2011). Portanto, a abordagem à utilizar depende apenas das características de modelagem de cada projeto.

4.3.1 Método de Euler-Lagrange

A equação de Euler-Lagrange é descrita por:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = f, \quad (4.43)$$

onde L é a Lagrangiana do sistema, y é a coordenada do sistema e f é a força externa. A equação de Euler-Lagrange fornece uma formulação das equações de movimentos equivalentes às equações derivadas da Segunda Lei de Newton, porém a abordagem Lagrangiana fornece vantagens em sistemas mais complexos como ligações robóticas (SPONG et al., 2006). Em geral, a aplicação das equações de Euler-Lagrange em ligações robóticas resulta em um sistema de n equações diferenciais ordinárias de segunda ordem da forma

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \quad i = 1, \dots, n, \quad (4.44)$$

onde τ_i são as forças/torques na i -ésima junta. A ordem n do sistema é determinado pelo número de variáveis das juntas (q_i) que são necessárias para descrever a evolução do sistema. A Lagrangiana L é dada pela diferença entre a energia cinética K e a energia potencial P , ou seja, $L = K - P$. Substituindo na Equação 4.44, pode-se escrever a equação de Euler-Lagrange da seguinte forma:

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} = \tau_i \quad i = 1, \dots, n. \quad (4.45)$$

A energia cinética K têm duas componentes em torno das variáveis das juntas: a velocidade linear e a velocidade rotacional. Os termos das velocidades podem ser escritos em função da matriz Jacobiana da seguinte forma:

$$\mathbf{v}_i = \mathbf{J}_{vi}(\mathbf{q})\dot{\mathbf{q}}, \quad \boldsymbol{\omega}_i = \mathbf{J}_{\omega i}(\mathbf{q})\dot{\mathbf{q}}, \quad i = 1, \dots, n, \quad (4.46)$$

onde \mathbf{J}_{vi} são as partes lineares das Jacobianas geométricas e $\mathbf{J}_{\omega i}$ são as partes angulares das Jacobianas geométricas.

A energia cinética geral do manipulador é dada por:

$$K = \frac{1}{2} \dot{\mathbf{q}}^T \sum_{i=1}^n [m_i \mathbf{J}_{vi}(\mathbf{q})^T \mathbf{J}_{vi}(\mathbf{q}) + \mathbf{J}_{\omega i}(\mathbf{q})^T \mathbf{R}_i(\mathbf{q}) \mathbf{I}_i \mathbf{R}_i(\mathbf{q})^T \mathbf{J}_{\omega i}(\mathbf{q})], \quad (4.47)$$

onde m_i é a massa da articulação i , $\mathbf{R}_i(\mathbf{q})$ é a matriz de rotação entre a estrutura do corpo e o referencial inercial e \mathbf{I}_i é o tensor de inércia da coordenada i . A energia cinética pode, também, ser escrita da forma $K = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$. Portanto, a matriz de inércia $\mathbf{M}(\mathbf{q})$ é dada por:

$$\mathbf{M}(\mathbf{q}) = \sum_{i=1}^n [m_i \mathbf{J}_{vi}(\mathbf{q})^T \mathbf{J}_{vi}(\mathbf{q}) + \mathbf{J}_{\omega i}(\mathbf{q})^T \mathbf{R}_i(\mathbf{q}) \mathbf{I}_i \mathbf{R}_i(\mathbf{q})^T \mathbf{J}_{\omega i}(\mathbf{q})], \quad (4.48)$$

e a equação de Euler-Lagrange pode agora ser escrita da seguinte forma:

$$\sum_j M_{i,j} \ddot{q}_j + \sum_{i,j} \left(\frac{\partial M_{k,j}}{\partial q_i} - \frac{1}{2} \frac{\partial M_{i,j}}{\partial q_k} \right) \dot{q}_i \dot{q}_j + \frac{\partial P}{\partial q_i} = \tau_i, \quad (4.49)$$

onde $M_{i,j}$ é o elemento (i,j) de $\mathbf{M}(\mathbf{q})$. Os coeficientes dos termos $\dot{q}_i \dot{q}_j$ são conhecidos por coeficientes de Christoffel e podem ser rescritos como:

$$c_{ijk} = \left(\frac{\partial M_{k,j}}{\partial q_i} - \frac{1}{2} \frac{\partial M_{i,j}}{\partial q_k} \right) = \frac{1}{2} \left[\frac{\partial M_{k,j}}{\partial q_i} + \frac{\partial M_{k,i}}{\partial q_j} - \frac{\partial M_{i,j}}{\partial q_k} \right]. \quad (4.50)$$

O elemento (k,j) da matriz $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ é definido como

$$c_{kj} = \sum_{i=1}^n c_{ijk}(\mathbf{q}) \dot{q}_i. \quad (4.51)$$

A energia potencial pode ser encontrada por

$$P = \sum_{i=1}^n m_i \mathbf{g}_o^T \mathbf{r}_{ci}, \quad (4.52)$$

onde \mathbf{g}_o é o vetor de gravidade da coordenada inercial e o vetor \mathbf{r}_{ci} é o vetor do centro de massa da articulação i . O vetor de gravidade $\mathbf{g}(\mathbf{q})$ é definido como

$$\mathbf{g}(\mathbf{q}) = \frac{\partial P}{\partial \mathbf{q}_i}. \quad (4.53)$$

As matrizes $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ e o vetor $\mathbf{g}(\mathbf{q})$ formam a dinâmica do manipulador robótico dada pela Equação 4.42.

4.4 Considerações Finais

Neste Capítulo foi apresentado o manipulador UR5 e a sua modelagem cinemática e dinâmica. A priori conceitos necessários para a modelagem de manipuladores série são descritos. A modelagem cinemática, que trata do movimento de um corpo sem levar em consideração as forças que atuam sobre ele, é abordada para o manipulador UR5. O mapeamento entre o espaço das juntas e o espaço cartesiano é definido utilizando a representação de Denavit-Hartenberg. Por outro lado, a cinemática inversa, que trata do mapeamento entre o espaço cartesiano e o espaço das juntas, é definida utilizando uma abordagem analítica baseada na geometria do manipulador. Por fim, a dinâmica do manipulador é descrita em função do método de Euler-Lagrange.

No Capítulo 5 é apresentado o sistema desenvolvido para a investigação das abordagens

de controle e predição no contexto de Internet Tátil. O manipulador UR5 descrito neste capítulo é utilizado pelo sistema de maneira virtual como robô remotamente controlado.

Capítulo 5

Sistema para Investigação de Abordagens de Inteligência Artificial Preditiva

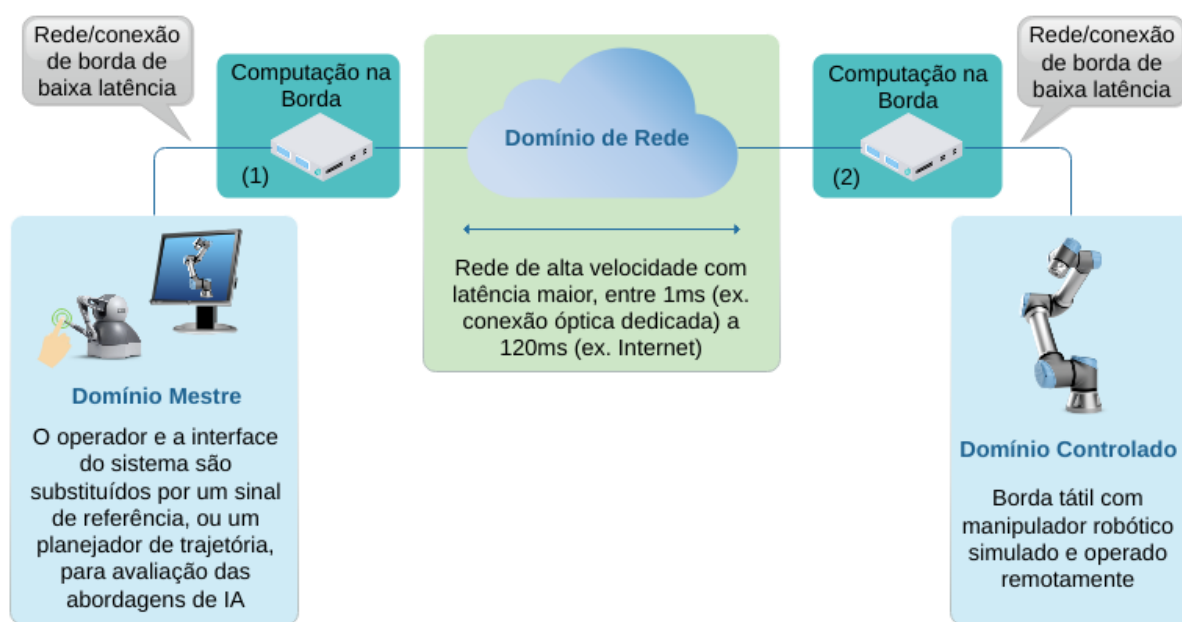
Esse trabalho é motivado principalmente pelo avanço da Internet Tátil e os desafios que ainda precisam ser solucionados para a sua consolidação. Grandes esforços estão sendo feitos para que a comunicação móvel 5G possua requisitos de conectividade ultra-responsiva e ultra-confiável, *Ultra-Reliable and Low-Latency Communication* (URLLC), compatíveis com a Internet Tátil. Com isso, o principal obstáculo para o estabelecimento da Internet Tátil, o desafio de 1 ms de latência, pode ser resolvido do alcance desse modo de comunicação. Visando esse cenário, o desafio passa a ser superar a latência inserida, por exemplo, por outras tecnologias de rede que façam parte do caminho de comunicação entre o domínio mestre e controlado. Para ultrapassá-lo, abordagens em torno de Inteligência Artificial Preditiva aliada com Computação de Borda podem ser usados para diminuir a percepção de atraso do sistema.

Neste Capítulo é apresentado o sistema desenvolvido para a investigação de abordagens de controle e predição no âmbito da Internet Tátil. Na Seção 5.1 a arquitetura base do sistema é descrita e os componentes do sistema apresentados. A simulação do manipulador robótico é abordada na Seção 5.2, onde é apresentado o ambiente de simulação de robôs utilizado no sistema. Em seguida, na Seção 5.3 é apresentada a arquitetura de comunicação do sistema e as aplicações desenvolvidas para o lado mestre e lado controlado. Por fim, as considerações finais a respeito do capítulo são relatadas na Seção 5.4.

5.1 Arquitetura Base

Tendo em vista as considerações previamente, é proposta uma arquitetura básica, apresentada no Diagrama da Figura 5.1, para investigar a diminuição da percepção do atraso em sistemas para a Internet Tátil por meio de tecnologias de IA preditiva na borda. Além dos três domínios característicos de qualquer sistema de Internet Tátil (domínio mestre, domínio de rede e domínio controlado), dispositivos de computação na borda, situados nas extremidades do sistema, são adicionados para dar suporte as abordagens de Inteligencia Artificial e controle, realizando a troca de informações entre os domínios do sistema. As estratégias de controle e Inteligencia Artificial baseadas em predição podem estar presentes em um dos dispositivos ou em ambos, dependendo da abordagem aplicada.

Figura 5.1: Diagrama da estrutura geral do sistema de Internet Tátil com computação na borda.



Fonte: Elaborada pelo autor.

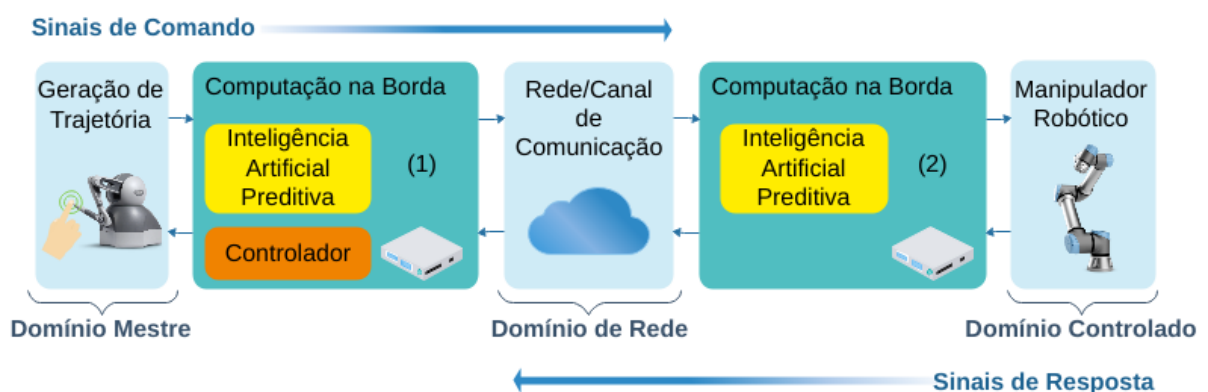
O padrão 5G está sendo moldado para suportar nativamente o URLLC. O 3GPP¹ (*3rd Generation Partnership Project*) determinou os cenários e requisitos, incluindo o levantamento de requisitos para o URLLC que é essencial para a Internet Tátil. Latência de 1 ms na interface aérea e um sistema 99,99 % confiável são requisitos atuais do 3GPP para o URLLC (AIJAZ; SOORIYABANDARA, 2018). Além disso, o 3GPP estima uma latência de 0,5 ms entre a conexão de borda e o usuário. No âmbito deste trabalho de avaliação, o impacto da conexão de borda de baixa latência será desconsiderado, e será

¹<https://www.3gpp.org/>

avaliado o impacto da latência de comunicação do Domínio de Rede (uma rede de alta velocidade com latência entre 1ms e 120ms).

Em uma aplicação de Internet Tátil a imersão do usuário no ambiente remoto é primordial para garantir a qualidade da experiência e evitar *cybersickness*. Dessa forma, o sistema de Internet Tátil deve prover resposta visual, auditiva e háptica fazendo com que o sistema possua vários laços de controle e, conseqüentemente, vários fluxos de informação na rede. A fim de investigar as abordagens para compensação do atraso da rede, o sistema desenvolvido possui apenas o laço de controle cinemático/háptico, já que o foco está na predição da resposta háptica/cinemática. Além do mais, para validação das abordagens estudadas o operador foi substituído por um gerador de trajetória sem a necessidade de uma interface homem-máquina, já que não têm-se resposta visual do lado controlado. Na Figura 5.2 ilustra-se o diagrama contendo os componentes do sistema e os fluxos dos sinais de comando e dos sinais de resposta que formam o laço cinemático do sistema. Varrendo o diagrama da direita para a esquerda têm-se: no domínio mestre a geração da trajetória ou posição de referência; o dispositivo de computação na borda (1), que possui algoritmo de controle cinemático e/ou algoritmo de inteligência artificial preditiva; no domínio de rede têm-se a rede ou canal de comunicação; o dispositivo de computação na borda (2) que pode, ou não, conter algoritmo de inteligência artificial preditiva; no domínio controlado o manipulador robótico com o controlador interno. Além disso, os dispositivos de computação na borda devem realizar toda a computação necessária para adquirir e tratar os dados provindos dos domínios do sistema.

Figura 5.2: Diagrama do sistema para investigação de técnicas de inteligência artificial preditiva.



Fonte: Elaborada pelo autor.

5.2 Simulação do Manipulador Robótico

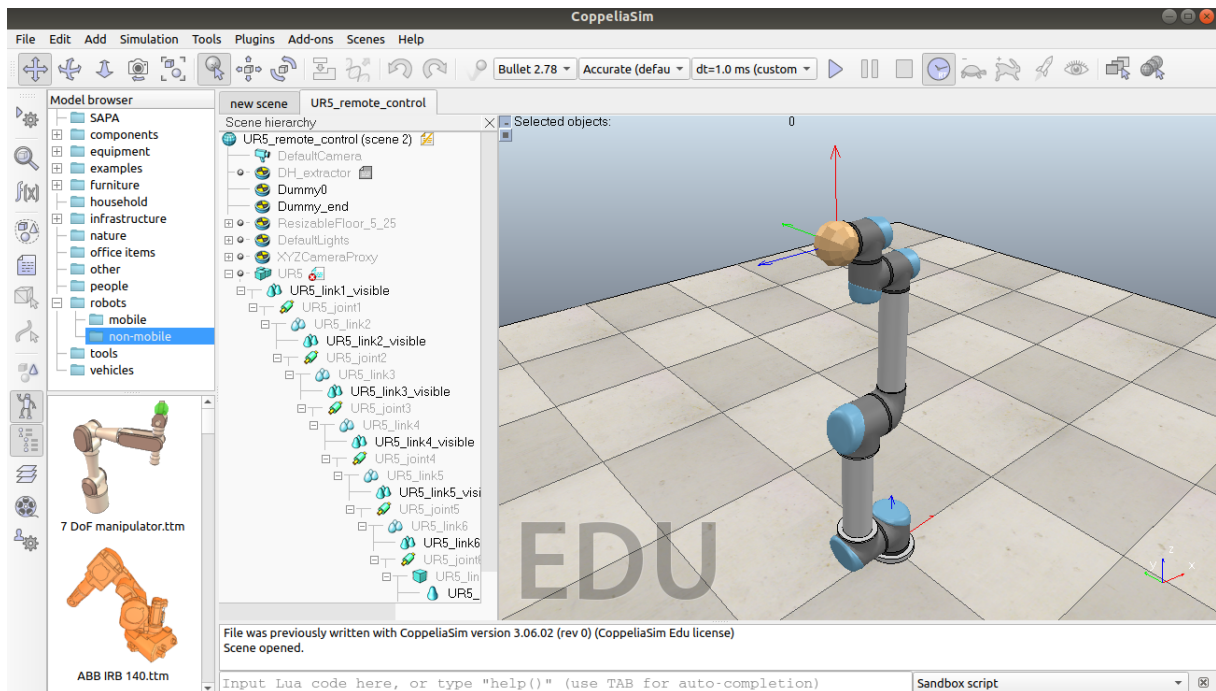
O robô presente no domínio controlado consiste de um braço robótico virtual (especificamente o manipulador UR5 desenvolvido pela *Universal Robots*) simulado no software Coppeliasim² (a interface do software Coppeliasim é apresentada na Figura 5.3). O software de simulação de robôs Coppeliasim pode ser utilizado para simulações de automação de fábrica, prototipagem e verificação rápidas, monitoramento remoto, verificação dupla de segurança como gêmeo digital, entre outros. Baseada em uma arquitetura de controle distribuído, no Coppeliasim cada objeto/modelo pode ser controlado individualmente por meio de um *script* embarcado, um *plugin*, um nó do *Robot Operating System* (ROS) ou BlueZero, uma solução customizada, ou um cliente remoto da API (*Remote API*). Composto por um cliente e por um servidor, a interface *Remote API* permite a iteração com a simulação do Coppeliasim por meio de uma aplicação externa via comunicação por *socket*. O lado servidor consiste da cena de simulação no Coppeliasim e o lado cliente é a aplicação externa desenvolvida. As seguintes linguagens podem ser utilizadas pelo cliente da *Remote API*: C/C++, Python, Java, Lua, Matlab e Octave. O cliente *Remote API* pode ser incorporado em qualquer hardware e robô real e permite as chamadas de funções remotas (ROHMER; SINGH; FREESE, 2013). A comunicação entre o cliente e o servidor é por padrão assíncrona, mas por meio do modo de operação síncrono, a cena no Coppeliasim pode ser simulada sincronicamente com a aplicação. Além disso, as chamadas de funções podem ser bloqueantes ou não-bloqueantes. Em uma chamada de função bloqueante, é necessário obter uma resposta do servidor para que a próxima função seja executada. Já em uma chamada de função não-bloqueante nenhuma resposta é esperada e a próxima função pode ser executada em seguida.

5.3 Processos e Interfaces de Comunicação

As APIs do lado mestre e do lado controlado são executadas nos dispositivos de computação na borda por meio de scripts Python. A interface de comunicação entre as APIs é baseada em *User Datagram Protocol* (UDP), já a comunicação entre o cliente e o servidor do *Remote API* é baseada em *Transmission Control Protocol* (TCP). Embora haja a possibilidade de perdas de pacotes com o protocolo UDP, a alta velocidade de comunicação é priorizada em detrimento da confiabilidade. O cliente *Remote API* é incorporado na API do domínio controlado para realizar a interface com o robô virtual no Coppeliasim. O cliente *Remote API* comunica-se com a cena no Coppeliasim de forma assíncrona com modos de operação bloqueantes e não-bloqueantes. A taxa de amostragem utilizada para

²<https://www.coppeliarobotics.com/>

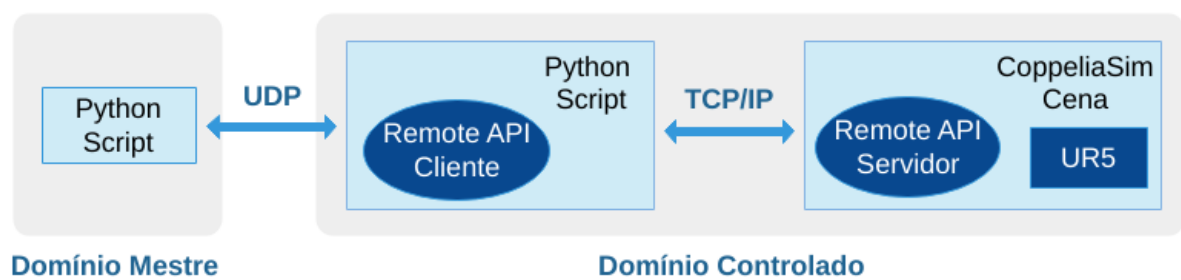
Figura 5.3: Interface do software de simulação de robôs CoppeliaSim.



Fonte: Elaborada pelo autor.

a cena no CoppeliaSim foi de 2 ms. A troca constante de informações entre os domínios é realizada por meio de funções de laços nas APIs, mediante os *sockets* de comunicação (dois *sockets* são abertos para comunicação entre as aplicações mestre e escravo). A arquitetura de comunicação do sistema é ilustrado na Figura 5.4.

Figura 5.4: Arquitetura de comunicação do sistema.

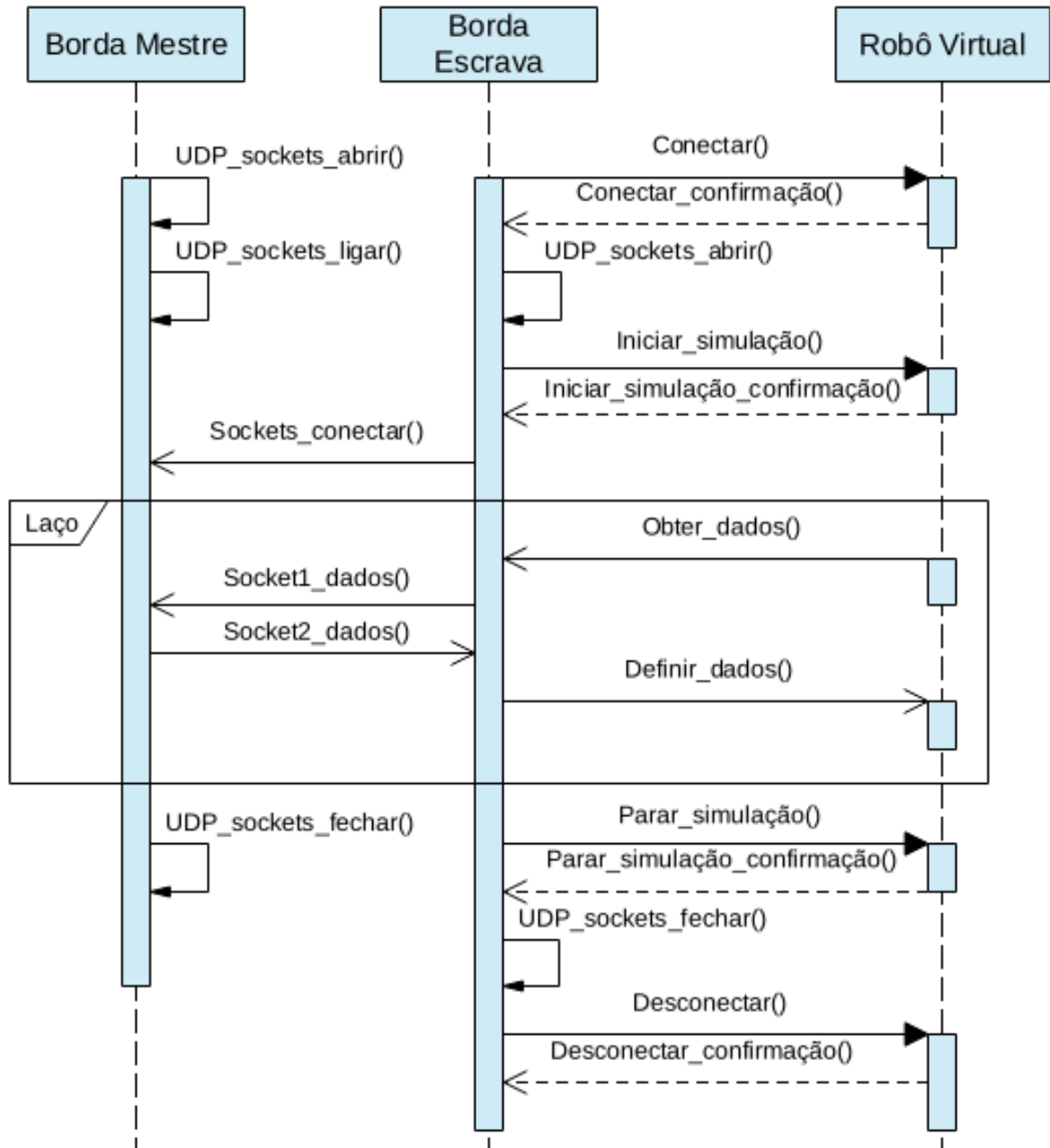


Fonte: Elaborada pelo autor.

O diagrama de fluxo de dados entre a borda mestre, a borda escrava e o manipulador UR5 simulado no CoppeliaSim é ilustrado na Figura 5.5. A priori, a API da borda mestre é executada, os *sockets* são abertos e ligados, e esperam a comunicação do cliente. Este, por sua vez, é a API da borda escrava, que é executada logo em seguida. A API da borda escrava solicita conexão com o robô virtual, se ela for aceita então a API abre os *sockets* para comunicação com a borda mestre e, em seguida, inicia a simulação do robô.

Por meio dos laços, as informações cinemáticas/hápticas são trocadas entre os domínios, para controle remoto do robô. Uma vez que a aplicação é encerrada, a API da borda mestre envia um aviso para a API da borda escrava e fecha os *sockets*. A API da borda escrava para a simulação do robô virtual, fecha os *sockets* e encerra a comunicação com a ferramenta de simulação robótica.

Figura 5.5: Diagrama do fluxo de dados entre os componentes do sistema.



Fonte: Elaborada pelo autor.

O fluxo de controle de controle da aplicação no lado mestre é ilustrado no fluxograma na Figura 5.6. Inicialmente, a aplicação inicializa as variáveis e especifica os ganhos do

controlador, em seguida os *sockets* UDP são abertos e ligados. Dentro do laço de repetição, a pose do efetor-final do manipulador é especificada no espaço da tarefa. A cinemática inversa calcula os valores das variáveis das juntas dado a pose no espaço da tarefa. Com a cinemática inversa calculada, o *sockets* espera os dados enviados pelo lado escravo. Esses dados são codificados em formato *JavaScript Object Notation* (JSON) da seguinte forma:

```

1 {
2   "msg_escravo": {
3     "q": [[q[0, 0]], [q[1, 0]], [q[2, 0]],
4           [q[3, 0]], [q[4, 0]], [q[5, 0]]]
5   }
6 }
```

O identificador “msg_escravo” informa que é uma mensagem enviada pela aplicação escrava. A mensagem contém o vetor de posição das juntas do manipulador. Uma vez que a mensagem é recebida, ela é decodificada para converter o dado para o formato de estrutura de Python. Com o vetor de posição das juntas desejado e atual, pode-se executar a estratégia de controle para obter a velocidade das juntas do manipulador. O vetor de velocidades das juntas e a mensagem para encerrar ou não a aplicação escrava são codificadas em formato JSON da seguinte forma:

```

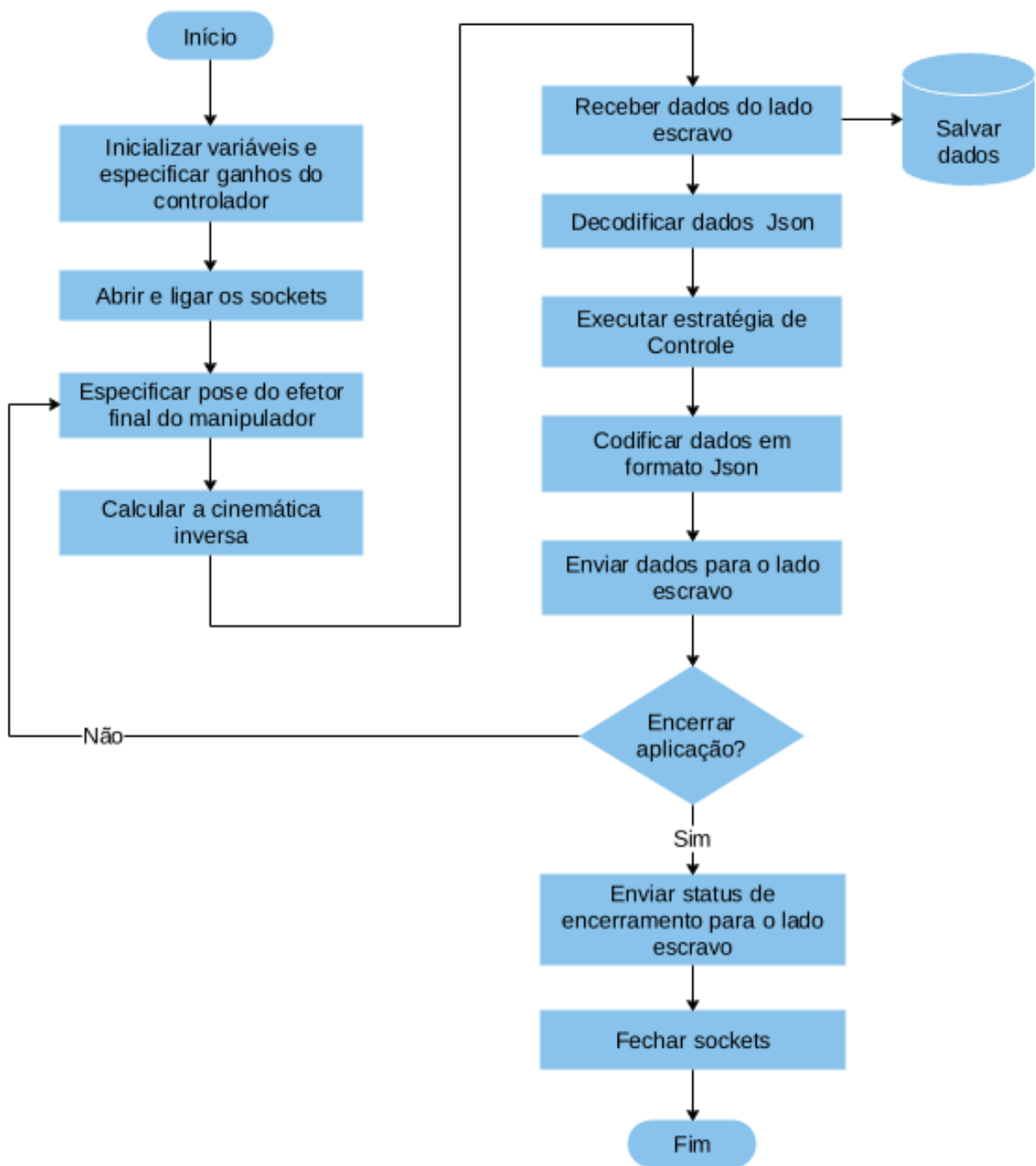
1 {
2   "msg_mestre": {
3     "dq": [[dq[0, 0]], [dq[1, 0]], [dq[2, 0]],
4           [dq[3, 0]], [dq[4, 0]], [dq[5, 0]]],
5     "Encerrar": False
6   }
7 }
```

O identificador “msg_mestre” informa que é uma mensagem enviada pela aplicação mestre. Em seguida, a mensagem é enviada via *socket* para a aplicação escrava. O laço continua até o critério de parada (tempo máximo de execução da aplicação) ter sido atingido. Se for solicitado o encerramento da aplicação, então é enviada uma mensagem para encerrar a aplicação escrava e, logo após, os *sockets* são fechados.

O fluxo de controle da aplicação do lado escravo, por sua vez, é ilustrada na Figura 5.7. A aplicação escrava contém o cliente Remote API para comunicação com o ambiente virtual criado no CoppeliaSim. A priori é necessário estabelecer a comunicação com o Servidor no CoppeliaSim. Realizada a conexão com o servidor Remote API, os *sockets* são abertos e são obtidos os identificadores das juntas do manipulador virtual. É necessário obter os

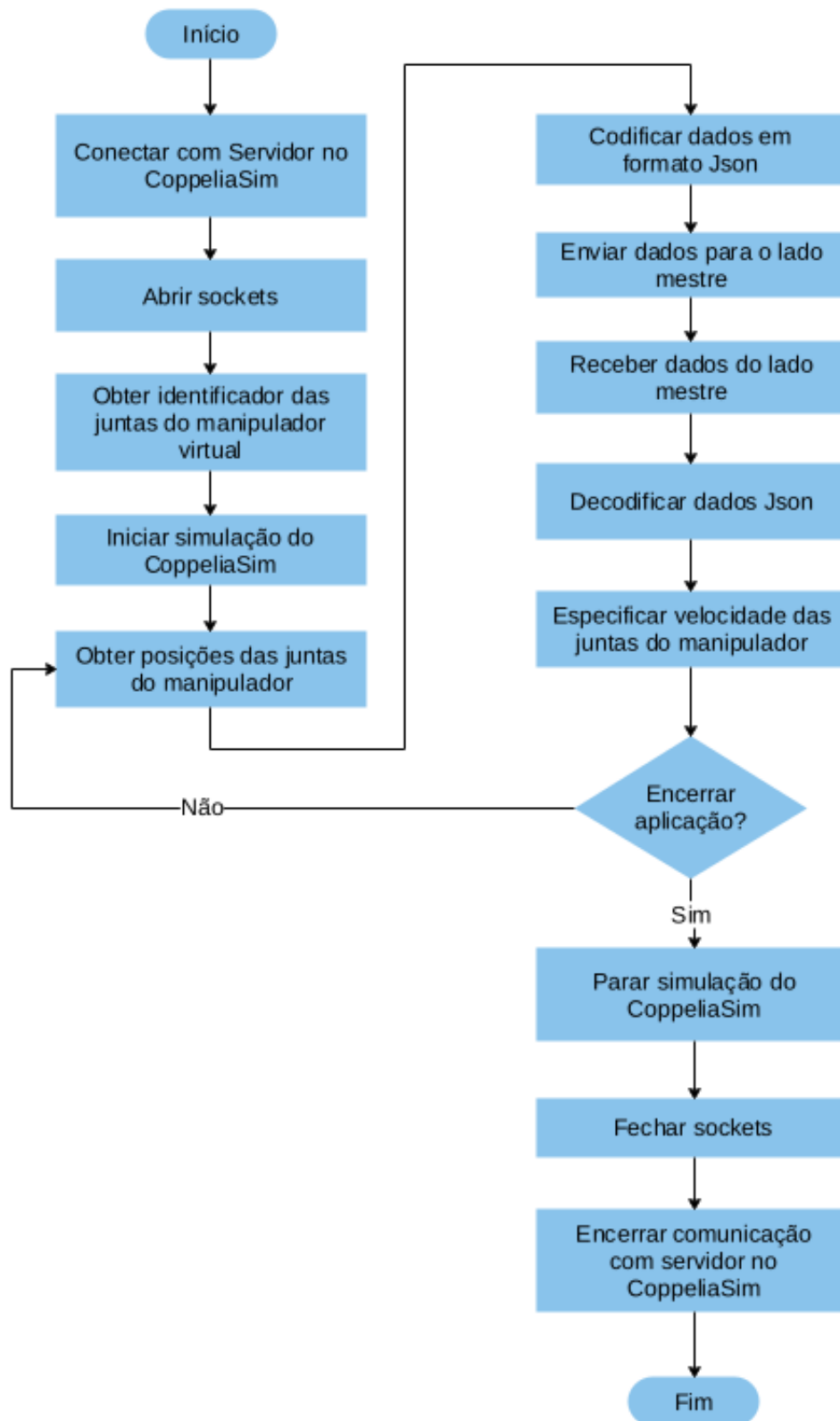
identificadores para controlar remotamente, por meio da API, cada junta do manipulador (ou objetos na cena). A simulação no CoppeliaSim é iniciada e o laço de repetição pode ser iniciado. Dentro do laço, as posições das juntas do manipulador virtual são obtidas e codificadas em formato JSON (discutida no parágrafo supra). Os dados são enviados para o lado mestre e, em seguida, é esperado os dados em formato JSON enviados pelo lado mestre. Uma vez recebidos os dados, eles são decodificados para obtenção das velocidades das juntas do manipulador. Logo após, as velocidades das juntas do manipulador virtual são estabelecidas. Se o critério de parada (mensagem enviada pelo cliente para encerrar a aplicação) for atingido, então o laço é encerrado, a simulação no CoppeliaSim é parada, os *sockets* são fechados e a comunicação com o servidor no CoppeliaSim é então finalizada.

Figura 5.6: Fluxo de controle da aplicação na borda mestre.



Fonte: Elaborada pelo autor.

Figura 5.7: Fluxo de controle da aplicação na borda escrava.



Fonte: Elaborada pelo autor.

5.4 Considerações Finais

Neste Capítulo foi apresentado o sistema desenvolvido para investigação das abordagens de controle e inteligência artificial preditiva. O sistema consiste dos domínios presentes na arquitetura da Internet Tátil com dispositivos de computação na borda que dão suporte as abordagens de controle e predição. O ambiente de simulação robótico CoppeliaSim foi utilizado para simulação do manipulador UR5 e um cliente *Remote API* é utilizado para realizar o controle remoto por meio de aplicação externa. A interface de comunicação entre as aplicações mestre e escrava é descrita, bem como os fluxos de controles de ambas aplicações.

No Capítulo 6 são descritas as abordagens de controle e predição investigadas utilizando o sistema apresentado neste capítulo.

Capítulo 6

Abordagens de Controle e Inteligência Artificial Preditiva

Para investigar as soluções de inteligência artificial e validar o sistema proposto no capítulo anterior, é necessário realizar estudos sobre as estruturas de controle do sistema. Com isso, duas abordagens de controle cinemático são investigadas utilizando o sistema proposto: (i) uma abordagem com controle Proporcional-Integral-Derivativo (PID) e, (ii) uma abordagem com controle preditivo baseado em modelo (do inglês: *Model Predictive Control-MPC*). Para projetar os controladores é fundamental realizar a modelagem do manipulador, dado que podemos utilizar um modelo do manipulador para projetar os ganhos do controlador PID, e o controlador MPC necessita de um modelo do manipulador para execução da estratégia de controle. Além disso, um modelo do manipulador é necessário para o projeto de um observador, como o filtro de Kalman.

Neste Capítulo as abordagens de controle e inteligência artificial preditiva investigadas são descritas. Para isso, na Seção 6.1 a identificação do manipulador virtual UR5 é discutida. Na Seção 6.2 são apresentadas as abordagens de controle investigadas. As abordagens de predição são discutidas na Seção 6.3. Por fim, na Seção 6.4 são apresentadas as considerações finais acerca do capítulo.

6.1 Identificação do Sistema

Um modelo pode ser obtido por meio da análise do comportamento do sistema que é colocado em forma matemática ou a partir de dados observados do sistema (abordagem experimental). Um modelo matemático do manipulador pode ser obtido por meio do método de Euler-Lagrange, discutido no Capítulo 4. Entretanto, o manipulador UR5 possui um controlador interno que faz parte do sistema e é desconhecido para o usuário. Dessa forma, uma abordagem experimental é preferível.

Para a identificação do sistema, é necessário obter dados do sistema. Para isso uma entrada deve ser gerada, a saída deve ser medida, e com isso um modelo pode ser calculado. Existem vários métodos para a identificação de sistemas, como métodos baseados em parametrização de sistemas e regressão linear. Entretanto, esses métodos necessitam de um conhecimento prévio do sistema, como a ordem do modelo (GIER, 2015).

A identificação do modelo foi realizada utilizando um método baseado em identificação do subespaço, *Subspace Identification Methods* (SID), o método *Multivariable Output-Error State Space* (MOESP) (VERHAEGEN; DEWILDE, 1992), (VERHAEGEN, 1994). Esse método não necessita de nenhum conhecimento prévio do sistema e, além disso, fornece o modelo em espaço de estados que é favorável para métodos de filtragem, predição e estimação, e é a representação utilizada no método de controle MPC. Foi assumido, para a utilização do método, que o sistema é linear, embora o manipulador UR5 seja não-linear. Entretanto, o controlador interno compensa a ação da gravidade e atrito em todo o sistema, justificando a modelagem linear do sistema. Embora a inércia do manipulador UR5 tenha um comportamento não-linear, em baixas velocidades essa não-linearidade não é significativa (GIER, 2015).

O método de identificação MOESP é realizado em malha fechada, o que é conveniente já que o manipulador UR5 possui um controlador interno em malha fechada. Entretanto, os dados em malha fechada possuem correlação entre o ruído e entrada, fazendo com que as estimativas não sejam tão precisas como a identificação em malha aberta.

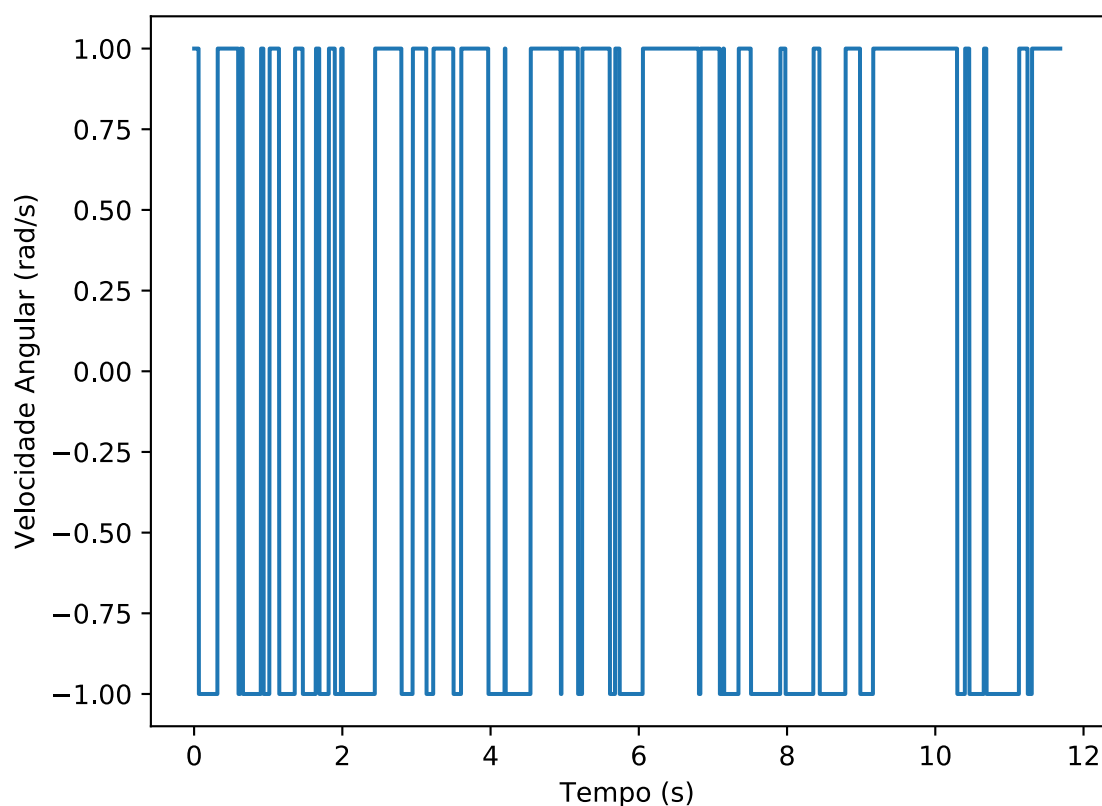
Assumindo que as juntas não interferem entre si, o manipulador UR5 pode ser modelado por seis sistemas *Single-Input Multiple-Output* (SIMO). A razão para a escolha de seis sistemas SIMO em detrimento de um sistema *Multiple-Input Multiple-Output* (MIMO), é o tempo de processamento computacional, já que para a implementação do controle MPC seis sistemas SIMO são mais rápidos que um sistema MIMO (GIER, 2015). Como o controle é realizado mediante a velocidade da junta, então a entrada do sistema é a velocidade da junta e as saídas são a posição e a velocidade da junta.

A identificação do manipulador consistiu dos seguintes passos: (i) gerar e salvar os dados de entrada e saída utilizando o cliente *Remote API* para controlar o manipulador virtual no CoppeliaSIM; (ii) com os dados obtidos, utilizar o método MOESP para obter a representação em espaço de estados das juntas do manipulador; (iii) avaliar as representações em espaço de estados obtidas.

Existem alguns requisitos que o sinal de entrada deve satisfazer para realizar a identificação do sistema. O sinal de entrada deve ter persistência de excitação para revelar as características do sistema. Uma maneira de gerar sinais de entrada para identificação de sistemas é utilizar *Pseudo-Random Binary Signals* (PRBS). Sinais PRBS são simples de implementar, periódicos, determinísticos e com propriedade de ruído branco (BAKKE,

2009). O sinal PRBS utilizado como entrada para identificação do sistema é ilustrado na Figura 6.1. A frequência de operação para obtenção dos sinais de saída do manipulador foi de 1 kHz (passo de discretização de 1 ms). Com os dados de entrada e saída do manipulador, a identificação do modelo pode ser feita. Para utilizar a abordagem MOESP é necessário especificar a ordem do sistema. Foi escolhido a utilização de 6 estados para representar o sistema, visto que um modelo maior exigiria mais esforço computacional e menos estados geram modelos menos precisos.

Figura 6.1: Representação do sinal PRBS utilizado como entrada para identificação do sistema.



Fonte: Elaborada pelo autor.

A dinâmica do sistema discreto no tempo, em espaço de estados, pode ser descrita por:

$$x(k+1) = Ax(k) + Bu(k) \quad (6.1)$$

$$y(k) = Cx(k) + Du(k) \quad (6.2)$$

onde $x(k)$ é o vetor de variáveis de estado do processo com dimensão $n = 6$, $u(k)$ é a entrada do processo, $y(k)$ é a saída do processo e A , B , C e D são as matrizes que caracterizam o modelo.

As matrizes A , B , C e D a seguir caracterizam o modelo da junta 1 (o esquemático das juntas e articulações do manipulador estão presentes no Capítulo 4), e foram identificadas pelo método de identificação de subespaço MOESP.

$$A = \begin{bmatrix} 9,99e-1 & -1,15e-4 & -2,60e-3 & 4,96e-3 & -9,60e-3 & -1,20e-2 \\ -3,44e-3 & 9,80e-1 & -4,30e-2 & 3,87e-2 & -3,71e-2 & -1,14e-1 \\ 1,24e-3 & 1,46e-2 & 9,56e-1 & -2,03e-1 & 8,63e-2 & 1,20e-1 \\ -1,64e-4 & -1,89e-3 & 3,09e-2 & 9,37e-1 & -2,77e-1 & 2,62e-2 \\ 5,43e-5 & -1,75e-4 & -4,56e-3 & 1,41e-1 & 9,09e-1 & 3,86e-1 \\ 1,34e-4 & 9,16e-4 & -7,31e-3 & 2,26e-2 & -1,32e-1 & 7,71e-1 \end{bmatrix}$$

$$B = \begin{bmatrix} -2,89e-5 \\ 2,87e-4 \\ 2,33e-3 \\ -3,25e-3 \\ -3,69e-3 \\ 4,83e-3 \end{bmatrix} \quad (6.3)$$

$$C = \begin{bmatrix} -3,94 & 2,87e-1 & -1,51e-2 & 2,00e-1 & -1,28e-1 & -2,14e-1 \\ -4,68e-1 & -4,27 & -2,34 & -1,23 & -7,41e-1 & 5,77e-1 \end{bmatrix} \quad (6.4)$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (6.5)$$

Uma vez que está sendo utilizado um manipulador UR5 virtual, todas as juntas possuem as mesmas características e, portanto, podemos tomar o modelo identificado para a junta 1 como o modelo das demais juntas. Para validar o desempenho deste modelo, a métrica utilizada foi a raiz quadrada do erro quadrático médio normalizado (do inglês: *Normalized Root Mean Square Error* - NRMSE), onde a raiz do erro quadrático médio (do inglês: *Root Mean Square Error* - RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (6.6)$$

é normalizada por

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \quad (6.7)$$

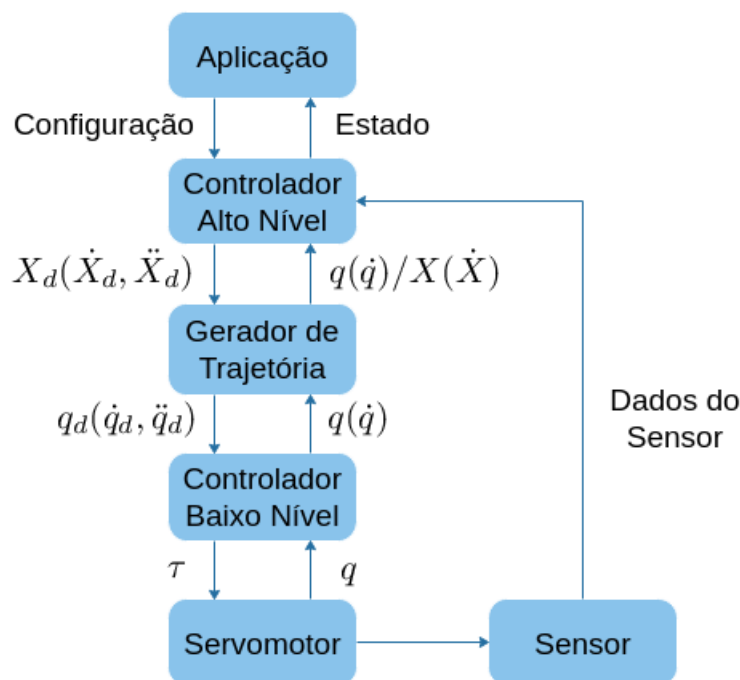
onde n é o número de dados, y_i é o i -ésimo valor medido, \hat{y}_i é o i -ésimo valor previsto e y_{max} e y_{min} é o maior e menor valor medido, respectivamente. O NRMSE do modelo identificado para a junta 1 é de 8,705 %.

Uma vez obtido os modelos das juntas do manipulador, na seção a seguir são tratadas as abordagens de controle para o sistema.

6.2 Abordagens de Controle

A arquitetura de controle de manipuladores robóticos geralmente segue a estrutura apresentada na Figura 6.2. A aplicação do sistema comunica-se com o controlador de alto nível, que recebe os dados do sensor e gera os dados para o gerador de trajetória. Este, por sua vez, é responsável por gerar a trajetória do manipulador no espaço das juntas a partir da trajetória no espaço da tarefa. O controlador de baixo nível gera os torques nas juntas (as entradas dos servomotores) dada a trajetória no domínio das juntas (geralmente é realizado o cálculo da dinâmica inversa do manipulador, para a realização desse controle de baixo nível).

Figura 6.2: Diagrama da estrutura geral de controle de manipuladores robóticos.



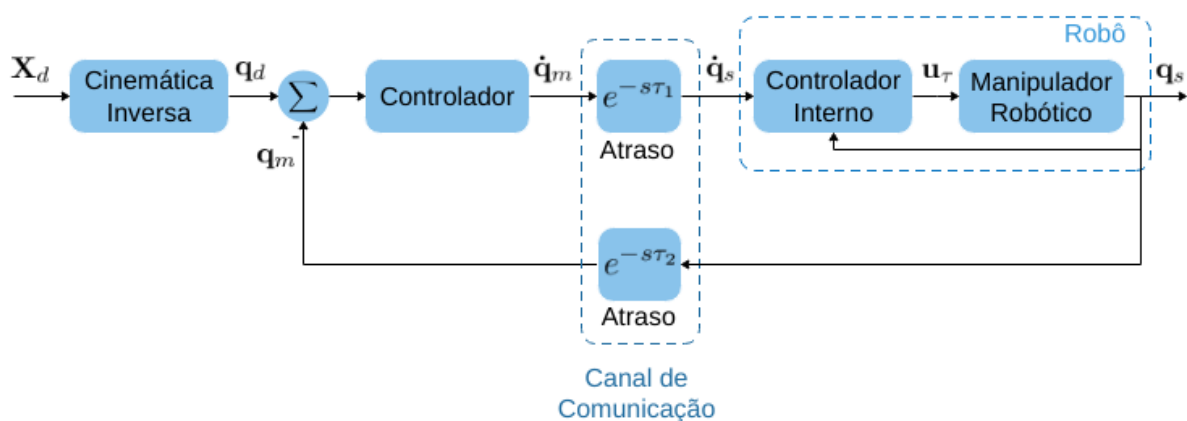
Fonte: Adaptado de Schrimpf, Lind e Mathisen (2013, p.3)

O controle de manipuladores pode ser projetado com base no espaço das juntas ou no espaço da tarefa. Os manipuladores realizam ações definidas no espaço da tarefa, assim no controle baseado no espaço das juntas primeiramente é realizado a cinemática inversa para que, em seguida, realize-se o controle, ou seja, o manipulador segue a trajetória no espaço das juntas. Por outro lado, no controle baseado no espaço das juntas, o controle é realizado em relação ao espaço da tarefa e, em seguida, é realizado o mapeamento no

espaço das juntas. Embora o controle no espaço da tarefa possua algumas vantagens (como a sensibilidade ao ambiente por exemplo), o controle no espaço das juntas foi selecionado devido a facilidade na sua implementação.

A estrutura de controle do sistema proposto é ilustrada na Figura 6.3, onde: \mathbf{X}_d é o vetor da pose desejada do efetor-final no espaço da tarefa; \mathbf{q}_d é o vetor de posição desejada das juntas; $\dot{\mathbf{q}}_m$ é o vetor de velocidade das juntas no lado mestre; \mathbf{q}_m é o vetor de posição das juntas no lado mestre; $\dot{\mathbf{q}}_s$ é o vetor de velocidade das juntas no lado escravo; \mathbf{u}_τ é o vetor dos torques nas juntas; \mathbf{q}_s é o vetor de posição das juntas no lado escravo; τ_1 é o atraso do sinal de comando; τ_2 o atraso no sinal de resposta no canal de comunicação (rede). O sinal de referência do controle é a posição desejada no espaço das juntas, o sinal de saída é a posição da junta e o sinal de controle é a velocidade da junta. No lado mestre têm-se primeiramente a posição no espaço da tarefa (a referência), que é mapeada no espaço das juntas por meio da cinemática inversa (discutida no Capítulo 4) e, em seguida, o controlador externo. No domínio de rede têm-se o canal de comunicação, com os respectivos atrasos dos sinais de comando e resposta, e no domínio controlado, o manipulador com seu controlador interno. É importante ressaltar que o manipulador têm seis graus de liberdade e, portanto, o sistema possui um total de seis malhas de controle. Entretanto, nos diagramas apresentados nesse trabalho, foram ilustrados blocos únicos com vetores de entrada e saída para representar os componentes das seis malhas do sistema.

Figura 6.3: Diagrama de controle do sistema.



Fonte: Elaborada pelo autor.

Sistemas de controle clássicos podem ser empregados para o rastreamento do sinal de referência: controlador Proporcional (P); Proporcional-Integral (PI); Proporcional-Derivativo (PD); Proporcional-Integral-Derivativo (PID). Controladores PID são bastante utilizados no setor industrial, pois possuem aplicabilidade em vários sistemas de con-

trole, são fáceis de implementar e de sintonizar. Portanto, um controle do tipo PID é proposto para o sistema, cuja lei de controle (no domínio do tempo) é dada por:

$$u(t) = K_P e(t) + \frac{K_P}{T_i} \int_0^t e(\tau) d\tau + K_P T_d \frac{de(t)}{dt} \quad (6.8)$$

onde $u(t)$ é o sinal de saída, $e(t)$ é o sinal erro, t é o tempo, K_P é o ganho proporcional, T_i é o tempo integrativo e T_d é tempo derivativo (KATHUSHIKO, 2011). O controlador PID também pode ser escrito em função do ganho integral

$$K_I = \frac{K_P}{T_i} \quad (6.9)$$

e do ganho derivativo

$$K_D = K_P T_d. \quad (6.10)$$

Com o objetivo de obter um conjunto ótimo de parâmetros do controlador PID, a sintonização do controlador PID é realizado utilizando Algoritmo Genético (GA), que é um tipo de algoritmo evolutivo. Algoritmos evolutivos são uma categoria de técnicas de aprendizado de máquina que baseiam-se na biologia evolutiva (seleção natural). Dada uma população de indivíduos, uma análise é realizada com base em uma função de custo de otimização para encontrar a solução ótima. Algoritmos genéticos são baseados na propagação de geração de indivíduos por seleção e podem ser utilizados em muitos problemas de otimização. Uma implementação de algoritmo genético começa com uma população aleatória de indivíduos que são avaliadas de acordo com o desempenho da métrica da função de custo. Os indivíduos melhores avaliados na geração têm maior probabilidade de passar para a próxima geração. Esses indivíduos são utilizados para gerar uma nova população baseado nas operações genéticas de elitismo, replicação, cruzamento e mutação. As gerações evoluem até que convirjam para um critério de parada (MIRANDA, 2007), (DURIEZ; BRUNTON; NOACK, 2017). Uma algoritmo genético é utilizado para encontrar os ganhos do controlador PID que minimizem a função de custo, descrita no domínio do tempo,

$$J(K_P, K_I, K_D) = (1 - e^{-\beta})(M_p + e_{ss}) + e^{-\beta}(t_s - t_r) \quad (6.11)$$

proposta em (GAING, 2004). M_p é o sobressinal, e_{ss} é o erro em estado estacionário, t_s é o tempo de acomodação, t_r o tempo de subida e β é o fator de ponderação. O fator de ponderação pode ser utilizado para satisfazer os requisitos desejados. Um fator de ponderação acima de 0,7 pode reduzir o sobressinal e o erro em estado estacionário, já

um fator menor que 0,7 pode reduzir o tempo de subida e o tempo de acomodação. O fator de ponderação foi definido em 1,5 pois uma resposta com redução do sobressinal e do erro em estado estacionário é desejada.

A sintonização do controlador PID foi realizada utilizando a função GA do Matlab de acordo com (BRUNTON; KUTZ, 2019). Primeiramente, uma função é criada para analisar o controlador PID com a função de custo, como apresentado no Algoritmo 1. As variáveis *parms* são os ganhos do controlador. A resposta ao degrau para o sistema em malha fechada é calculada e, a partir dela, são adquiridos o sobressinal, o tempo de acomodação, o tempo de subida e o erro estacionário para o cálculo da função de custo. Um algoritmo genético pode ser utilizado para otimizar os ganhos do controlador PID, como apresentado no Algoritmo 2. Uma população de 25 indivíduos é produzida em cada geração (total de 10 gerações).

Algoritmo 1 Análise da função de custo para o controlador PID

Function pidtest(*G*, *dt*, *parms*)

```

/* Função de transferência do controlador PID */
K ← parms(1) + parms(2)/s + parms(3) * s/(1 + .001 * s);

/* Resposta ao degrau do sistema em malha fechada */
Loop ← series(K, G);
ClosedLoop ← feedback(Loop, 1);
t ← 0 : dt : 20;
[y, t] ← step(ClosedLoop, t);
infStep ← stepinfo(ClosedLoop); /* Informações da resposta ao degrau */
β ← 1.5; /* Fator de ponderação */

/* Função de custo */
J ← (1 - e-β) * (infStep.Overshoot + (infStep.SettlingMax
- infStep.SettlingMin)/2) + e-β * (infStep.SettlingTime
- infStep.RiseTime);
return J

```

Algoritmo 2 Algoritmo Genético para sintonização do controlador PID

```

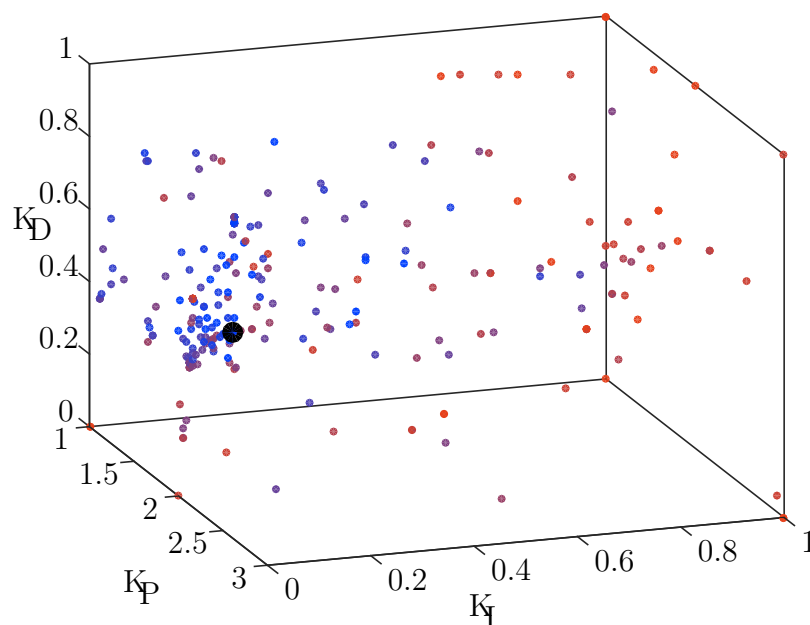
/* Inicialização */
dt ← 0.001;
PopSize ← 25;
MaxGenerations ← 10;
G ← modelo do manipulador;

/* Otimização dos ganhos do controlador PID utilizando GA */
options ← gaoptimset('PopulationSize', PopSize, 'Generations', MaxGenerations);
[x, fval] ← ga(@(K)pidtest(G, dt, K), 3, -eye(3), zeros(3, 1), [], [], [], [], [], options);

```

Os ganhos gerados para o controlador PID são ilustrados na Figura 6.4, onde os pontos mais vermelhos correspondendo as gerações iniciais e os pontos mais azuis correspondendo as gerações posteriores e o ponto preto correspondendo a solução ótima. Com a progressão do algoritmo genético, os ganhos começam a convergir para uma solução ótima. A solução encontrada pelo algoritmo genético foi: $K_P = 2,8$, $K_I = 3,31e - 05$ e $K_D = 0,56$. Com o ganho integral próximo de zero, a solução corresponde a um controlador PD. Portanto, a abordagem de controle investigada e referida de agora em diante é o método de controle PD com ganhos $K_P = 2,8$ e $K_D = 0,56$.

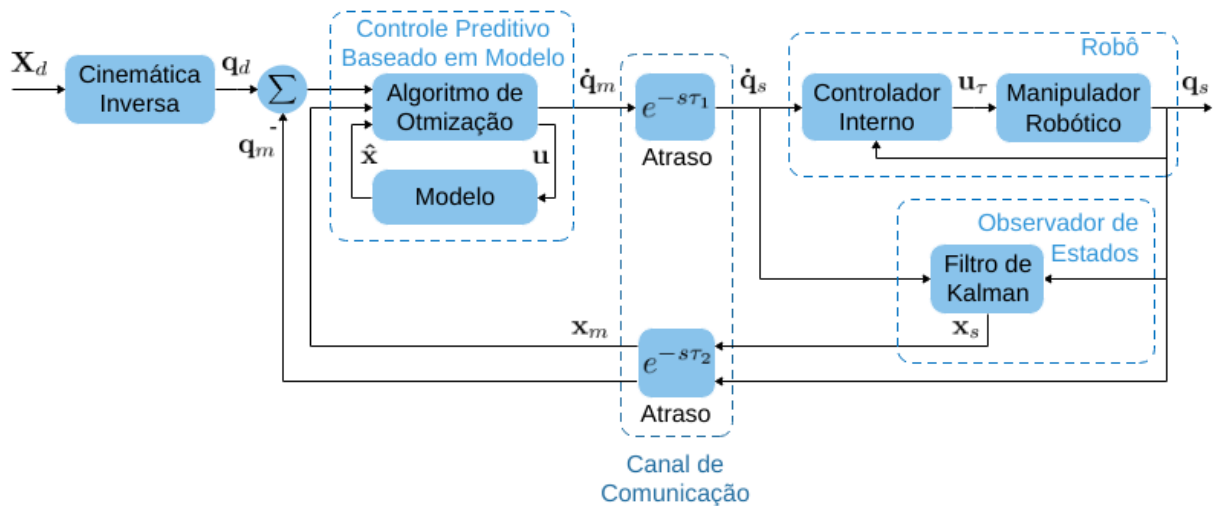
Figura 6.4: Ganhos do controlador PID gerados pelo algoritmo genético. O ponto preto é a solução ótima encontrada pelo algoritmo genético.



Fonte: Elaborada pelo autor.

A fim de obter um controlador com melhor desempenho, uma estratégia de controle preditivo baseado em modelo por realimentação de estados, *Model Predictive Control* (MPC), é investigado e comparado com o controlador clássico. O controle preditivo baseado em modelo utiliza um modelo da planta para prever saídas futuras do sistema. Uma sequência de valores futuros para a variável manipulada é obtida a partir das previsões realizadas, a uma taxa de amostragem, de forma a otimizar a saída do sistema. Na Figura 6.5 ilustra-se o diagrama do sistema com controle preditivo baseado em modelo, onde: \hat{x} são os estados preditos do modelo do sistema; u são as entradas do modelo; x_m e x_s os estados observados no lado mestre e no lado escravo, respectivamente. A diferença desse diagrama para o diagrama com controle clássico está na substituição pelo controlador preditivo no lado mestre, que possui um algoritmo de otimização e um modelo, e como o sistema possui realimentação de estados, no lado controlado um filtro de Kalman é adicionado para prover a estimação dos estados do sistema.

Figura 6.5: Diagrama do sistema com controle preditivo baseado em modelo.



Fonte: Elaborada pelo autor.

O controle preditivo baseado em modelo utiliza uma otimização numérica online para calcular o sinal de controle ao ser aplicado no processo. A cada amostragem, o controlador MPC prevê os estados futuros do sistema até um horizonte de previsão especificado no modelo (PANE, 2015). A função de custo quadrática utilizada nessa pesquisa é descrita por:

$$J_k = \sum_{i=k}^{k+N_p-1} x_i^T Q_x x_i + u_i^T Q_u u_i \quad (6.12)$$

onde x é variável de estado, u é a entrada do processo, Q_x e Q_u são as matrizes de ponderação para a função de custo e N_p é o horizonte de previsão/controle. A cada amostragem, o controlador deve utilizar a entrada e saída da planta para calcular o estado

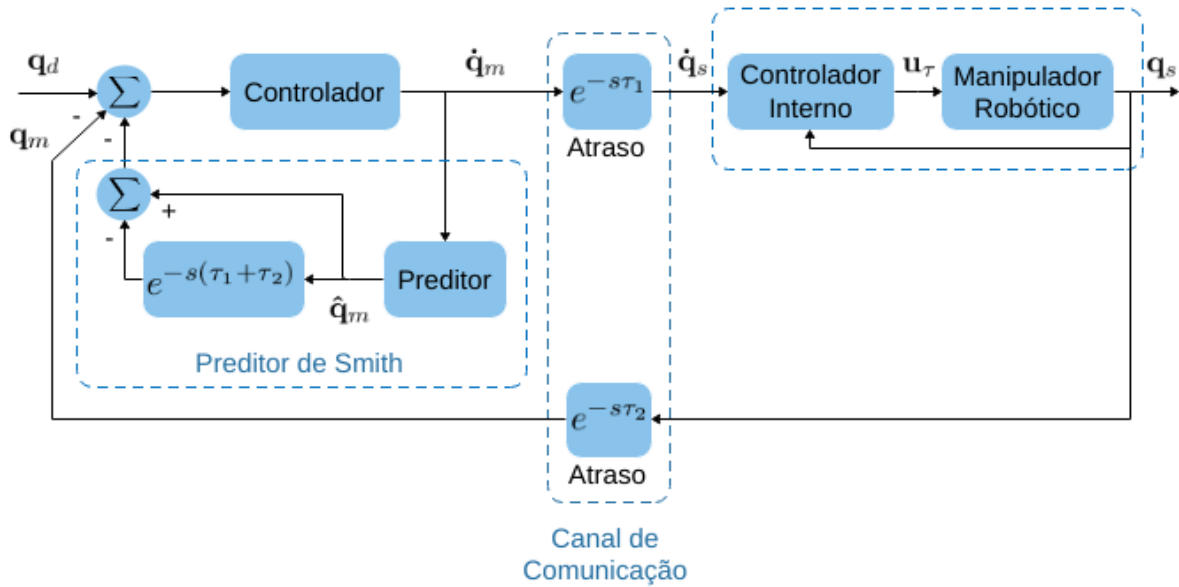
x_i que será utilizada como entrada na otimização. Um observador de estados (filtro de Kalman) é então utilizado para esse fim. Em seguida, o otimizador utiliza o erro entre os estados real e desejado para prever e otimizar o comportamento da planta e gerar uma sequência de sinais de entrada que minimizam a função de custo. Entretanto, apenas o primeiro sinal de entrada, sinal de controle, é utilizado no processo para garantir a robustez a distúrbios e incertezas (SKARIN et al., 2019). Os parâmetros para ajuste do controlador MPC são o horizonte de predição/controle N_p e as matrizes de ponderação Q_x e Q_u . Uma abordagem heurística foi realizada para ajuste dos parâmetros do controlador MPC. A priori, o horizonte de predição foi ajustado para 10, pois valores acima de 10 aumentaram significativamente o tempo de processamento do laço de controle, uma vez que mais cálculos são necessários para realizar as previsões e, conseqüentemente, maior o esforço computacional requerido. Q_x e Q_u foram inicializados com valores pequenos e, em seguida, os termos em Q_x foram aumentados gradualmente com o intuito de diminuir os erros em regime estacionário e o tempo de subida do controlador. Como resultado, os termos diagonais da matriz Q_x foram definidos como sendo 1000 e os termos diagonais da matriz Q_u foram definidos como sendo 1.

6.3 Abordagens de Predição

Com o intuito de investigar uma solução baseada em modelo, primeiramente foi implementado um sistema com preditor de Smith, discutido anteriormente no Capítulo 3. Com os atrasos de comunicação, o desempenho do sistema diminui, por consequência, a instabilidade do sistema aumenta. A configuração com preditor de Smith, ilustrado Figura 6.6, permite superar a instabilidade no sistema causada pelo atraso de comunicação no laço de controle. O atraso de ida da comunicação é τ_1 e o atraso de retorno da comunicação é τ_2 , dessa forma o atraso total do canal de comunicação do sistema é de $\tau_1 + \tau_2$. A abordagem com preditor de Smith necessita de um modelo do sistema controlado, portanto, quanto mais aproximado esse modelo for do sistema real, melhor será o desempenho do preditor. O manipulador UR5 é não-linear, dessa forma, um modelo não-linear do manipulador tem uma resposta melhor do que um modelo linear. Para aproximar essa resposta não-linear do manipulador, uma rede neural artificial é investigada. Redes neurais artificiais já são bastantes difundidas e a sua capacidade de modelar e estimar funções não lineares fazem delas uma ótima ferramenta na identificação de sistemas (TUTUNJI, 2016). No lado controlado, utiliza-se um estimador para identificar o modelo não-linear baseado em Rede Neural do manipulador, o modelo estimado é então reproduzido no lado escravo para realizar as predições, para isso, os pesos da Rede Neural Artificial são enviados pelo canal de comunicação. A estimação é feita de forma online e os pesos são atualizados

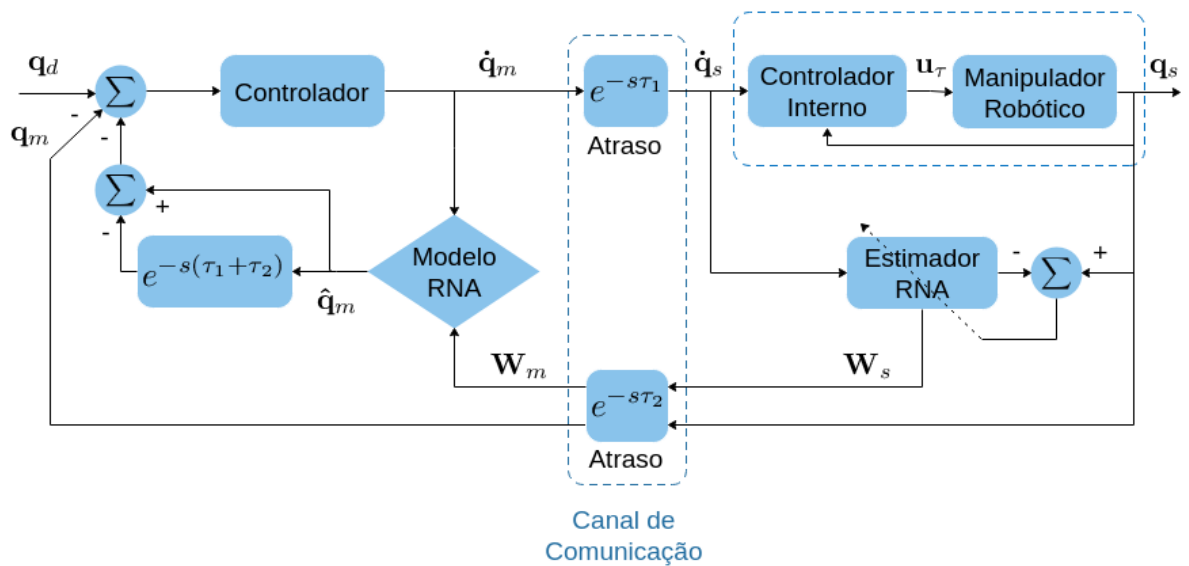
frequentemente. Na Figura 6.7 ilustra-se o diagrama do sistema com preditor de Smith baseado em Rede Neural Artificial.

Figura 6.6: Diagrama do sistema com preditor de Smith.



Fonte: Elaborada pelo autor.

Figura 6.7: Diagrama do sistema com preditor de Smith baseado em Rede Neural Artificial.

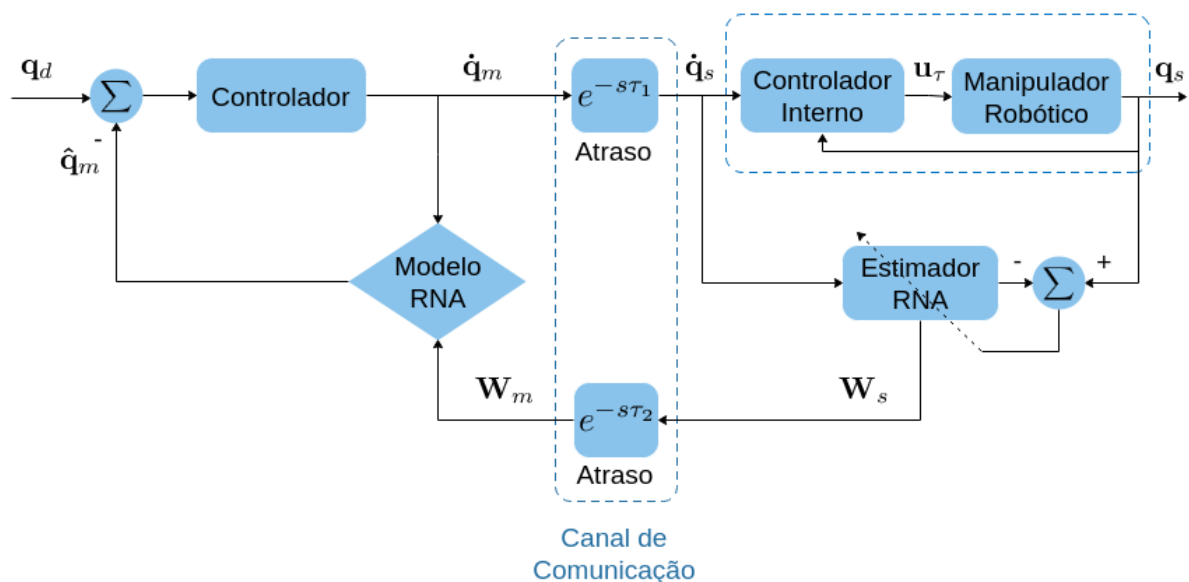


Fonte: Elaborada pelo autor.

O esquema anterior segue a estrutura do preditor de Smith, onde além do sinal de realimentação vindo com atraso do lado controlado, é necessário conhecer o atraso de

comunicação na rede para mitigar o efeito do atraso no sinal de realimentação. Além do mais, se a modelagem da manipulador não for precisa, então não haverá o cancelamento completo do sinal de realimentação provocando instabilidades e também dificultando o controle do sistema. Uma abordagem sem realimentação é ilustrada na Figura 6.8. Nessa abordagem a realimentação é provinda do modelo do manipulador e não mais do manipulador no lado controlado. Dessa forma, elimina-se o fato de conhecer a priori o atraso no canal de comunicação. Essa abordagem traz uma maior transparência para o sistema. Embora vantajosa, se a atualização do modelo no lado mestre não for realizada de forma correta e/ou a modelagem do sistema não for precisa, então haverá grandes instabilidades, prejudicando o controle e transparência do sistema.

Figura 6.8: Diagrama do sistema com preditor de Smith baseado em Rede Neural sem realimentação.



Fonte: Elaborada pelo autor.

Para realização das abordagens de predição discutidas previamente, é necessário a utilização de redes neurais artificiais que descrevem o modelo do manipulador. Redes Neurais Artificiais (RNA) são modelos matemáticos que são inspirados pelo neurônio biológico. Uma rede neural artificial pode ser definida, segundo Kevin Gurney, como:

Uma rede neural é um conjunto interconectado de elementos, unidades ou nós de processamento simples, cuja funcionalidade é vagamente baseada no neurônio animal. A capacidade de processamento da rede é armazenada nas intensidades de conexão entre unidades, ou pesos, obtidos por um processo de adaptação ou aprendizado de um conjunto de padrões de treinamento (GURNEY, 1997, p.13).

As RNAs são altamente adaptáveis e capazes de capturar comportamentos complexos, aprendendo e armazenando informações. Elas podem ser usadas como modelos caixa preta para identificar funções, realizando o mapeamento não linear entre os dados de entrada e saída. Portanto, as RNAs podem ser utilizadas para realizar a identificação de sistemas.

Objetivando desenvolver um modelo do manipulador no lado mestre do sistema, uma abordagem com RNA recursiva é investigada. Para isso, uma rede neural baseada em modelo autorregressivo com entradas exógenas, *Autoregressive Exogenous* (ARX), é utilizada para realizar as previsões da resposta do manipulador. Modelos ARX são utilizados para realizar a modelagem de sistemas dinâmicos e séries temporais. A saída observada consiste da soma da regressão de observações anteriores da entrada e saída do processo (DIVERSI; GUIDORZI; SOVERINI, 2010). Um modelo ARX discreto no tempo é descrito por:

$$y_{k+1} = \sum_{i=1}^m a_i y_{k-i+1} + \sum_{i=1}^n b_i x_{k-i+1} \quad (6.13)$$

onde x e y são a entrada e a saída do modelo, a e b são os parâmetros do modelo, m e n são os horizontes passados da entrada e saída do modelo e k é a amostra. A predição pode ser realizada de maneira online por meio de uma arquitetura série-paralela ou offline por meio de uma arquitetura paralela. Na predição online as saídas passadas utilizadas no modelo são adquiridas do processo real, enquanto que na predição offline as saídas passadas utilizadas no modelo são as saídas passadas preditas pelo modelo ARX.

Nesse trabalho, para realizar a abordagem baseada em modelo apresentada na Figura 6.8, é necessário a utilização de um modelo de RNA que não empregue as saídas reais observadas do processo para a predição da amostra futura. Para isso, a priori, um modelo ARX com arquitetura do tipo paralela foi investigado, ou seja, as saídas passadas preditas pela rede neural ARX foram utilizadas para a predição da resposta futura, entretanto o erro na predição aumentou consideravelmente. Logo, uma rede neural ARX com predição online é avaliada. A utilização de uma arquitetura série-paralela possui a vantagem de ser mais precisa, já que valores reais observados do processo são utilizados como entrada da rede, outra vantagem é que a rede não possui laço de realimentação e, portanto, são redes neurais alimentadas adiante (*feedforward*). Dessa forma, podem ser empregados os algoritmos utilizados para treinamento de redes perceptron multicamada, *Multi-Layer Perceptron* (MLP) (BOUSSAADA et al., 2018).

Um neurônio artificial simples, ilustrado na Figura 6.9, é descrito matematicamente por:

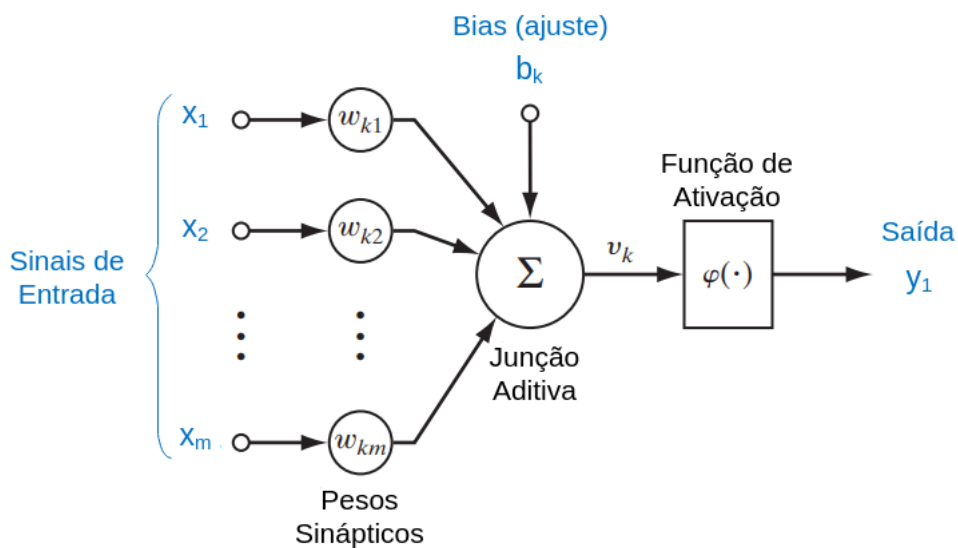
$$u_k = \sum_{p=1}^m w_{kp} x_p \quad (6.14)$$

$$v_k = u_k + b_k \tag{6.15}$$

$$y_k = \phi(v_k) \tag{6.16}$$

onde x_1, \dots, x_m são os sinais de entrada, w_{k1}, \dots, w_{km} são os pesos sinápticos do neurônio k , u_k é a saída do combinador linear dado o sinal de entrada, v_k é a saída u_k com a aplicação do ajuste b_k , $\phi(\cdot)$ é a função de ativação e y_k é o sinal de saída do neurônio. A função de ativação defini a saída do neurônio em função de v_k , dentre as funções básicas pode-se listar a função de ativação linear, sigmóide e tangente hiperbólica.

Figura 6.9: Modelo de um neurônio artificial.



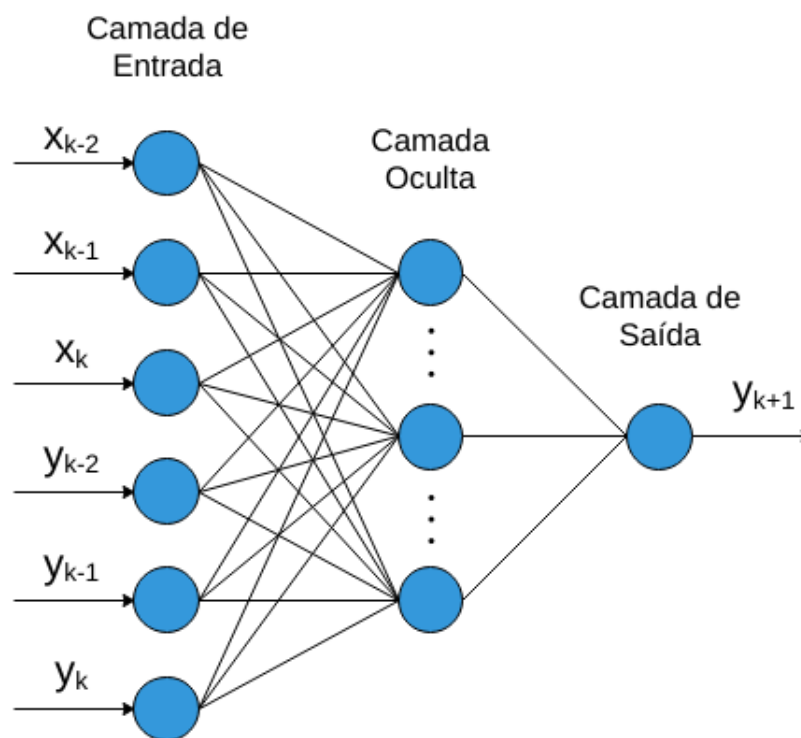
Fonte: Adaptado de Haykin (2010, p.11).

Redes perceptron multicamada são compostas por uma camada de entrada, uma ou mais camadas de neurônios ocultos e uma camada de neurônios de saída. A informação flui da camada de entrada para a camada de saída. Para realizar o treinamento da rede neural pode-se utilizar o algoritmo de retropropagação de erro (*backpropagation*). O algoritmo de retropropagação de erro utiliza os dados de entrada e saída para treinar a rede. O algoritmo consiste da fase de propagação (*forward*) e da fase de retropropagação (*backward*). Na fase de propagação os pesos sinápticos permanecem inalterados e a entrada é aplicada para aquisição dos sinais de saída e dos erros. Na fase de retropropagação os erros adquiridos são retropropagados pela rede para ajuste dos pesos sinápticos que minimizem o erro obtido de acordo com um método de otimização. Uma época consiste do treinamento de todo o conjunto de amostras. O treinamento é realizado época a época até que o número de épocas desejada tenha sido alcançada ou até que algum critério de parada tenha sido atingido (HAYKIN, 2010).

A arquitetura de rede neural artificial utilizada é apresentada na Figura 6.10. A

rede neural consiste da camada de entrada, a camada oculta e a camada de saída. O horizonte passado escolhido para o modelo de rede neural ARX é definido como três, logo na camada de entrada da rede neural tem-se seis dados que correspondem ao dados de entrada e saída passados, velocidade e posição respectivamente, adquiridos do processo. Foram avaliadas mais de uma camada oculta, entretanto apenas uma camada revelou-se suficiente. Também foram avaliados diferentes quantidades de neurônios na camada oculta, a utilização de dez neurônios na camada oculta apresentou um erro menor na resposta de predição. Por fim, a ultima camada é composta apenas por um neurônio cuja saída é a posição futura da junta do manipulador.

Figura 6.10: Arquitetura da rede neural artificial.



Fonte: Elaborada pelo autor.

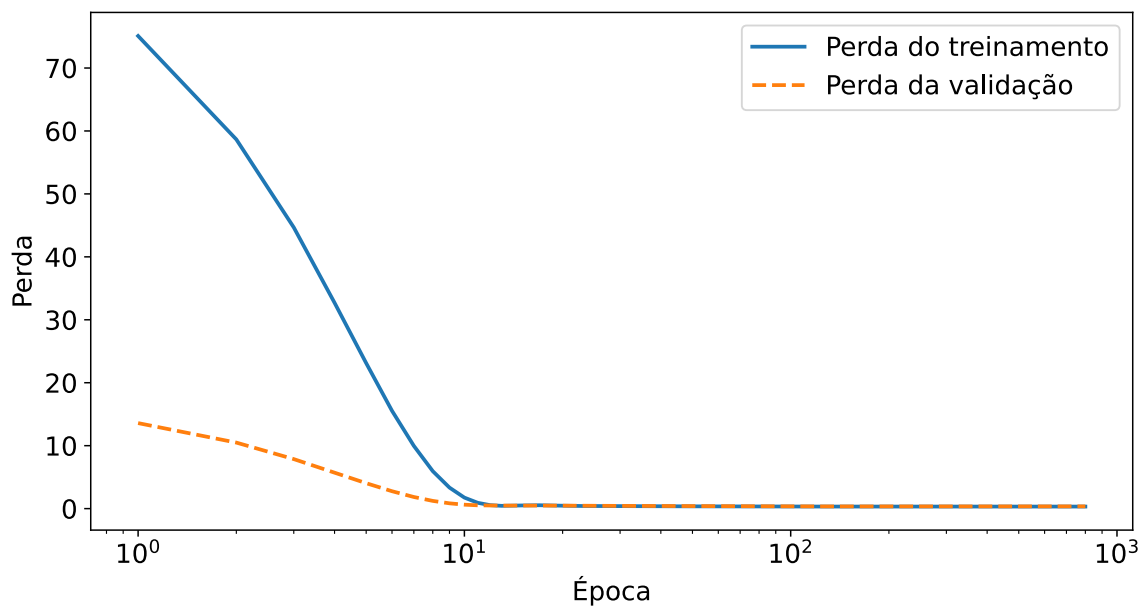
Para realizar o treinamento da rede neural ARX, um conjunto de treinamento e um conjunto de validação foram separados. Para isso, um sinal de entrada do tipo PRBS (apresentado na Seção 6.1) foi utilizado para adquirir os dados de entrada e saída do manipulador. Os dados foram adquiridos a cada 40 ms, resultando em um total de 5000 amostras. Também foi separado um conjunto de teste com 2000 amostras para avaliar o modelo de rede neural ARX treinada. A rede pode ser treinada para realizar previsões em um horizonte futuro n , sendo necessário apenas organizar a representação dos dados para treinamento e validação. Dados os conjuntos de dados adquiridos, para

um horizonte de predição igual a 10 (previsão de 200ms) a rede foi treinada por 800 épocas com função de ativação linear. O método de otimização escolhido foi um método de gradiente descendente estocástico. Foram utilizadas as bibliotecas de código aberto TensorFlow¹ e Keras² para treinamento e simulação da rede neural. Na Figura 6.11 ilustra-se as perdas do treinamento e validação. A saída real e predita para o conjunto de teste é apresentada na Figura 6.12. Para avaliar o desempenho da rede foi calculado o erro relativo percentual

$$\varepsilon = \frac{|Valor Real - Valor Predito|}{Valor Real} \times 100\%. \quad (6.17)$$

O erro obtido foi de 5,51 %. O erro tende a aumentar com o aumento do horizonte de predição, bem como tende a diminuir com a diminuição do horizonte de predição.

Figura 6.11: Curvas de perdas do treinamento e validação da rede neural ARX para um horizonte de predição igual a 10.



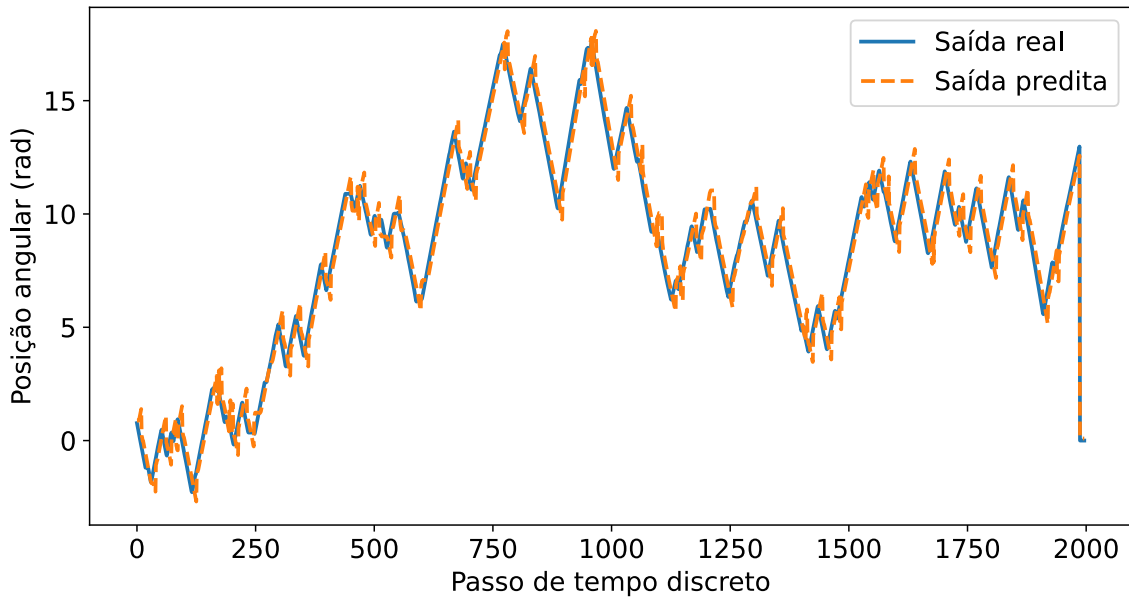
Fonte: Elaborada pelo autor.

O modelo apresentado na Figura 6.8 precisa ser adaptado para a utilização da rede neural ARX. O diagrama do sistema com o preditor baseado em rede neural ARX é apresentado na Figura 6.13. O lado controlado envia os dados de entrada e saída do manipulador para o lado mestre para que a rede neural realize a predição.

¹<https://www.tensorflow.org>

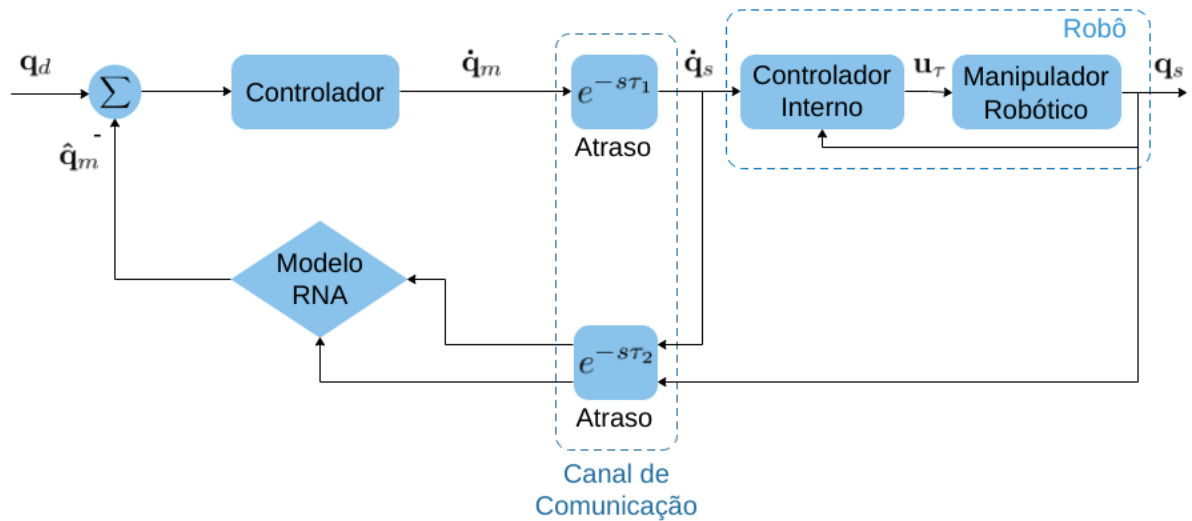
²<https://keras.io/>

Figura 6.12: Representação da saída real e saída predita para o conjunto de teste utilizando a rede neural ARX com horizonte de predição igual a 10.



Fonte: Elaborada pelo autor.

Figura 6.13: Sistema com o preditor baseado em rede neural ARX.



Fonte: Elaborada pelo autor.

6.4 Considerações Finais

Neste Capítulo foram apresentadas as abordagens de controle e inteligência artificial preditivas investigadas no âmbito desta pesquisa. A priori é realizada uma descrição a respeito da identificação do modelo do manipulador utilizando um método baseado em

identificação do subespaço. Em seguida, a arquitetura de controle no espaço das juntas é apresentado. A estratégia de controle PID foi investigada e os ajustes dos parâmetros resultaram em um controlador PD como uma solução ótima do objetivo de controle. Portanto, a abordagem é referida no restante do trabalho por controle PD. A abordagem de controle MPC também foi discutida e, por fim, foram apresentadas as estratégias de inteligência artificial preditiva investigadas.

No Capítulo 7 são apresentadas as simulações das abordagens de controle e predição que validam o sistema desenvolvido e realizada a análise dos resultados obtidos com as simulações.

Capítulo 7

Simulações e Resultados

Neste Capítulo, simulações das abordagens de controle e inteligência artificial propostas no Capítulo 6 são realizadas com o sistema proposto no Capítulo 5 afim de validar e testar o desempenho do sistema bem como das abordagens de controle e predição. Na Seção 7.1 é apresentada a configuração experimental utilizada para realizar as simulações. As abordagens de controle propostas no Capítulo 6, controlador PD e MPC, são avaliadas na Seção 7.2. Por fim, na Seção 7.4 são apresentadas as considerações finais acerca do capítulo.

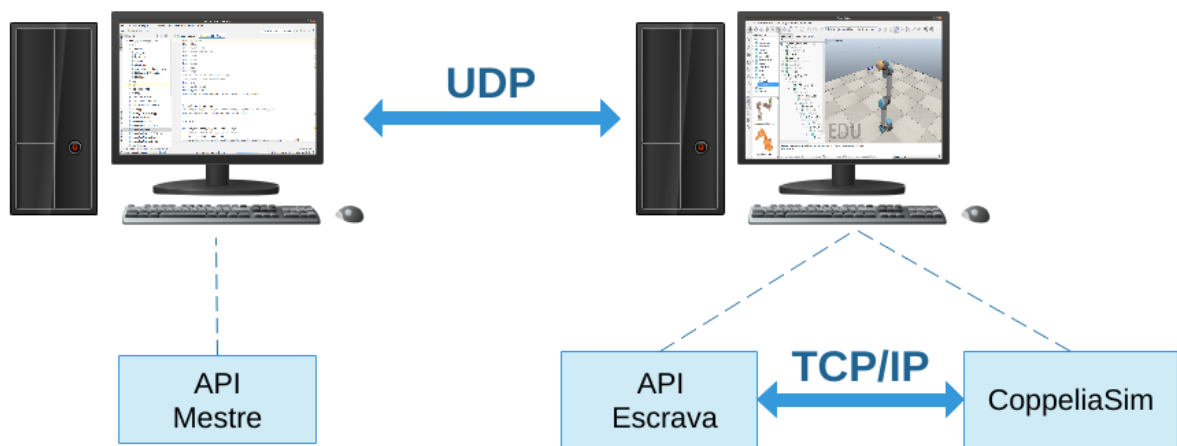
7.1 Configuração Experimental

A configuração experimental estabelecida para validar o sistema e avaliar as abordagens de controle e predição é apresentada na Figura 7.1. O domínio mestre e domínio controlado são simulados em computadores diferentes com distribuição Linux que representam as bordas do sistema no contexto de Internet Tátil. O domínio de rede é estabelecido por meio de dois canais de comunicação UDP, um canal destinado ao sinal de comando e outro canal destinado ao sinal de resposta. A mensagem enviada por cada canal depende do tipo de abordagem de controle e predição utilizado no sistema. Como mencionado no Capítulo 5, o formato de mensagem JSON é utilizado para realizar a troca de dados entre as bordas. No lado mestre, a API mestre, escrita em Python, é responsável por gerar o sinal de referência (ou a trajetória) que representa o dado háptico/cinemático produzido pelo sistema mestre. Além disso, a API mestre também implementa a estratégia de controle cinemático de alto nível do robô. Por sua vez, no lado escravo (domínio controlado) é realizada a execução da API escrava e do simulador virtual CoppeliaSim com a cena criada para representar o braço robótico e o ambiente. A API escrava é responsável por realizar a interface de comunicação com o simulador virtual e a computação necessária pela abordagem utilizada. A comunicação entre a API escrava e a cena no

CoppeliaSim é realizada por meio da Remote API utilizando o protocolo de comunicação TCP/IP.

O atraso nos canais de comunicação no domínio de rede pode ser especificado e controlado utilizando a ferramenta *Traffic Control* (TC) que permite configurar o agendador de pacotes.

Figura 7.1: Diagrama da configuração experimental.



Fonte: Elaborada pelo autor.

7.2 Desempenho das Abordagens de Controle

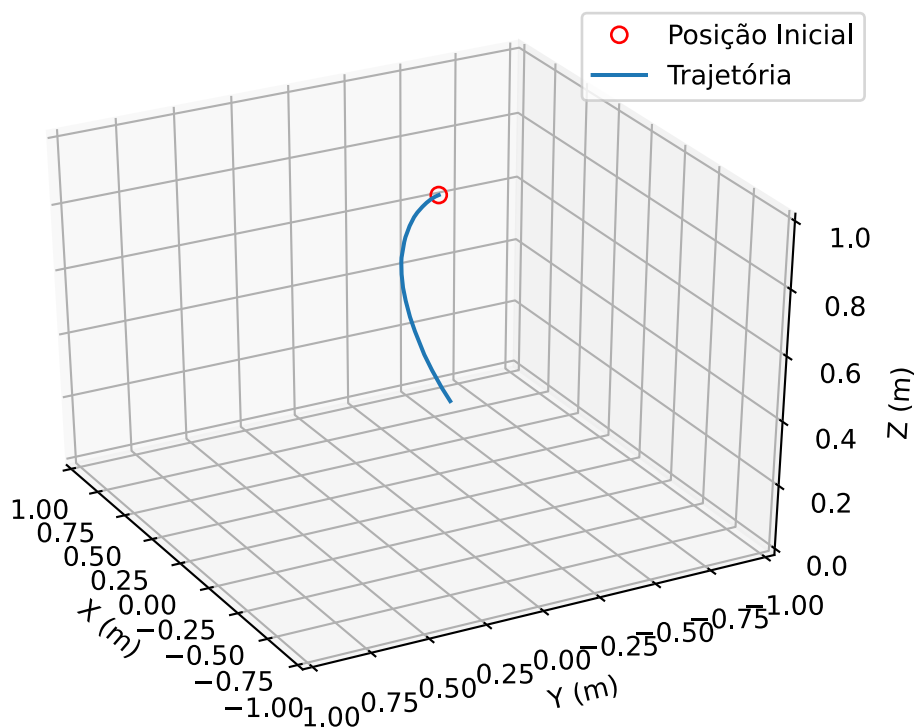
A resposta do controlador e a estabilidade do sistema está fortemente relacionada ao tempo de execução do laço nas aplicações mestre e escravo. Dessa forma, primeiramente é definido o parâmetro Δ para representar o tempo de espera em um laço de controle. Segundo (POLACHAN et al., 2019), um sistema que possui um Δ muito pequeno pode responder mais rapidamente ao sinal com tempos de subida menores. Entretanto, um Δ muito pequeno pode também levar a oscilações na resposta do sistema. No sistema aqui proposto, o valor do parâmetro Δ é altamente correlacionado com o desempenho computacional e também com a abordagem de controle utilizada, uma vez que uma abordagem que possua mais operações exige um maior empenho computacional e, conseqüentemente, possui um maior atraso em sua resposta.

É importante mencionar que o manipulador é controlado via velocidade das juntas, portanto o controlador atua em cada junta do sistema utilizando um sinal de controle de velocidade. O controlador MPC apresentou um tempo de execução no laço de controle maior que o controlador PD, com $\Delta = 15 \text{ ms}$ para o controlador MPC e $\Delta = 2 \text{ ms}$ para o controlador PD. Dessa forma, a fim de comparar o desempenho dos controladores no

rastreamento da posição, o tempo de execução do laço no controle PD foi ajustado para 15 ms .

A fim de avaliar os controladores, um sinal de referência do tipo degrau é aplicado ao sistema. O atraso padrão do canal de comunicação, em torno de $0,7\text{ ms}$, é mantido. A pose inicial do manipulador corresponde a configuração inicial do manipulador no simulador CoppeliaSim, que resulta na posição do efetor-final de $(-0,19, 0, 1)\text{ m}$ no \mathbb{R}^3 . A referência é um degrau de 1 rad na junta 1 e 2 do manipulador, a posição angular das demais juntas permanece constante. Os ganhos do controlador PD foram ajustados para $K_P = 2,8$ e $K_D = 0,56$, e os parâmetros do controlador MPC foram ajustados de acordo com a sintonia apresentada no Capítulo 6. A saída do controlador (sinal de controle) foi limitada à $\pi\text{ rad/s}$ de acordo com a velocidade máxima da junta do manipulador robótico. A trajetória executada pelo manipulador é ilustrada na Figura 7.2.

Figura 7.2: Representação da trajetória realizada pelo manipulador UR5.



Fonte: Elaborada pelo autor.

A resposta do sistema ao degrau na junta 1 utilizando o controlador PD e utilizando o controlador MPC são ilustradas na Figura 7.3 e 7.4, respectivamente. A resposta na junta 2 é semelhante, já que é utilizado o mesmo controlador e o comportamento cinemático das juntas são iguais no software de simulação. Como as demais juntas não tem sua posição angular modificada suas respostas não foram analisadas. Nas Figuras 7.5 e 7.6 ilustram-

se as respostas obtidas nos eixos x, y e z no espaço cartesiano para os controladores PD e MPC, respectivamente. O atraso do sistema é de aproximadamente $50ms$. Os picos obtidos na resposta do sistema no eixo y para ambos os controladores são ocasionados devido o controle ser realizado no espaço das juntas e também, em razão de uma trajetória não ter sido especificada e sim um sinal de referência do tipo degrau.

Com o intuito de avaliar o desempenho dos controladores, é computada a medida do erro em estado estacionário (e_{ss}), a medida do erro médio absoluto, *Mean Absolute Error* (MAE), descrito por

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (7.1)$$

e a medida da raiz do erro quadrático médio, *Root Mean Square Error* (RMSE), descrito por

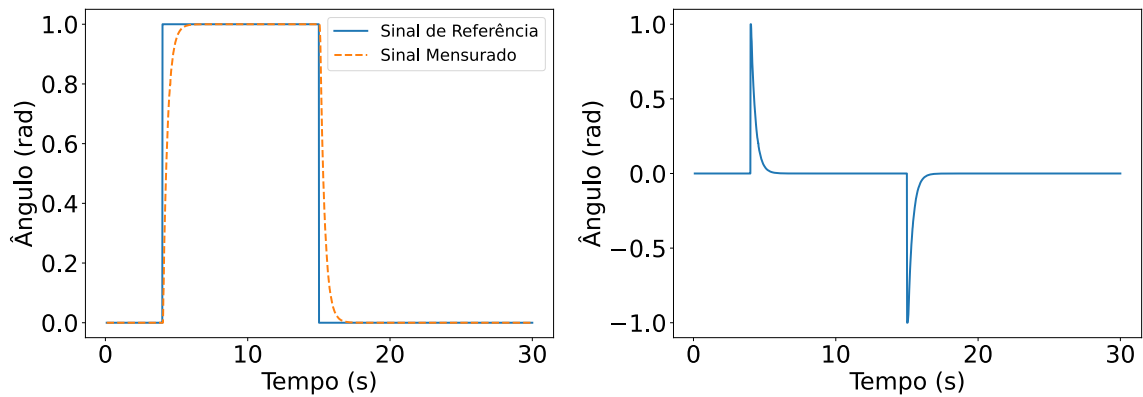
$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}, \quad (7.2)$$

onde n é o número de dados, y_i é o i -ésimo valor do sinal de referência e \hat{y}_i é o i -ésimo valor do sinal mensurado. O MAE e o RMSE possuem a unidade igual a dimensão dos valores observados e suas respostas representam o desvio médio entre o sinal de referência e o sinal mensurado, entretanto, o RMSE varia de acordo com a variação da magnitude do erro, enquanto o MAE penaliza os erros de forma equivalente. Portanto o RMSE sempre é maior que o MAE (CHAI; DRAXLER, 2014). Os dados necessários para o cálculo dos erros foram obtidos no domínio mestre. Na Tabela 7.1 são apresentados os valores MAE, RMSE e e_{ss} obtidos para os controladores PD e MPC na junta 1 e nos eixos x, y e z do espaço cartesiano. Ambos os controladores obtiveram êxito no rastreamento do sinal de referência. Conforme apresentado, o controlador PD obteve melhores resultados no rastreamento do sinal de referência, apresentando erros menores em relação ao controlador MPC. Isso pode ser explicado devido a sintonia realizado nos parâmetros do controlador para encontrar os ganhos que minimizassem a função de custo abordada no Capítulo 6. Além do mais, os parâmetros do controlador MPC foram sintonizados utilizando uma abordagem heurística, o que pode resultar em uma solução que não seja ótima. Um ajuste dos parâmetros do controlador MPC pode ser realizado utilizando algoritmo genético assim como foi feito para o controlador PD, entretanto, como o controlador PD apresentou uma resposta satisfatória e o esforço computacional mostrou-se bem menor em relação ao controlador MPC, decidiu-se adotar o controlador PD no sistema para investigação das técnicas de predição e indicar o ajuste dos parâmetros do controlador MPC como trabalhos futuros no Capítulo 8.

O sistema apresentou um bom desempenho de controle já que foi obtido êxito na tarefa do rastreamento do sinal de referência com a execução das abordagens de controle no sistema. Além disso, utilizando o controlador PD, foram aplicados atrasos de 0,7 ms,

100 ms, 200 ms, 400 ms, e 500 ms nos canais de comunicação, no domínio da rede, para avaliar o comportamento do sistema. As respostas obtidas são apresentadas na Figura 7.7. Como esperado houve atraso na resposta ao degrau de acordo com o atraso aplicado no canal de comunicação. Além disso, as respostas obtidas comprovam o aumento da instabilidade do sistema com o aumento do atraso no canal de comunicação.

Figura 7.3: Resposta do sistema ao degrau na junta 1 com o controlador PD

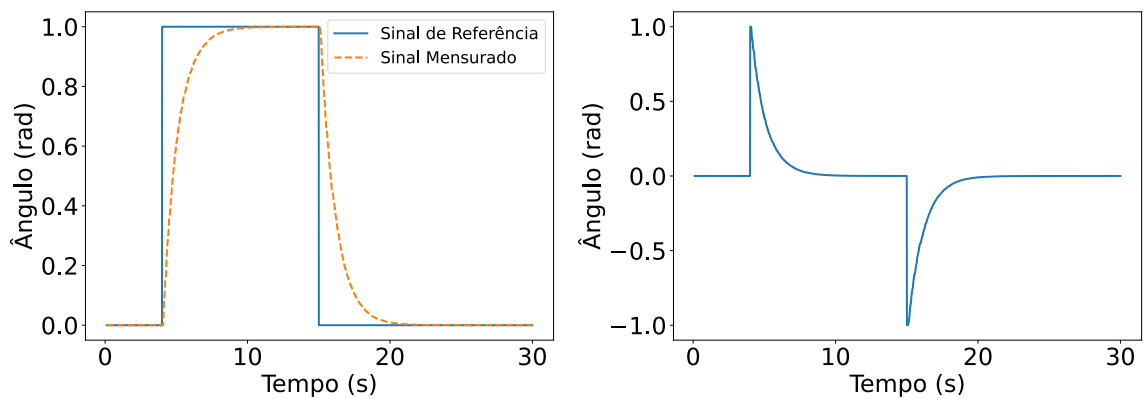


(a) Posição angular da junta 1.

(b) Erro absoluto da posição angular da junta 1.

Fonte: Elaborada pelo autor.

Figura 7.4: Resposta ao degrau para o controlador MPC

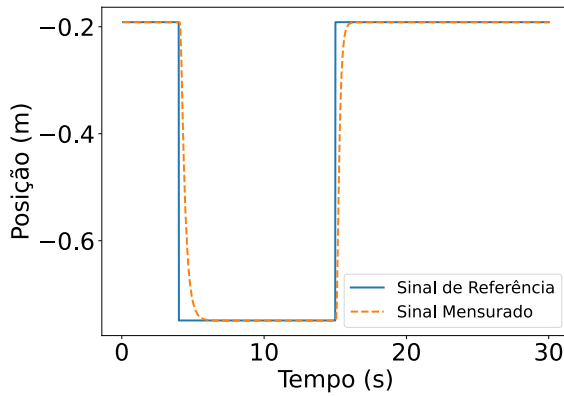


(a) Posição angular da junta 1.

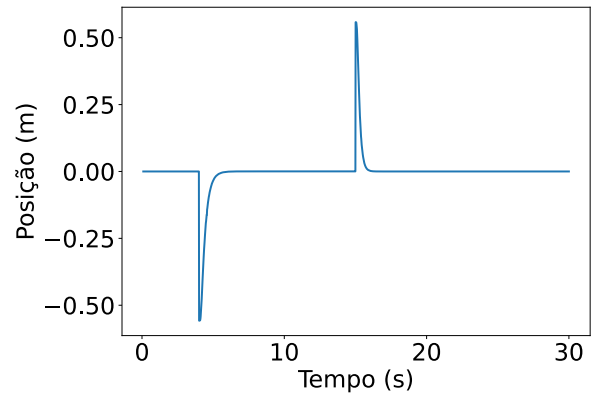
(b) Erro absoluto da posição angular da junta 1.

Fonte: Elaborada pelo autor.

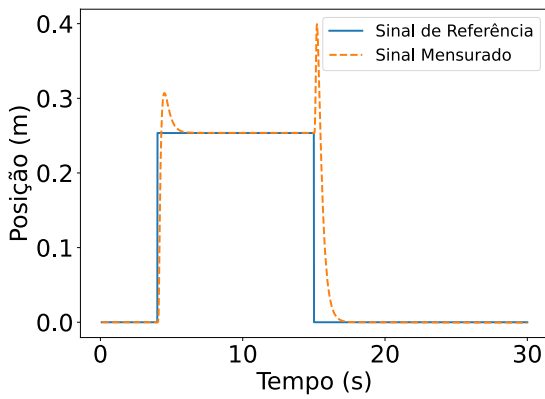
Figura 7.5: Resposta ao degrau para o controlador PD



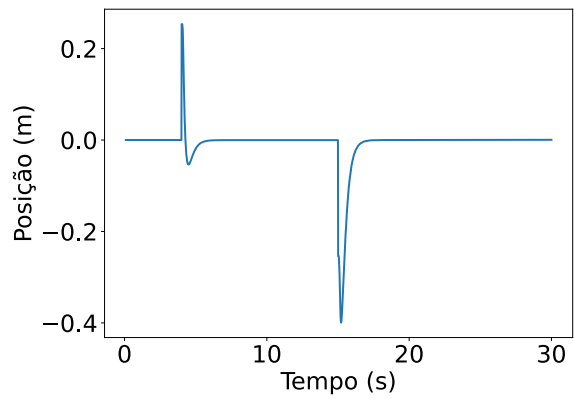
(a) Posição no eixo x



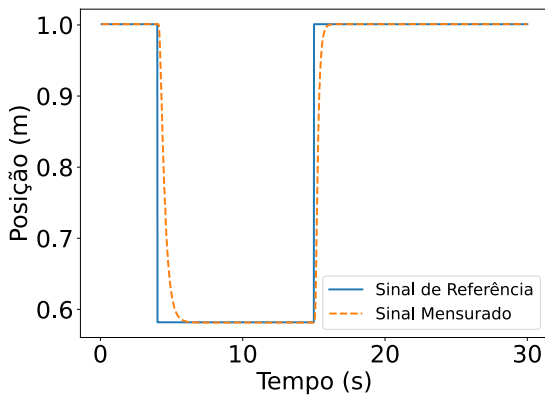
(b) Erro absoluto da posição no eixo x



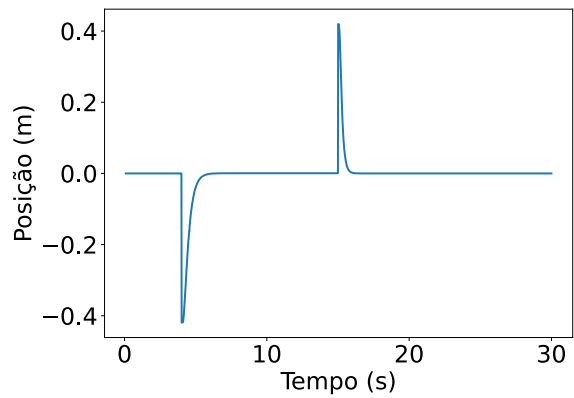
(c) Posição no eixo y



(d) Erro absoluto da posição no eixo y



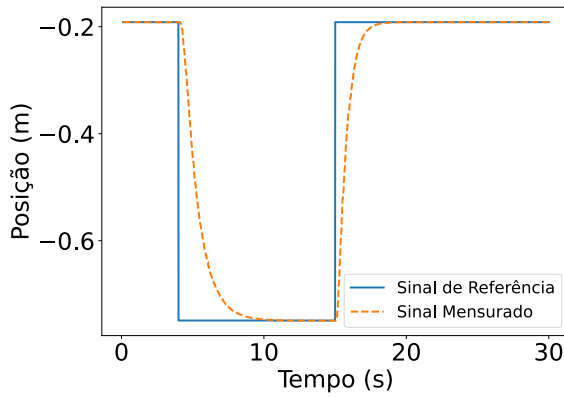
(e) Posição no eixo z



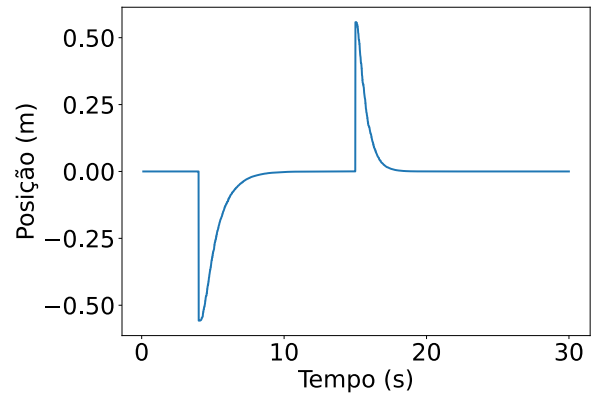
(f) Erro absoluto da posição no eixo z

Fonte: Elaborada pelo autor.

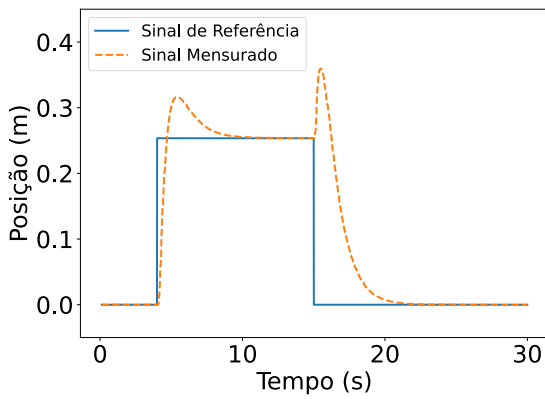
Figura 7.6: Resposta ao degrau para o controlador MPC



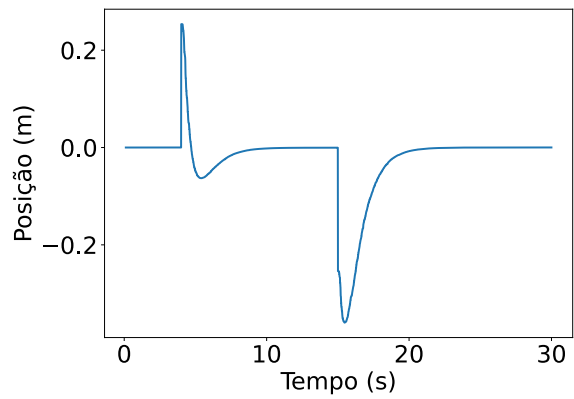
(a) Posição no eixo x



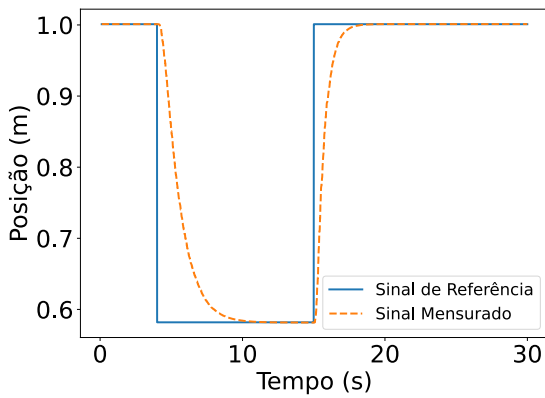
(b) Erro absoluto da posição no eixo x



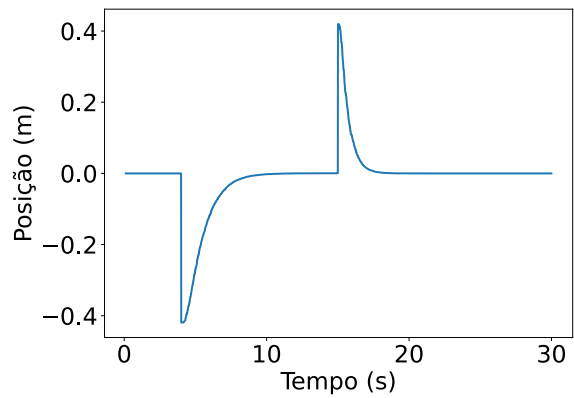
(c) Posição no eixo y



(d) Erro absoluto da posição no eixo y



(e) Posição no eixo z



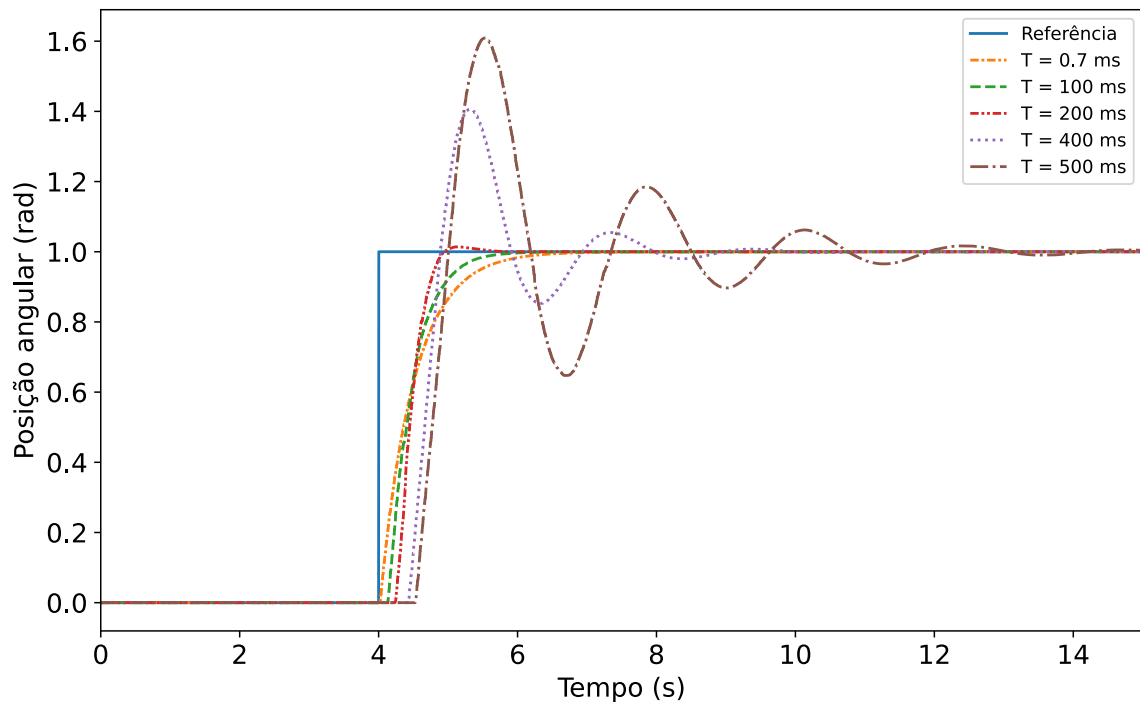
(f) Erro absoluto da posição no eixo z

Fonte: Elaborada pelo autor.

Tabela 7.1: Valores MAE, RMSE e e_{ss} obtidos para os controladores PD e MPC na junta 1 e nos eixos x, y e z do espaço cartesiano.

| Posição | MAE | | RMSE | | e_{ss} | |
|----------------------|--------|--------|--------|--------|------------|------------|
| | PD | MPC | PD | MPC | PD | MPC |
| Junta 1 (rad) | 0,0144 | 0,0405 | 0,1201 | 0,2014 | 5,4836e-06 | 1,4543e-05 |
| Eixo x (m) | 0,0052 | 0,0141 | 0,0725 | 0,1190 | 3,4841e-04 | 2,8129e-04 |
| Eixo y (m) | 0,0023 | 0,0063 | 0,0484 | 0,0936 | 2,1733e-04 | 3,5138e-04 |
| Eixo z (m) | 0,0033 | 0,0087 | 0,0575 | 0,0799 | 4,0556e-04 | 3,7295e-04 |

Figura 7.7: Resposta ao degrau para atrasos de 0,7 ms, 100 ms, 200 ms, 400 ms, e 500 ms no canal de comunicação.



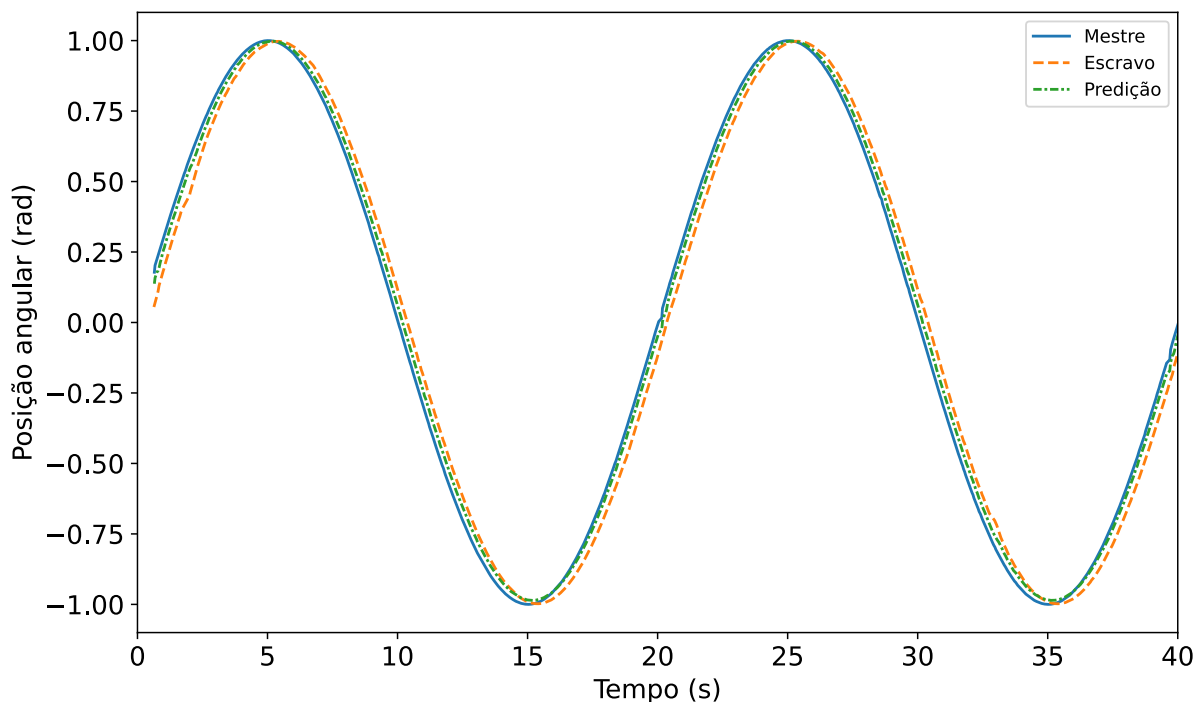
Fonte: Elaborada pelo autor.

7.3 Desempenho da Abordagem de Predição

A abordagem de predição com a utilização de rede neural ARX abordada no Capítulo 6 (ilustrada na Figura 6.13) é avaliada. Primeiramente, a adição da rede neural no sistema aumentou o atraso na execução do laço de controle. Portanto, o tempo de espera no laço de controle foi definido como $\Delta = 40 \text{ ms}$. Além disso, foi adicionado um atraso de 150 ms nos canais de comunicação do sistema. O horizonte de predição foi definido como 10 amostras

no futuro, logo espera-se a obtenção de uma predição de 200 ms. Aplicando um sinal de entrada do tipo senoidal com frequência de 50 mHz, foram adquiridos a resposta no lado escravo e também a predição realizada pela rede neural ARX dado o sinal de entrada. Na Figura 7.8 ilustra-se a posição de referência no lado mestre, a resposta atrasada no lado escravo e a predição realizada para a junta 1. O atraso total do sistema é de 350 ms e a abordagem alcançou a predição desejada de 200 ms.

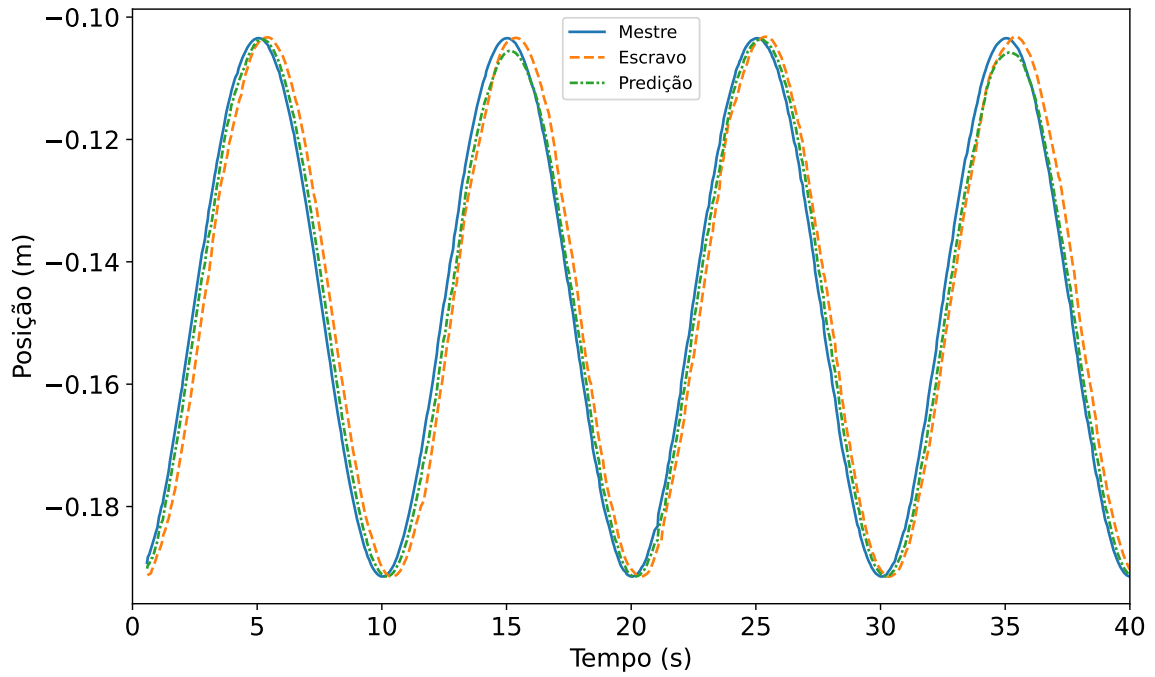
Figura 7.8: Resposta do sistema com preditor baseado em rede neural ARX para a junta 1.



Fonte: Elaborada pelo autor.

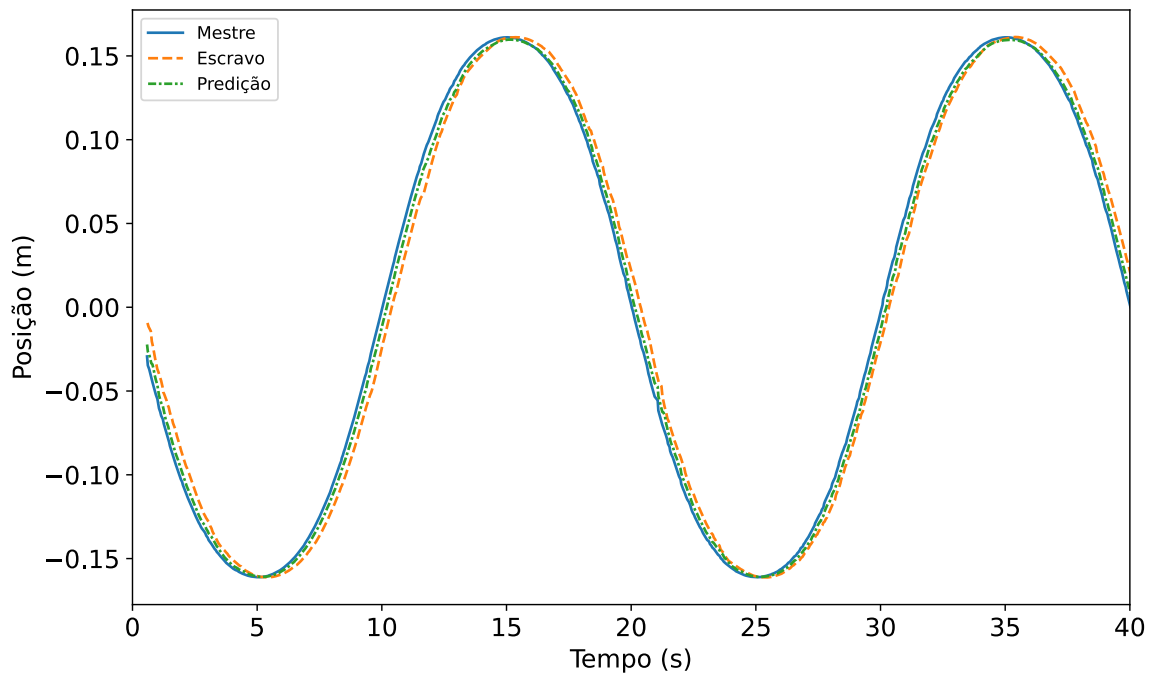
O rastreamento das posições, bem como a predição da posição escrava, nos eixos x e y são ilustradas nas Figuras 7.9 e 7.10, respectivamente. Não teve variação da resposta no eixo z dado que a variação da posição na junta 1 do manipulador, considerando sua posição inicial, não altera a sua posição no eixo z. Com o objetivo de avaliar o desempenho no rastreamento e predição da posição para a abordagem de predição utilizada, foram calculados os erros MAE e RMSE e também o erro relativo percentual ε . Os erros apresentados na Tabela 7.3 são muito pequenos considerando o atraso total do sistema, comprovando a eficácia do modelo de predição utilizado. O erro relativo percentual calculado para a junta 1 é de 4,9370 %, esse erro seria consideravelmente maior considerando a posição obtida do lado escravo sem a predição.

Figura 7.9: Resposta do sistema com preditor baseado em rede neural ARX para o eixo x.



Fonte: Elaborada pelo autor.

Figura 7.10: Resposta do sistema com preditor baseado em rede neural ARX para o eixo y.



Fonte: Elaborada pelo autor.

Tabela 7.2: Valores MAE, RMSE e ε obtidos para o sistema com preditor baseado em rede neural ARX.

| Posição | MAE | RMSE | ε (%) |
|----------------------|------------|---------|-------------------|
| Junta 1 (rad) | 0,0012 | 0,03498 | 4,9370 |
| Eixo x (m) | 1,0073e-05 | 0,00317 | 2,1120 |
| Eixo y (m) | 3,4766e-05 | 0,00589 | 4,9346 |

É importante ressaltar que o objetivo de latência de 1ms não é alcançado pelo sistema devido as suas limitações físicas e de poder de processamento computacional. O tempo de espera Δ no laço de controle utilizando a estratégia de predição com rede neural ARX e controle PD é de 40 ms, o que é muito superior a 1 ms. Esse tempo deveria ser, portanto, menor que 1 ms dado que a latência fim-a-fim requerida em um sistema de Internet Tátil é de até 1 ms. Além disso, a aquisição da resposta do manipulador utilizando o simulador virtual possui um atraso em torno de 50 ms que influencia na resposta fim-a-fim do sistema. Embora a latência de 1 ms não tenha sido alcançada nesse trabalho, as técnicas de controle e predição investigadas são promissoras no contexto de Internet Tátil, uma vez que, com o aumento do poder computacional e diminuição da latência no meio físico, as abordagens podem ser ajustadas para operar no sistema com frequência superior.

7.4 Considerações Finais

Neste capítulo foram apresentadas as simulações das abordagens e os resultados obtidos com o sistema desenvolvido. A configuração experimental estabelecida para realizar as simulações foi apresentada e detalhada. Os controladores avaliados apresentaram erros mínimos no rastreamento da posição angular das juntas provando a eficácia do sistema desenvolvido. Foram apresentadas as respostas ao degrau para diferentes atrasos nos canais de comunicação que comprovam o impacto do atraso no sistema e, conseqüentemente, a capacidade do sistema de operar sob diferentes atrasos no domínio da rede podendo ser utilizado para avaliação de abordagens de inteligência artificial preditiva e controle no contexto de Internet Tátil. A abordagem de predição utilizada também obteve um bom desempenho dado que realizou a predição desejada (a rede foi treinada para realizar predições em um horizonte de 200 ms) da posição do lado escravo.

No Capítulo 8, as conclusões acerca deste trabalho de pesquisa são apresentadas e trabalhos futuros são sugeridos.

Capítulo 8

Conclusões

A Internet Tátil está emergindo com a promessa de prover interações físicas de forma remota por meio da Internet. Mediante a transmissão de dados audiovisuais e dados hápticos como por exemplo o toque, pressão e vibração, a Internet Tátil deve possibilitar o controle remoto de objetos reais ou virtuais tornando a Internet mais abrangente e imersiva. Para que isso seja possível, uma conectividade ultra-responsiva e ultra-confiável é requerida, bem como devem ser garantidas a segurança e privacidade dos dados. Esses requisitos restritivos da Internet Tátil são comuns há vários outros segmentos da sociedade e portanto provém uma gama de cenários de aplicações na indústria, educação, saúde, transporte, realidade aumentada, realizada virtual, teleoperação, robótica e muitos outros setores.

Os requisitos da Internet Tátil impõem grandes desafios para a sua concretização. Os desafios tecnológicos são inúmeros, tanto do ponto de vista de dispositivos hápticos e cinemáticos relacionados diretamente a capacidade de transmissão táteis, como também desafios relacionados a diminuição da latência de rede, compressão de dados entre outros. Com uma conectividade ultra-responsiva, a quinta geração de Internet móvel (5G) surge como uma tecnologia de suporte para o desenvolvimento da Internet Tátil. Embora a latência no sistema possa ser diminuída ela ainda é restringida pelo meio físico. Portanto, o uso da computação na borda na rede pode ser expandido com um sistema de inteligência artificial baseada em predição para antecipar a atuação tátil. Desse modo, as abordagens baseadas em inteligência artificial preditiva atuais devem ser investigadas e melhoradas, e novas abordagens devem ser desenvolvidas.

Para realizar a investigação de abordagens de controle e inteligência artificial preditiva no cenário de Internet Tátil um sistema foi desenvolvido neste trabalho. O sistema desenvolvido é composto pelos três domínios característicos da Internet Tátil: domínio mestre; domínio de rede; domínio controlado. Dado as condições de custo e recursos restritos o dispositivo escravo foi simulado na plataforma de simulação de robôs CoppeliaSim,

bem como o dispositivo mestre foi substituído por sinais de referência e planejamento de trajetória.

O manipulador UR5 foi definido como o robô escravo a ser utilizado no domínio controlado. Usando a representação de Denavit-Hartenberg um modelo cinemático pôde ser construído para o manipulador. A cinemática inversa, que corresponde ao mapeamento do espaço cartesiano para o espaço das juntas foi realizado utilizando para isso uma solução analítica fazendo uso das propriedades geométricas do manipulador. Por ser um método mais rápido e fornecer todas as respostas possível ele foi adotado nessa pesquisa. Entretanto, necessita de um estudo maior, principalmente em relação a configurações próximas de singularidades. A dinâmica do manipulador, por sua vez, é implementada pelo simulador virtual.

O sistema desenvolvido possui duas APIs que são executadas nas bordas da rede: API mestre e API escrava. O domínio de rede corresponde a canais de comunicação com protocolo de comunicação UDP que realiza a troca de informações entre as APIs. A comunicação entre a API escrava e a cena (com o manipulador) no simulador virtual foi provida por meio de um cliente *Remote API* que faz parte do CoppeliaSim e permite a comunicação com aplicações externas ao software via TCP/IP. As APIs mestre e escrava foram sincronizadas para o correto funcionamento do sistema e a cena no simulador virtual foi executada de forma assíncrona as aplicações.

Foram propostas duas abordagens de controle para o sistema: controle PD e controle MPC. A análise do desempenho dos controladores foi realizado na tarefa do rastreo do sinal de referência. Ambos os controladores apresentaram bons resultados, embora o controlador PD tenha obtido um desempenho melhor. O desempenho superior do controlador PD pode ser explicado pelo uso do algoritmo genético utilizado para a sintonia dos ganhos do controlador. Além disso, o controlador MPC apresentou um tempo de execução superior ao controlador PD, o que é bastante relevante em um sistema onde deseja-se diminuir a latência. O desempenho do sistema pode ser comprovado diante dos resultados satisfatórios obtidos com as abordagens de controle.

Inicialmente foi investigada uma abordagem baseada em modelo a partir de um preditor de Smith. Para reprodução do modelo do manipulador no lado mestre foi considerada uma rede neural artificial baseada em modelo autorregressivo com entradas exógenas, *Autoregressive Exogenous* (ARX). Para a utilização dessa abordagem, uma rede neural ARX com arquitetura do tipo paralela foi investigada. Porém ela apresentou erros de predição muito elevados em razão da predição ser realizada de forma offline. Com isso, uma nova abordagem utilizando uma rede neural ARX com arquitetura série-paralela foi investigada para obter um melhor desempenho de predição, já que nessa arquitetura a predição é realizada de forma online. Essa nova abordagem foi capaz de realizar previsões

da posição do lado escravo em um horizonte de 200 ms.

As abordagens de controle e inteligência artificial preditiva foram investigadas utilizando o sistema desenvolvido com o intuito de validá-lo. Com o sistema é possível, além de simular as abordagens em dispositivos de computação nas bordas, realizar o controle do tráfego no domínio da rede (por exemplo, adicionando atraso ou perda de pacotes). O sistema, entretanto, possui algumas limitações de processamento que impactaram nas simulações das abordagens investigadas adicionando atraso. O atraso adicionado pelo processamento aumentou a latência fim-a-fim além da latência de 1 ms requerida pela Internet Tátil. Contudo, o sistema desenvolvido representa uma base sólida para pesquisas futuras no domínio da Internet Tátil, robótica, teleoperação, controle de sistemas com atraso no tempo, inteligência artificial entre outros. O sistema pode ser facilmente adaptado para admitir novas tecnologias e dispositivos que possam ser integrados a posteriori. Além do mais, novas abordagens de controle e inteligência artificial podem ser facilmente simuladas utilizando o sistema.

8.1 Sugestões de Trabalhos Futuros

A partir deste trabalho de pesquisa, algumas recomendações consideradas promissoras são sugeridas para trabalhos futuros:

- Um dispositivo háptico pode ser acoplado ao sistema no lado mestre para converter as entradas do operador em dados hápticos e fornecer também a resposta háptica. Com isso, é possível realizar o controle háptico bidirecional entre os domínios mestre e escravo. Além disso, o sistema pode ser modificado para fornecer resposta audiovisual ao usuário, sendo suficiente para isso a adição de laços de controle audiovisual no sistema;
- A sintonia dos parâmetros do controlador MPC pode ser realizada de forma a otimizar sua resposta. Além disso, o controlador MPC requer um uso intensivo de computação o que abre precedentes para a investigação desse tipo de controle em dispositivos locais ou na nuvem como abordado em (SKARIN et al., 2019);
- A abordagem de predição utilizada pode ser melhorada, inclusive com a predição da resposta do lado mestre no lado escravo para que tenha-se a diminuição da sensação de atraso em ambos os lados do sistema. Novas abordagens podem ser investigadas utilizando o sistema, dado que ele é capaz de simular os atrasos no domínio da rede mantendo a frequência de operação do laço de controle.

Referências Bibliográficas

AIJAZ, A.; SOORIYABANDARA, M. The tactile internet for industries: A review. *Proceedings of the IEEE*, IEEE, v. 107, n. 2, p. 414–435, 2018.

ALLIN, S.; MATSUOKA, Y.; KLATZKY, R. Measuring just noticeable differences for haptic force feedback: implications for rehabilitation. In: IEEE. *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*. [S.l.], 2002. p. 299–302.

ANDERSEN, R. S. Kinematics of a ur5. *Aalborg University*, 2018.

ANTONAKOGLU, K. et al. Toward haptic communications over the 5g tactile internet. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 4, p. 3034–3059, 2018.

ATEYA, A. et al. Model mediation to overcome light limitation toward a secure tactile internet system. *Journal of Sensor and Actuator Networks*, Multidisciplinary Digital Publishing Institute, v. 8, n. 1, p. 6, 2019.

BAKKE, M. *Subspace Identification using Closed-Loop Data*. Dissertação (Mestrado) — Institutt for teknisk kybernetikk, 2009.

BERG, D. V. D. et al. Challenges in haptic communications over the tactile internet. *IEEE Access*, IEEE, v. 5, p. 23502–23518, 2017.

BOUSSAADA, Z. et al. A nonlinear autoregressive exogenous (narx) neural network model for the prediction of the daily direct solar radiation. *Energies*, Multidisciplinary Digital Publishing Institute, v. 11, n. 3, p. 620, 2018.

BRANDI, F.; STEINBACH, E. G. Prediction techniques for haptic communication and their vulnerability to packet losses. In: CITESEER. *HAVE*. [S.l.], 2013. p. 63–68.

BRUNTON, S. L.; KUTZ, J. N. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. [S.l.]: Cambridge University Press, 2019.

CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature. *Geoscientific model development*, Copernicus GmbH, v. 7, n. 3, p. 1247–1250, 2014.

CORKE, P. *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*. [S.l.]: Springer, 2017. v. 118.

CRAIG, J. J. *Introduction to robotics: mechanics and control, 3/E*. [S.l.]: Pearson Education India, 2009.

- DIVERSI, R.; GUIDORZI, R.; SOVERINI, U. Identification of arx and ararx models in the presence of input and output noises. *European Journal of Control*, Elsevier, v. 16, n. 3, p. 242–255, 2010.
- DOHLER, M. et al. Internet of skills, where robotics meets ai, 5g and the tactile internet. In: IEEE. *2017 European Conference on Networks and Communications (EuCNC)*. [S.l.], 2017. p. 1–5.
- DOLUI, K.; DATTA, S. K. Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In: IEEE. *2017 Global Internet of Things Summit (GIoTS)*. [S.l.], 2017. p. 1–6.
- DURIEZ, T.; BRUNTON, S. L.; NOACK, B. R. *Machine learning control-taming nonlinear dynamics and turbulence*. [S.l.]: Springer, 2017. v. 116.
- FETTWEIS, G. P. The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine*, IEEE, v. 9, n. 1, p. 64–70, 2014.
- GAING, Z.-L. A particle swarm optimization approach for optimum design of pid controller in avr system. *IEEE transactions on energy conversion*, IEEE, v. 19, n. 2, p. 384–391, 2004.
- GIER, M. D. Control of a robotic arm: Application to on-surface 3d-printing. 2015.
- GRAVDAHL, J. T. *Force estimation in robotic manipulators: Modeling, simulation and experiments*. Tese (Doutorado) — Citeseer, 2014.
- GUO, F. et al. Haptic data compression based on quadratic curve reconstruction and prediction. In: ACM. *Proceedings of the Third International Conference on Internet Multimedia Computing and Service*. [S.l.], 2011. p. 193–196.
- GURNEY, K. *An introduction to neural networks*. [S.l.]: CRC press, 1997.
- HADDADIN, S.; JOHANNSMEIERS, L.; LEDEZMA, F. D. Tactile robots as a central embodiment of the tactile internet. *Proceedings of the IEEE*, IEEE, v. 107, n. 2, p. 471–487, 2019.
- HAWKINS, K. P. *Analytic inverse kinematics for the universal robots UR-5/UR-10 arms*. [S.l.], 2013.
- HAYKIN, S. *Neural Networks and Learning Machines, 3/E*. [S.l.]: Pearson Education India, 2010.
- HINTERSEER, P. et al. Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Transactions on Signal Processing*, IEEE, v. 56, n. 2, p. 588–597, 2008.
- HØIFØDT, H. *Dynamic modeling and simulation of robot manipulators: The newton-euler formulation*. Dissertação (Mestrado) — Institutt for teknisk kybernetikk, 2011.
- HUANG, J.-Q.; LEWIS, F. L. Neural-network predictive control for nonlinear dynamic systems with time-delay. *IEEE Transactions on Neural Networks*, IEEE, v. 14, n. 2, p. 377–389, 2003.

KALASARIYA, M.; PATEL, V.; THAKKAR, A. Comparative study of iterative inverse kinematics methods for serial manipulators.

KATHUSHIKO, O. Engenharia de controle moderno. 5^o edição. *Editora LTC*, 2011.

KEBRIA, P. M. et al. Kinematic and dynamic modelling of ur5 manipulator. In: *IEEE. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2016. p. 004229–004234.

KUFIETA, K. Force estimation in robotic manipulators: Modeling, simulation and experiments. *Department of Engineering Cybernetics NTNU Norwegian University of Science and Technology*, 2014.

LIU, C.; GUO, J.; POIGNET, P. Nonlinear model-mediated teleoperation for surgical applications under time variant communication delay. *IFAC-PapersOnLine*, Elsevier, v. 51, n. 22, p. 493–499, 2018.

MAIER, M.; EBRAHIMZADEH, A.; CHOWDHURY, M. The tactile internet: automation or augmentation of the human? *IEEE Access*, IEEE, v. 6, p. 41607–41618, 2018.

MIRANDA, M. N. de. *Algoritmos genéticos: fundamentos e aplicações*. 2007.

PANE, Y. P. Reinforcement learning for tracking control in robotics. Citeseer, 2015.

POLACHAN, K. et al. Quality of control assessment for tactile internet based cyber-physical systems. *arXiv preprint arXiv:1910.08743*, 2019.

RAGAZZON, M. R. P. Robot manipulator collision handling in unknown environment without using external sensors. *Department of Engineering Cybernetics Norwegian University of Science and Technology*, 2012.

ROBOTS, U. User manual, ur5/cb3. Disponível em: https://www.universalrobots.com/media/8704/ur5_user_manual_gb.pdf, 2015.

ROBOTS, U. *PARAMETERS FOR CALCULATIONS OF KINEMATICS AND DYNAMICS*. 2020. Disponível em: <https://www.universal-robots.com/articles/ur-articles/parameters-for-calculations-of-kinematics-and-dynamics/>.

ROHMER, E.; SINGH, S. P. N.; FREESE, M. Coppeliassim (formerly v-rep): a versatile and scalable robot simulation framework. In: *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2013.

SACHS, J. et al. Adaptive 5g low-latency communication for tactile internet services. *Proceedings of the IEEE*, IEEE, n. 99, p. 1–25, 2018.

SAKR, N. et al. Motion and force prediction in haptic media. In: *IEEE. 2007 IEEE International Conference on Multimedia and Expo*. [S.l.], 2007. p. 2242–2245.

SCHRIMPF, J.; LIND, M.; MATHISEN, G. Real-time analysis of a multi-robot sewing cell. In: *IEEE. 2013 IEEE International Conference on Industrial Technology (ICIT)*. [S.l.], 2013. p. 163–168.

SHEN, S. et al. Time delay compensation for nonlinear bilateral teleoperation: A motion prediction approach. *Transactions of the Institute of Measurement and Control*, SAGE Publications Sage UK: London, England, v. 41, n. 16, p. 4488–4498, 2019.

SHI, W. et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, IEEE, v. 3, n. 5, p. 637–646, 2016.

SIMSEK, M. et al. 5g-enabled tactile internet. *IEEE Journal on Selected Areas in Communications*, IEEE, v. 34, n. 3, p. 460–473, 2016.

SIMSEK, M.; FETTWEIS, G. P.; I, C.-L. *Tactile Internet*. [S.l.]: IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC 445 HOES LANE, PISCATAWAY, NJ , 2019.

SKARIN, P. et al. An assisting model predictive controller approach to control over the cloud. *arXiv preprint arXiv:1905.06305*, 2019.

SMITH, A. C.; HASHTRUDI-ZAAD, K. V. Neural network-based teleoperation using smith predictors. In: IEEE. *IEEE International Conference Mechatronics and Automation, 2005*. [S.l.], 2005. v. 3, p. 1654–1659.

SONG, J. et al. Model-mediated teleoperation with improved stability. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 15, n. 2, p. 1729881418761136, 2018.

SPONG, M. W. et al. *Robot modeling and control*. [S.l.: s.n.], 2006.

STEINBACH, E. et al. Haptic data compression and communication. *IEEE Signal Processing Magazine*, IEEE, v. 28, n. 1, p. 87–96, 2010.

STEINBACH, E. et al. Haptic codecs for the tactile internet [40pt]. *Proceedings of the IEEE*, IEEE, n. 99, p. 1–24, 2018.

TEE, T. H. et al. A network-adaptive compensation technique for tele-haptics using position prediction algorithm. In: IEEE. *2013 IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*. [S.l.], 2013. p. 39–44.

TUTUNJI, T. A. Parametric system identification using neural networks. *Applied Soft Computing*, Elsevier, v. 47, p. 251–261, 2016.

UZUNOĞLU, E.; DEDE, M. İ. C. Extending model-mediation method to multi-degree-of-freedom teleoperation systems experiencing time delays in communication. Cambridge University Press, 2017.

VERHAEGEN, M. Identification of the deterministic part of mimo state space models given in innovations form from input-output data. *Automatica*, Elsevier, v. 30, n. 1, p. 61–74, 1994.

VERHAEGEN, M.; DEWILDE, P. Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International journal of control*, Taylor & Francis, v. 56, n. 5, p. 1211–1241, 1992.

WILLAERT, B. et al. Towards multi-dof model mediated teleoperation: Using vision to augment feedback. In: IEEE. *2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings*. [S.l.], 2012. p. 25–31.

XU, X. et al. Model-mediated teleoperation: toward stable and transparent teleoperation systems. *IEEE Access*, IEEE, v. 4, p. 425–449, 2016.

YAZDANKHOO, B. et al. Improvement of operator position prediction in teleoperation systems with time delay: Simulation and experimental studies on phantom omni devices. *JJMIE*, v. 13, n. 3, 2019.

ZHENG, Y. et al. A delay compensation framework for predicting heading in teleoperated ground vehicles. *IEEE/ASME Transactions on Mechatronics*, IEEE, v. 24, n. 5, p. 2365–2376, 2019.