

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DAVI ALBERTO SALA

**CONTROLE DE POSIÇÃO COM
MÚLTIPLOS SENSORES EM UM
ROBÔ COLABORATIVO
UTILIZANDO LIQUID STATE
MACHINES**

Porto Alegre
2017

DAVI ALBERTO SALA

**CONTROLE DE POSIÇÃO COM
MÚLTIPLOS SENSORES EM UM
ROBÔ COLABORATIVO
UTILIZANDO LIQUID STATE
MACHINES**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.
Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Valner João Brusamarello

Porto Alegre
2017

DAVI ALBERTO SALA

**CONTROLE DE POSIÇÃO COM
MÚLTIPLOS SENSORES EM UM
ROBÔ COLABORATIVO
UTILIZANDO LIQUID STATE
MACHINES**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Valner João Brusamarello, UFRGS
Doutor pela Universidade Federal de Santa Catarina - Florianópolis, Brasil

Banca Examinadora:

Profa. Dra. Adriane Parraga, UERGS
Doutora pela Universidade Federal do Rio Grande do Sul- Porto Alegre, Brasil

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela Universidade Federal de Minas Gerais - Belo Horizonte, Brasil

Prof. Dr. Herbert Martins Gomes, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul- Porto Alegre, Brasil

Coordenador do PPGE: _____
Prof. Dr. Valner João Brusamarello

Porto Alegre, Março de 2017.

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa.

Aos meus pais por todo apoio e suporte aos meus estudos.

Ao meu orientador Prof. Dr. Valner Brusamarello por todos ensinamentos, toda paciência e principalmente o apoio neste trajeto que cada vez mais se mostra rumo à uma vida de pesquisador.

Ao colega pesquisador e amigo Ricardo de Azambuja, por todo apoio e pela grande oportunidade de trabalho conjunto, onde tive o prazer de aprender muito nas áreas de robótica e redes neurais.

Ao professor Dr. Angelo Cangelosi pela oportunidade de aprendizado na Plymouth University, onde fui convidado para um estágio de dois meses para execução experimental deste trabalho.

Aos colegas do PPGEE pelo seu auxílio nas tarefas desenvolvidas durante o curso, e apoio na revisão deste trabalho.

Ao CNPq pela provisão da bolsa de mestrado.

RESUMO

A ideia de usar redes neurais biologicamente inspiradas na computação tem sido amplamente utilizada nas últimas décadas. O fato essencial neste paradigma é que um neurônio pode integrar e processar informações, e esta informação pode ser revelada por sua atividade de pulsos. Ao descrever a dinâmica de um único neurônio usando um modelo matemático, uma rede pode ser implementada utilizando um conjunto desses neurônios, onde a atividade pulsante de cada neurônio irá conter contribuições, ou informações, da atividade pulsante da rede em que está inserido. Neste trabalho é apresentado um controlador de posição no eixo Z utilizando fusão de sensores baseado no paradigma de Redes Neurais Recorrentes. O sistema proposto utiliza uma Máquina de Estado Líquido (LSM) para controlar o robô colaborativo BAXTER. O *framework* foi projetado para trabalhar em paralelo com as LSMs que executam trajetórias em formas fechadas de duas dimensões, com o objetivo de manter uma caneta de feltro em contato com a superfície de desenho, dados de sensores de força e distância são alimentados ao controlador. O sistema foi treinado utilizando dados de um controlador Proporcional Integral Derivativo (PID), fundindo dados de ambos sensores. Resultados mostram que a LSM foi capaz de aprender o comportamento do controlador PID em diferentes situações.

Palavras-chave: Fusão de sensores, Spiking Neural Networks, Liquid State Machine, Controle de posição.

ABSTRACT

The idea of employing biologically inspired neural networks to perform computation has been widely used over the last decades. The essential fact in this paradigm is that a neuron can integrate and process information, and this information can be revealed by its spiking activity. By describing the dynamics of a single neuron using a mathematical model, a network in which the spiking activity of every single neuron will get contributions, or information, from the spiking activity of the embedded network. A positioning controller based on Spiking Neural Networks for sensor fusion suitable to run on a neuromorphic computer is presented in this work. The proposed framework uses the paradigm of reservoir computing to control the collaborative robot BAXTER. The system was designed to work in parallel with Liquid State Machines that performs trajectories in 2D closed shapes. In order to keep a felt pen touching a drawing surface, data from sensors of force and distance are fed to the controller. The system was trained using data from a Proportional Integral Derivative controller, merging the data from both sensors. The results show that the LSM can learn the behavior of a PID controller on different situations.

Keywords: Sensor Fusion, Spiking Neural Networks, Liquid State Machines, Positioning Control, BAXTER robot, PID controller.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	10
LISTA DE ABREVIATURAS	11
LISTA DE SÍMBOLOS	12
1 INTRODUÇÃO	14
1.1 Objetivo Geral	17
1.1.1 Objetivos Específicos:	17
1.2 Contribuição	17
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 Redes Neurais Artificiais	18
2.1.1 Redes Neurais Feedforward	18
2.1.2 Redes Neurais Recorrentes	19
2.2 Elementos de Sistemas Neurais	19
2.2.1 Trem de Pulsos	20
2.2.2 Sinapses	21
2.2.3 Potenciais Pós-sinápticos	21
2.2.4 Limiar de disparo e Potencial de Ação	21
2.3 Reservoir Computing	23
2.3.1 Liquid State Machines	24
2.3.2 Modelo neural Integra-e-Dispara	24
2.3.3 Machine Learning	25
2.4 Controle de robô com Spiking Neural Network	26
3 MATERIAIS E MÉTODOS	30
3.1 Robô BAXTER	31
3.1.1 Sensor de distância	32
3.1.2 Força estimada	33
3.2 Trajetórias	33
3.3 Conjunto de treinamento	34
3.4 Projeto e Simulação dos Líquidos	37
3.5 Criação do controlador	40

4	RESULTADOS	43
4.0.1	Avaliação inicial	45
4.0.2	Dados experimentais	47
4.0.3	Nova trajetória	58
5	CONCLUSÃO	61
5.1	Trabalhos Futuros	62
	REFERÊNCIAS	63

LISTA DE ILUSTRAÇÕES

Figura 1:	Exemplo do Robô BAXTER desenhando em um quadro branco com e sem controle no eixo Z	16
Figura 2:	Estrutura genérica de redes dos tipos FNNs e RNNs.	19
Figura 3:	Estrutura básica de um neurônio e ocorrência de uma sinapse.	20
Figura 4:	Exemplo simplificado para ocorrência de sinapses e disparos.	22
Figura 5:	Estrutura de uma RNN do framework Reservoir Computing.	23
Figura 6:	Diagrama representando um modelo de Máquina de Estado Líquido.	24
Figura 7:	Diagrama esquemático de um neurônio Integra-e-Dispara.	25
Figura 8:	Trajetórias 2D utilizadas na rede de controle.	27
Figura 9:	Ângulos das juntas utilizados para realizar a trajetória quadrada com o robô BAXTER.	28
Figura 10:	Disposição de uma LSM para controle de trajetória.	28
Figura 11:	Disposição de LSMs paralelas.	29
Figura 12:	Disposição do robô BAXTER para executar a tarefa.	31
Figura 13:	Localização do sensor de distância no <i>gripper</i> do robô BAXTER.	32
Figura 14:	Cinco trajetórias das juntas S1 e W1.	33
Figura 15:	Distância no eixo Z-axis.	34
Figura 16:	Diagrama do sistema utilizado para criar o conjunto de treinamento.	35
Figura 17:	Disposições da superfície de desenho utilizadas.	35
Figura 18:	Valores médios de força do conjunto de treinamento.	36
Figura 19:	Valores médios de distância do conjunto de treinamento.	37
Figura 20:	Valores médios de saída de controle dos PIDs do conjunto de treinamento.	37
Figura 21:	Normalização da entrada para uma entrada gaussiana. Fonte: Adaptado de (AZAMBUJA; CANGELOSI; ADAMS, 2016)	39
Figura 22:	Normalização do sinal de força para um índice neural.	40
Figura 23:	Normalização do sinal de distância para um índice neural.	41
Figura 24:	Atividade neural no líquido durante simulação utilizando dados do conjunto de treinamento <i>PID - Set 1</i> como entrada.	41
Figura 25:	Atividade neural no líquido durante simulação utilizando dados do conjunto de treinamento <i>PID - Set 2</i> como entrada.	42
Figura 26:	Diagrama do sistema para realizar os experimentos.	42
Figura 27:	Posição no eixo Z utilizando a LSM de controle de trajetória.	44
Figura 28:	Teste inicial das LSMs, utilizando dados do conjunto de treinamento.	45
Figura 29:	Força resultante das LSMs no conjunto de experimentos A.	46
Figura 30:	Resultados obtidos no Experimento A.	48

Figura 31:	Resultados obtidos no Experimento B.	49
Figura 32:	Resultados obtidos no Experimento C.	50
Figura 33:	Resultados obtidos no Experimento D.	51
Figura 34:	Resultados combinados das quatro etapas experimentais (A, B, C e D).	52
Figura 35:	Comparação entre os valores médios de força dos controladores LSM 3 e PID.	54
Figura 36:	Comparação entre os valores médios de distância dos controladores LSM 3 e PID.	55
Figura 37:	Distribuição de probabilidade das leituras do sensor de força.	56
Figura 38:	Distribuição de probabilidade das leituras do sensor de distância.	57
Figura 39:	Exemplo do Robô BAXTER desenhando em um quadro branco com e sem controle no eixo Z	58
Figura 40:	Comparação entre as Trajetórias 1 e 2.	59
Figura 41:	Média dos valores de força, utilizando a LSM 3 na Trajetória 2.	59
Figura 42:	Média dos valores de distância, utilizando a LSM 3 na Trajetória 2.	60

LISTA DE TABELAS

Tabela 1:	Definição dos experimentos.	44
Tabela 2:	Erro quadrático médio das medidas de força.	52
Tabela 3:	Resultados do sensor de força do controlador LSM 3.	53
Tabela 4:	Resultados do sensor de distância do controlador LSM 3.	56

LISTA DE ABREVIATURAS

2D	Bidimensional
ANN	<i>Artificial Neural Network</i> (Rede Neural Artificial)
EPSP	<i>Excitatory Post-Sinaptic Potential</i> (Potencial Pós-Sináptico Excitatório)
FNN	<i>Feed-forward Neural Network</i>
IC	Intervalo de Confiança
IK	<i>Inverse Kinematics</i> (Cinemáticas Inversas)
IPSP	<i>Inhibitory Post-Sinaptic Potential</i> (Potencial Pós-Sináptico Inibitório)
LIF	<i>Leaky Integrate and Fire</i> (Integra-e-Dispara com Vazamento)
LSM	<i>Liquid State Machine</i> (Máquina de Estado Líquido)
MSE	<i>Mean Square Error</i> (Erro Médio Quadrático)
OLS	<i>Ordinary Least Squares</i> (Mínimos Quadrados Ordinários)
PID	Proporcional Integral Derivativo
PSP	<i>Post-Sinaptic Potential</i> (Potencial Pós-Sináptico)
SNN	<i>Spiking Neural Network</i> (Rede Neural Pulsada)
RC	<i>Resistor-Capacitor</i>
RMSE	<i>Root-Mean-Square Error</i> (Raiz do Erro Quadrático Médio)
ROS	<i>Robot Operating System</i>
STP	<i>Short-Term Plasticity</i> (Plasticidade de Curto Prazo)
SWN	<i>Small World Network</i> (Rede de Pequeno Mundo)
UK	<i>United Kingdom</i> (Reino Unido)

LISTA DE SÍMBOLOS

X	Eixo das abscissas
Y	Eixo das ordenadas
Z	Eixo das cotas
$v(t)$	Potencial da membrana
v_{rest}	Potencial de descanso da membrana
j	neurônio pré-sináptico
i	neurônio pós-sináptico
$\epsilon_{ij}(t)$	potencial pós-sináptico
$v_i(t)$	Potencial da membrana pós-sináptica
t	Instante de tempo
$t_j^{(f)}$	Instante de tempo de disparo
ϑ	Limiar da membrana
$\delta(t - t_i^{(f)})$	Pulso pré-sináptico
$\delta(t - t_j^{(f)})$	Pulso pós-sináptico
$\alpha(t - t_j^{(f)})$	Pulso de corrente
$i_e(t)$	corrente excitatória do neurônio
$i_i(t)$	corrente inibitória do neurônio
i_{offset}	correntes de ruído fixa
i_{noise}	correntes de ruído variável
C_m	Capacitância da membrana
τ_m	Constante de tempo da membrana
τ_{syn_i}	tempo de decaimento da sinapse inibitória
τ_{syn_e}	tempo de decaimento da sinapse excitatória
w	vetor de pesos da função de leitura
K_F	Saída de controle do controlador PID de força
K_D	Saída de controle PID de distância

K_T	Saída de controle combinada
res_{Input}	Resolução da entrada
$Input_{max}$	Valor máximo da entrada
$Input_{min}$	Valor mínimo da entrada
NdN	Número de Neurônios de entrada
μ_{PID_f}	Média de força obtida com o controlador PID
σ_{PID_f}	Desvio padrão de força obtida com o controlador PID
μ_{LSM_f}	Média de força obtida com o controlador LSM
σ_{LSM_f}	Desvio padrão de força obtida com o controlador LSM
μ_{PID_d}	Média de distância obtida com o controlador PID
σ_{PID_d}	Desvio padrão de distância obtida com o controlador PID
μ_{LSM_d}	Média de distância obtida com o controlador LSM
σ_{LSM_d}	Desvio padrão de distância obtida com o controlador LSM

1 INTRODUÇÃO

Sabe-se por experiência que o cérebro humano é capaz de manter um modelo do mundo ao seu redor, atualizado constantemente através de informações sensoriais. Este modelo permite que nossos sistemas motores possam interagir com o mundo real. Porém, apesar de toda evolução na neurociência, ainda não se tem ideia definitiva de como o cérebro realmente cria e mantém este modelo (DRUBACH, 2000).

Um fluxo de pulsos são enviadas pelos órgãos sensoriais ao nosso sistema neurológico, cada uma dessas informações atualizam o modelo atual que representa todo o contexto em que o indivíduo está inserido. Essa informação é integrada aos estados internos do sistema que mantemos, ou seja, o estado atual é o resultado cumulativo de pulsos passados.

Fusão de dados com múltiplos sensores (STIEBER; PETRIU; VUKOVICH, 1998) é um assunto amplamente disseminado e possui aplicações em diversas áreas, como vigilância, controle de robôs, pilotagem automática de veículos, entre outros. Em aplicações típicas, dados de sensores redundantes e complementares são integrados, produzindo sistemas mais robustos, aumentando sua confiabilidade. Geralmente, fusão de sensores envolve associação de dados, estimativa de estado, e ou decisão de fusão, através da aplicação de ferramentas matemáticas como filtro de Kalman (FONTANELLI et al., 2016), redes Bayesianas (MORAIS BORGES; BRUSAMARELLO, 2014), lógica *fuzzy* (VILLANUEVA et al., 2012) e redes neurais, entre outras (BROOKS; IYENGAR, 1998).

Muitas aplicações empregam redes neurais. Em (KANDILLI et al., 2007), dados de múltiplos sensores são usados como entrada para o sistema, treinado para monitorar o desgaste da ferramentas de perfuração e fresagem. O sistema utilizado é capaz de monitoramento preciso do desgaste da ferramenta. Os resultados experimentais, apresentados no trabalho, em diferentes condições, demonstraram que o monitoramento da condição ferramental pode ser implementada usando redes neurais.

Em (PALUMBO et al., 2013), os autores apresentam um sistema de fusão de sensores utilizando o paradigma de *Reservoir Computing*, onde dados amostrados por um acelerômetro e valores de indicação de força do sinal são recebidos de sensores sem fio do usuário. A rede é utilizada para determinar e classificar atividades dentro conjunto de atividades diárias comuns, com o objetivo de prestar assistência e cuidados aos idosos.

Os autores de (GARCIA et al., 2012) apresentam modelos de fusão de sensores utilizando Redes Neurais Artificiais (ANN - *Artificial Neural Network*). O primeiro trabalho combina as informações fornecidas por dois sensores, um radar e uma câmera, para auxiliar na direção de veículos autônomos, melhorando a detecção

frontal de veículos e aumentando a taxa de detecção de veículos em ultrapassagens. Em (O’CONNOR et al., 2013) uma *Spiking Deep Belief Network* é implementada para classificação em tempo real, unindo informações visuais e auditivas. A aplicação demonstrada é para reconhecimento de letras escritas, mas o sistema pode ser expandido para outras tarefas.

Os trabalhos citados previamente apresentaram bons resultados em suas aplicações, demonstrando que o processamento de informações de vários sensores usando redes neurais pode melhorar consideravelmente a identificação e monitoramento de processos.

O cérebro humano é normalmente utilizado como inspiração quando se trabalha com ANNs, principalmente por suas características de robustez e eficiência energética. Enquanto um computador moderno em uma configuração comum precisa de uma fonte de alimentação de cerca de 500W, nosso cérebro gasta apenas 20W (DRUBACH, 2000). Na engenharia neuromórfica (MONROE, 2014), dispositivos são inspirados pelo cérebro humano. Atualmente, vários projetos de computadores neuromórficos estão em desenvolvimento, como SpiNNaker (FURBER et al., 2013), Darwin (SHEN et al., 2016), IBM-TrueNorth (MEROLLA et al., 2014) e Spikey (PFEIL et al., 2013). Todos eles são baseados em Spiking Neural Networks (SNN), o modelo de neurônio artificial que melhor representa a forma como nossos neurônios trabalham.

Além dos conceitos habituais de rede neural, a SNN também acrescenta a ideia de tempo em seu modo de operação. O neurônio não apenas dispara em um ciclo de propagação, mas quando seu potencial atinge um certo limite. Quando um neurônio ativa ou dispara, ele gera um sinal que viaja para outros neurônios que, por sua vez, aumentam ou diminuem seu potencial de acordo com o sinal recebido (MAASS, 1997).

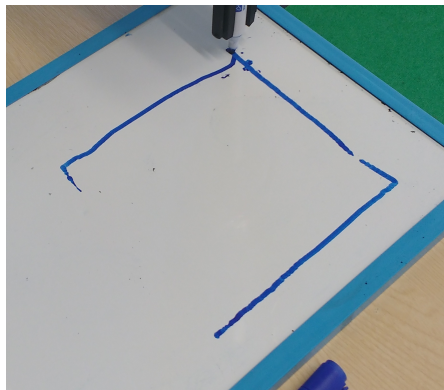
Uma Máquina de Estado Líquido (em inglês, *Liquid State Machine* - LSM) (MAASS; NATSCHLÄGER; MARKRAM, 2002) é constituída por camadas "líquidas" de neurônios recorrentemente e aleatoriamente conectados, aos quais são enviados padrões de pulsos temporais. Devido à sua aleatoriedade nas interconexões, a camada líquida pode ser vista como uma função aleatória da entrada. Este tipo de rede é comumente aplicado a problemas de classificação, onde o tempo e o espaço são informações relevantes. Em (VERSTRAETEN et al., 2005), por exemplo, uma LSM é usada para classificar a fala de dígitos isolados, onde a codificação biologicamente inspirada reproduziu resultados melhores que os métodos mais convencionais.

Com a intenção de utilizar o hardware SpiNNaker para o controle de robôs, pesquisadores da *Plymouth University* (AZAMBUJA; CANGELOSI; ADAMS, 2016) desenvolveram um modelo de redes paralelas, biologicamente inspiradas, implementado com LSMs. As redes são utilizadas para controlar BAXTER (FITZGERALD, 2013), um robô humanóide colaborativo. O sistema é capaz de aprender as trajetórias necessárias para desenhar uma forma bidimensional na superfície de uma mesa. No entanto, o sistema em questão não implementa controle de eixo-Z, uma vez que o feedback proprioceptivo vem apenas a partir dos ângulos de juntas, e quaisquer alterações na superfície, nivelamento do robô ou ruído externo introduzem erros nos desenhos.

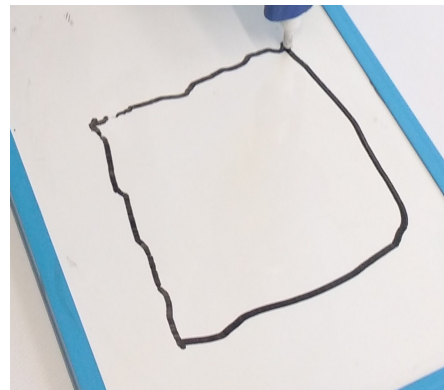
No presente trabalho, partir de uma parceria com o projeto desenvolvido em *Plymouth University*, propõe-se uma estrutura de controle, também baseada em LSM, para que possa ser, posteriormente, importada em um hardware neuromórfico, com

o objetivo de reduzir a variabilidade das trajetórias realizadas pelo robô BAXTER utilizando o controle com LSMs. As quais são executadas paralelamente à malha de controle do robô, a fim de reduzir o erro de posicionamento da caneta. O sistema proposto é capaz de aprender o comportamento de um controlador Proporcional Integral Derivativo (PID), mantendo uma distância constante da mesa e aplicando força estável e, portanto, mantendo o contato entre caneta e mesa.

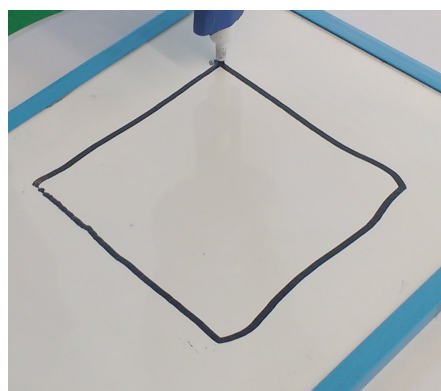
Para exemplificar, as figuras 1(a) e 1(b) apresentam resultados de dois quadros desenhados sem o sistema de controle de posicionamento proposto, com o robô BAXTER realizando a trajetória quadrada com uma caneta de feltro sobre um quadro branco. Na primeira imagem, a caneta de feltro não toca a superfície em alguns momentos da trajetória, resultando em falhas, ou faltas. Na segunda, muita força é aplicada, causando uma distorção no desenho devido ao atrito da caneta na superfície. A figura 1(c) mostra a trajetória quadrada utilizando o sistema de controle proposto. Neste caso, a caneta de feltro não flutuou e nem arrastou, o que é o resultado esperado. Os dados experimentais mostraram que o controlador foi capaz de reduzir significativamente o erro anteriormente observado ao desenhar a trajetória quadrada.



(a) Baixa força (sem controle)



(b) Alta força (sem controle)



(c) Força Estável (controle LSM)

Figura 1: Exemplo do Robô BAXTER desenhando em um quadro branco com e sem controle no eixo Z

Apesar do exemplo ser uma simples trajetória quadrada, teoricamente o sistema pode aprender, e se necessário corrigir, qualquer trajetória 2D. O sistema de controle apresentado por (AZAMBUJA; CANGELOSI; ADAMS, 2016) também realizou trajetórias circulares e triangulares.

1.1 Objetivo Geral

A presente dissertação tem como objetivo:

- Desenvolver e avaliar um sistema de controle para redução de erro de posicionamento, utilizando o paradigma de *Spiking Neural Networks*, capaz de ser exportado para um computador neuromórfico, a fim de auxiliar um robô em sua tarefa já existente de desenhar uma forma quadrada sobre papel, visando reduzir a susceptibilidade a erros.

1.1.1 Objetivos Específicos:

- Aumentar o desempenho da tarefa usando e integrando dados multisensoriais, utilizando dados de distância e força;
- Projetar uma rede neural de controle capaz de ser compatível com o hardware SpiNNaker, para posterior portabilidade;
- Avaliar a capacidade de uma Rede Neural Recorrente em aprender o comportamento de um controlador PID;
- Contribuir com o trabalho que está em desenvolvimento na University of Plymouth (UK) com base na utilização de robôs colaborativos e Redes Neurais Pulsadas, utilizando o mesmo modelo neural para maior compatibilidade.

1.2 Contribuição

A principal contribuição deste trabalho está na investigação do uso de Redes Neurais baseadas em Spiking Neural Networks para sistemas de controle. Este interesse surge a partir do momento que novos hardwares biologicamente inspirados no cérebro humano são desenvolvidos. Também conhecidos como computadores neuromórficos, alguns implementados de forma digital, outros analógica, construídos com o objetivo de processador dados de redes SNN mais rápido que um computador. Diversos projetos buscam entender melhor como o cérebro humano funciona, um dos maiores é o *The Human Brain Project*, que disponibiliza cerca de 1 bilhão de euros para pesquisa científica em neurociência, dentre elas desenvolver modelos computacionais, análise de dados e robótica inspirados no cérebro.

2 FUNDAMENTAÇÃO TEÓRICA

Para uma compreensão das redes neurais baseadas no paradigma de Máquinas de Estado Líquido, e do sistema aqui proposto, o conhecimento do funcionamento desse tipo de rede, bem como suas etapas de aprendizado, comportamento e limitações são necessários. Portanto, este capítulo focará em uma breve revisão sobre Redes Neurais Artificiais baseadas em neurônios pulsantes, bem como seu funcionamento. Também será explicado como as LSMs estão inseridas no paradigma de *Reservoir Computing* e como pode ser feito e equacionado da etapa de aprendizado das redes. Por fim é descrito o funcionamento do sistema de controle de trajetória, em que este trabalho foi projetado para funcionar em paralelo.

2.1 Redes Neurais Artificiais

Partindo da hipótese de que a atividade de pulsos de um neurônio é o elemento chave para o processamento de informação no cérebro, redes neurais artificiais (em inglês, *Artificial Neural Networks*, ANN) têm sido utilizadas para simular e ajudar a compreender estruturas neurais maiores. As ANNs consistem essencialmente de uma população de nós não-lineares que modelam neurônios biológicos, trocando pulsos (*spikes*) entre si através de conexões ponderadas, refletindo a maneira como os neurônios reais projetam para os outros e interagem através de sinapses. Assim, a informação processada por cada neurônio é modulada pela atividade da rede em que está inserido, a qual, por sua alta dimensionalidade e pela não-linearidade intrínseca de seus elementos, pode apresentar vários tipos de comportamentos coletivos. Além disso, essas estruturas podem "aprender" como realizar uma tarefa específica e constituir uma parte significativa dos tópicos em inteligência artificial e aprendizado de máquina.

2.1.1 Redes Neurais Feedforward

Redes Neurais Feedforward (em inglês, *Feed-forward Neural Networks* - FNNs) foram as primeiras redes artificiais de neurônios pulsantes a serem introduzidas (BIBIS; GEORGIPOULOS, 1994). Compostas de várias camadas de neurônios (camadas de entrada, saída e camada(s) oculta(s)) e processam a informação em apenas uma direção, de uma camada para outra, em cada passo de tempo. Essa arquitetura é mostrada na Figura 2(a), partindo da camada de entrada e finalizando na camada de saída.

Do ponto de vista matemático, as FNNs são mapas estáticos de entradas-saídas: a ativação passa de camada para camada e o sistema não possui um estado in-

terno dinâmico. De um ponto de vista biológico, não há evidências da existência de estruturas *Feedforward* dentro de conjuntos neuronais.

2.1.2 Redes Neurais Recorrentes

As redes neuronais recorrentes (RNNs) mostram pelo menos um caminho cíclico de conexões sinápticas. Portanto, em um dado momento t , a ativação de um dado neurônio dentro desse ciclo não depende apenas da ativação de neurônios que se projetam no tempo $t - 1$, mas também da atividade da rede no tempo $t - 2$. Ou seja, a rede possui uma certa memória. Devido à estrutura recorrente, os neurônios podem esperar que sua ativação no tempo t seja influenciada pela atividade da rede até cerca de $t - T$, se ela tomar T passos para cobrir o caminho cíclico (veja a Figura 2(b)).

Opostamente às FNNs, as RNNs são sistemas dinâmicos. Sua estrutura fornece um estado interno dinâmico com memória interna, como a ativação dos neurônios é enviada de volta na rede e depende de mais do que apenas entradas anteriores, as RNNs se tornam mais adequadas para lidar com entradas dependentes do tempo. As RNNs também representam uma abordagem mais plausível para modelos computacionais biologicamente inspiradas, já que todas as redes neurais reais apresentaram conexões recorrentes.

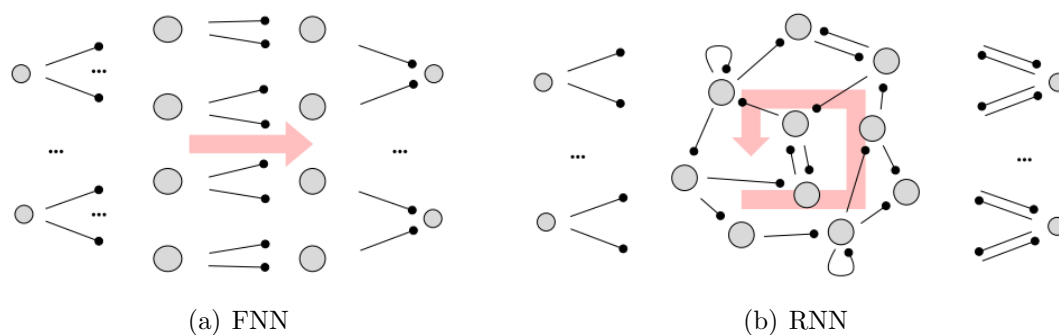


Figura 2: Estrutura genérica de redes dos tipos FNNs e RNNs.

Este trabalho é baseado na utilização de RNNs, utilizando um modelo de neurônios pulsantes (*spiking*), todas as discussões seguintes serão focadas neste modelo.

2.2 Elementos de Sistemas Neurais

O objetivo desta seção é introduzir noções elementares da neurociência, em particular, os conceitos de potenciais de ação (*action potentials*), potenciais pós-sinápticos (*postsynaptic potentials*), limiares de disparo (*firing thresholds*), e refratividade (*refractoriness*). O intuito aqui é prover explicações básicas e simplistas como um ponto de partida para o entendimento da codificação, ou modelagem, neural. Para discussões mais aprofundadas sobre o tópicos de neurobiologia recomenda-se a literatura apresentada em (GERSTNER; KISTLER, 2002, 37), onde uma revisão bibliográfica para modelos neurais é feita de forma incisiva.

Um neurônio pode ser dividido em três partes funcionalmente distintas: dendritos (em inglês, *dendrites*), soma (em inglês, *soma*) e axônio (*axon*). De uma forma simplificada, as dendritos atuam como um dispositivo de entrada, coletando sinal de outros neurônios e transmitindo eles para a soma. A soma trabalha como um circuito

RC acoplado de um circuito comparador, se a carga do circuito RC excede um certo limiar, um sinal de saída é gerado. O sinal de saída é distribuído pelo dispositivo de saída, o axônio, que entrega o sinal aos neurônios conectados. A Figura 3(a) ilustra o um único neurônio onde dendrito, axônio e soma podem ser facilmente distintos. O círculo tracejado mostra o sinal de saída de um neurônio, chamado de potencial de ação.

A junção entre dois neurônios é chamada de sinapse. Na literatura, o neurônio que está enviando um sinal através da sinapse é chamado de célula pré-sináptica e o neurônio receptor como célula pós-sináptica. Essa estrutura é exemplificada na Figura 3(b), onde j é o neurônio pré-sináptico e i o pós-sináptico, o pulso gerado por j é enviado pelo axônio e recebido pelos dendritos de i , a região onde ocorre a sinapse está marcada pelo círculo pontilhado.

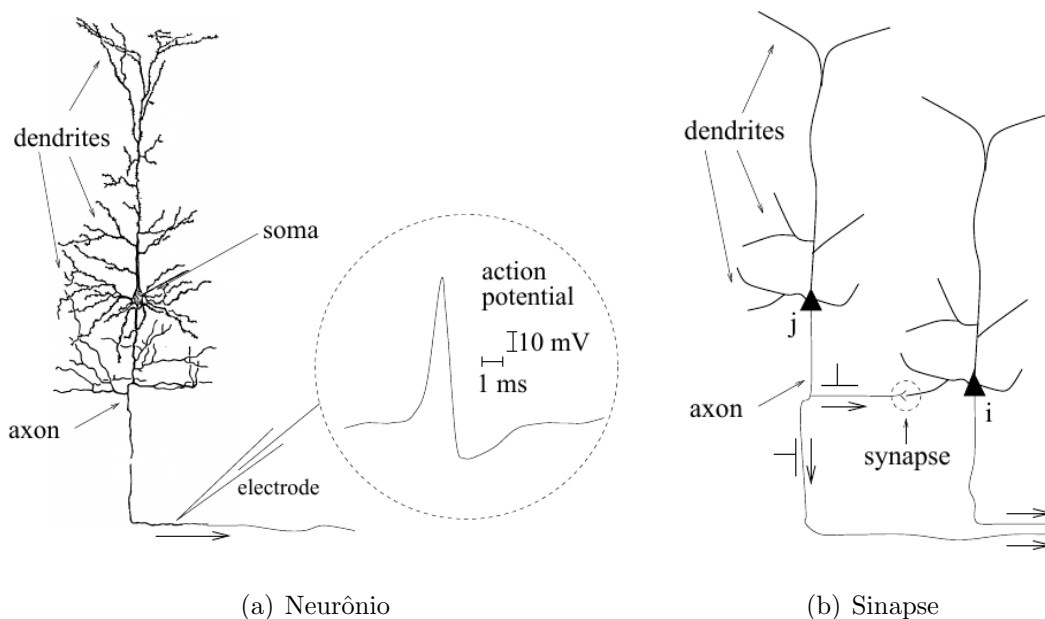


Figura 3: Estrutura básica de um neurônio e ocorrência de uma sinapse. Fonte: Adaptado de (GERSTNER; KISTLER, 2002)

2.2.1 Trem de Pulsos

Um sinal neural consiste de curtos pulsos elétricos, também chamados de potenciais de ação (*action potentials*) ou *spikes*, tem duração aproximada de $1-2ms$ e uma amplitude de cerca de $100mV$ (ver Figura 3(a)). Uma cadeia de pulsos emitidos por um neurônio é chamado de trem de pulsos (*spike train*), uma sequência de eventos em intervalos regulares ou irregulares de tempo. Como a forma do potencial de ação não muda ao se propagar no axônio e todos possuem o mesmo formato, sua forma não carrega nenhuma informação. O que importa são a quantidade e temporização dos pulsos enviados.

Os *spikes* em um trem de pulsos são, normalmente, bem separados, é impossível excitar um segundo pulso durante ou imediatamente após o primeiro ter sido disparado. A distância mínima entre dois pulsos define o período refratário absoluto do neurônio. Este é seguido por uma fase de refratividade relativa, onde é difícil, mas não impossível, excitar outro potencial de ação.

2.2.2 Sinapses

Como mencionado anteriormente, a junção entre um o axônio de um neurônio pré-sináptico e o dendrito de uma célula pós-sináptica é chamado de sinapse. Quando um pulso chega em uma sinapse, uma complexa cadeia de processos bio-químicos é desencadeada que culminam na liberação de um neurotransmissor do terminal pré-sináptico na fenda sináptica (uma pequena lacuna entre as membranas pré e pós-sinápticas). Receptores especializados detectam as moléculas transmissoras, resultando em fluxo de íons, que no fim serão transformados em uma resposta elétrica. A resposta em tensão do neurônio pós-sináptico à ação de potencial pré-sináptico é chamado de potencial pós-sináptico.

O efeito de um pulso no neurônio pós-sináptico pode ser dado por uma diferença de potencial $v(t)$ entre o interior da célula e seu ambiente, esta diferença de potencial é chamada de potencial da membrana. Sem nenhum pulso na entrada, o neurônio em repouso possui um potencial de membrana constante, v_{rest} . Após a ocorrência de uma sinapse o potencial se altera. Se a mudança é positiva, a sinapse é dita como excitatória, caso negativa, a sinapse é inibitória.

2.2.3 Potenciais Pós-sinápticos

Dado um neurônio i e seu potencial de membrana $v_i(t)$. Antes da chegada de qualquer pulso de entrada, tem-se $v_i(t) = v_{rest}$. Caso um neurônio pré-sináptico j dispare um pulso em $t = 0$, observa-se, para $t > 0$

$$v_i(t) - v_{rest} =: \epsilon_{ij}(t) \quad (1)$$

No lado direito da equação 1, $\epsilon_{ij}(t)$ define o potencial pós-sináptico (*postsynaptic potential* - PSP). Se a diferença de tensão for positiva, tem-se um PSP excitatório (EPSP), caso negativo, um PSP inibitório (IPSP). Esse comportamento pode ser observado na ilustração apresentada na Figura 4(a), cada pulso recebido em i evoca um EPSP que pode ser medido como $v_i(t) - v_{rest}$.

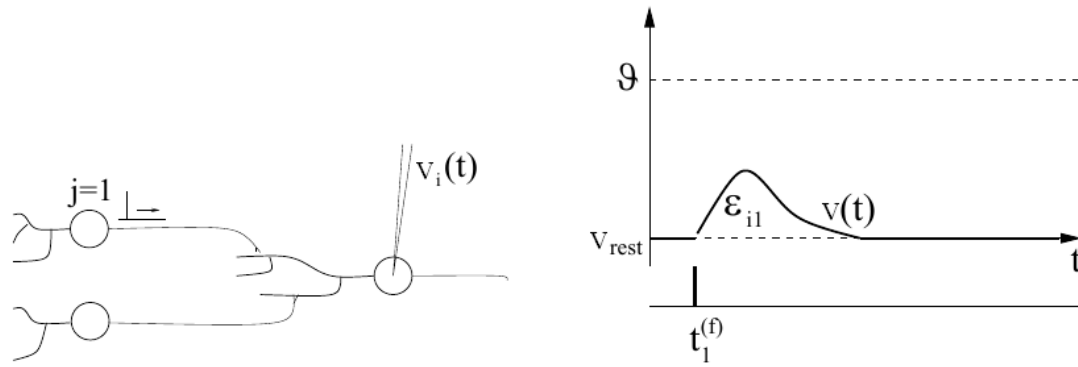
2.2.4 Limiar de disparo e Potencial de Ação

Dado dois neurônios pré-sinápticos $j = 1, 2$, onde ambos enviam pulsos ao neurônio pós-sináptico i . O neurônio $j = 1$ dispara nos tempos $t_1^{(1)}, t_1^{(2)}, \dots$, e o neurônio $j = 2$ nos tempos $t_2^{(1)}, t_2^{(2)}, \dots$. Cada pulso gera um PSP ϵ_{i1} ou ϵ_{i2} . Enquanto houver poucas sinapses excitatórias, o potencial da membrana pós-sináptica, $v_i(t)$, é aproximadamente a soma dos PSPs individuais,

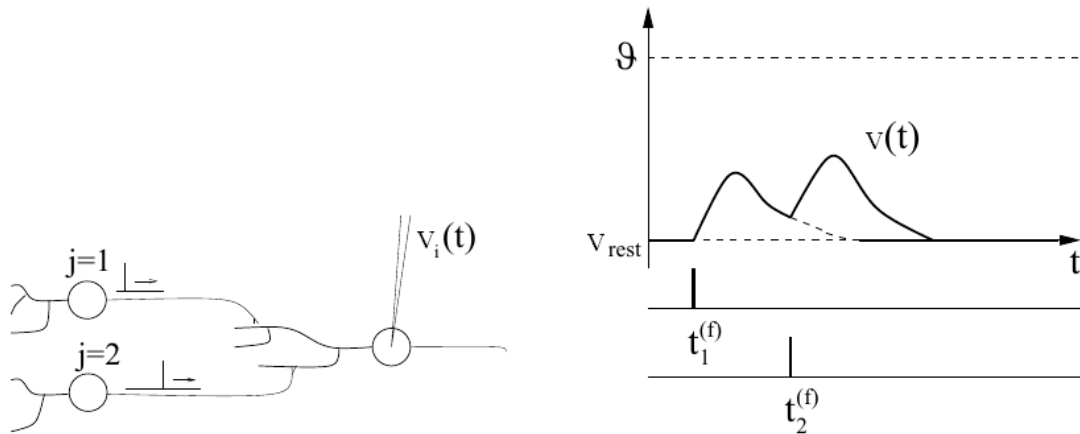
$$v_i(t) = \sum_j \sum_f \epsilon_{ij}(t - t_j^{(f)}) + v_{rest} \quad (2)$$

ou seja, o potencial da membrana responde linearmente aos pulsos de entrada, isso pode ser observado na Figura 4(b), onde o neurônio pós-sináptico i recebe apenas dois EPSPs. Porém, a linearidade não se mantém caso muitas sinapses excitatórias ocorram em um curto espaço de tempo. Assim que o potencial de membrana atinge um valor crítico ϑ , sua trajetória mostra um comportamento diferente de uma simples soma de PSPs, $v_i(t)$ apresenta uma excursão com cerca de $100mV$ de amplitude, ou seja, um pulso ou potencial de ação. Este potencial de ação vai propagar através do axônio do neurônio i para as sinapses dos outros neurônios. Após o pulso o potencial da membrana não volta diretamente para o potencial de repouso v_{rest} ,

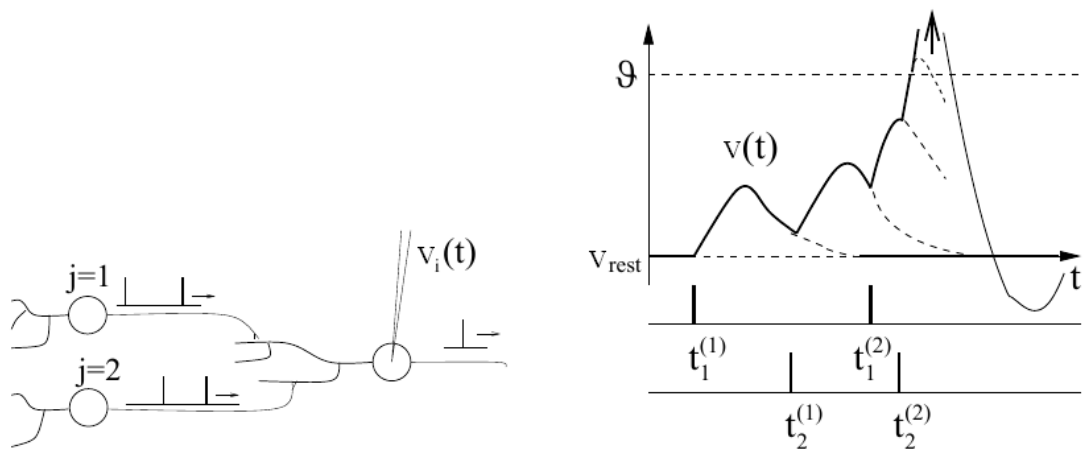
passando por uma fase de hiperpolarização abaixo do valor de descanso. A Figura 4(c) ilustra esse comportamento, onde quatro pulsos consecutivos fazem com que o potencial da membrana $v(t)$ ultrapasse o limiar ϑ , fazendo com que o neurônio pós-sináptico i gere um pulso.



(a) Uma sinapse excitatória



(b) Duas sinapses excitatórias



(c) Quatro sinapses excitatórias

Figura 4: Exemplo simplificado para ocorrência de sinapses e disparos. Fonte: Adaptado de (GERSTNER; KISTLER, 2002)

2.3 Reservoir Computing

Em *Reservoir Computing*, um sistema dinâmico, o reservatório, é alimentado com um sinal de entrada. A dinâmica do reservatório mapeia a entrada para um sistema de maior dimensionalidade. Uma função de ativação (*readout*) é treinada para ler o estado do reservatório e dar o resultado esperado. A Figura 5 mostra a estrutura de um *framework Reservoir Computing* onde a entrada é representada como um trem de pulsos que alimentam os neurônios do reservatório que possui diversas conexões recorrentes entre os neurônios e apenas as conexões de sinapse tracejadas são treinadas (função de *Readout*).

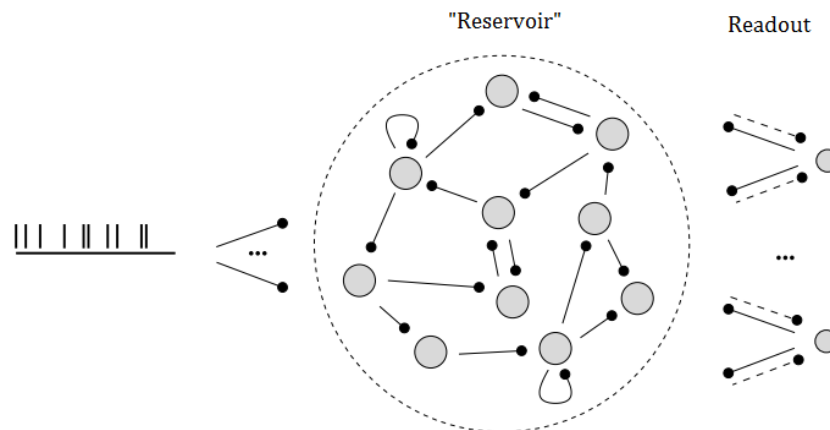


Figura 5: Estrutura de uma RNN do framework Reservoir Computing.

Qualquer tipo de rede ou sistema, que tenha dinâmica interna suficiente para obter realizar o aprendizado, pode ser escolhido como reservatório do sistema. Vários tipos de reservatórios são usados na literatura. Em (JAEGER, 2001), por exemplo, *Echo State Machines* são usadas como *Reservoir* para uma rede recorrente, onde é apresentado uma análise profunda desse tipo de *framework*. Já os autores de (FERNANDO; SOJAKKA, 2003) mostraram que as ondas criadas em um reservatório com água e discretizadas por sensores podem ser usadas como reservatório, para identificar local de origem das perturbações no líquido. O reservatório utilizado neste trabalho é uma Rede Neural Pulsada (em inglês, *Spiking Neural Network* - SNN). Este tipo de *framework* é chamado de Máquina de Estado Líquido (em inglês, *Liquid State Machine* - LSM), apresentada pela primeira vez em (MAASS; NATSCHLÄGER; MARKRAM, 2002).

Em uma SNN, a ideia principal é que os neurônios da rede não ativem um ciclo de propagação em um intervalo de tempo constante conforme redes neurais convencionais e sim quando seu potencial atinge um certo limite. Quando um neurônio ativa, ou dispara, ele gera um sinal que viaja para os outros neurônios que por sua vez aumentam ou diminuem seu potencial de acordo com o sinal recebido e o tipo de conexão. Existem vários modelos de neurais disponíveis na literatura que podem ser utilizados nessas redes. Neste trabalho, seguindo a implementação em (AZAMBUJA; CANGELOSI; ADAMS, 2016), utilizou-se o modelo neural Integra-e-Dispara com vazamento. Os parâmetros utilizados para a LSM, o modelo neural e treinamento da função de ativação serão discutidos nas seções 2.3.1, 2.3.2 e 2.3.3 respectivamente.

2.3.1 Liquid State Machines

Uma Máquina de Estados Líquidos (LSM - Liquid State Machine) consiste de neurônios aleatoriamente e recorrentemente conectados entre si (o "líquido"), que recebem séries temporais de pulsos (*spikes*) como entrada. O líquido, ou reservatório, pode ser visto como uma função aleatória da entrada.

O conceito de LSM pode ser explicado como um reservatório de nós recorrentemente iterativos, estimulado por uma entrada $u(t)$. Um estado $x(t)$ é extraído a partir do estado da rede e uma função de leitura f^M converte o estado de alta dimensão $x(t)$ em uma saída $y(t)$. A figura 6 mostra um diagrama com esta arquitetura, onde uma entrada analógica é convertida para um vetor de pulsos que excitam o líquido, uma saída é obtida a partir do estado do reservatório. Os laços presentes nas conexões recorrentes entre os nós do reservatório criam uma forma de memória de curto prazo, onde os efeitos das entradas permanecem dentro da rede por um certo tempo, de acordo com os parâmetros do líquido.

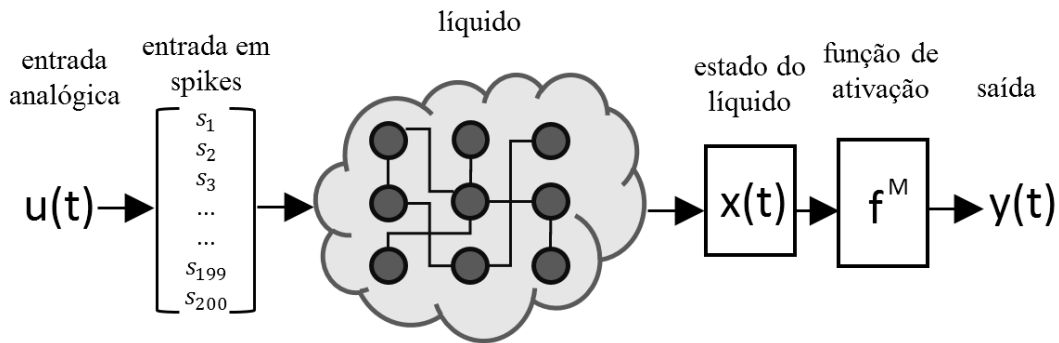


Figura 6: Diagrama representando um modelo de Máquina de Estado Líquido.

2.3.2 Modelo neural Integra-e-Dispara

Uma LSM, normalmente, é implementada utilizando neurônios pulsados do tipo Integra-e-Dispara com vazamento (LIF - *Leaky Integrate and Fire*) (GERSTNER; KISTLER, 2002, p.94), conectados de forma recorrente como sugerido em (MAASS; NATSCHLÄGER; MARKRAM, 2002) formando uma rede de Pequeno Mundo (SWN - *Small World Network*). As SWNs são representadas por grafos matemáticos onde grande parte das conexões são estabelecidas entre os vértices mais próximos.

O simulador para os líquidos utilizado neste trabalho implementa uma variante do modelo neural LIF com correntes exponenciais. Esse modelo LIF comporta-se como um circuito capacitor-resistor, com um circuito adicional para gerar o pulso (potencial de ação) e também manter ele descarregado durante o período refratário (GERSTNER; KISTLER, 2002, p.101).

Uma representação desse modelo é mostrado na Figura 7(a), onde a corrente $I(t)$ carrega o circuito RC, e a tensão $v(t)$ ao longo da capacitância (pontos) é comparada com um limiar ϑ . Se $v(t) = \vartheta$ no tempo $t_i^{(f)}$ um pulso de saída $\delta(t - t_i^{(f)})$ é gerado. Na Figura 7(b) um pulso pré-sináptico $\delta(t - t_j^{(f)})$ é filtrado por um passa-baixa na sinapse e gera um pulso de corrente $\alpha(t - t_j^{(f)})$ como entrada para o neurônio i .

O modelo LIF pode ser parcialmente descrito pelo conjunto de equações diferenciais na Equação 3, onde c_m é a capacitância da membrana (em F), τ_m é a constante de tempo (em s) da membrana, τ_{syn_i} e τ_{syn_e} são os tempos (em s) de decaimento

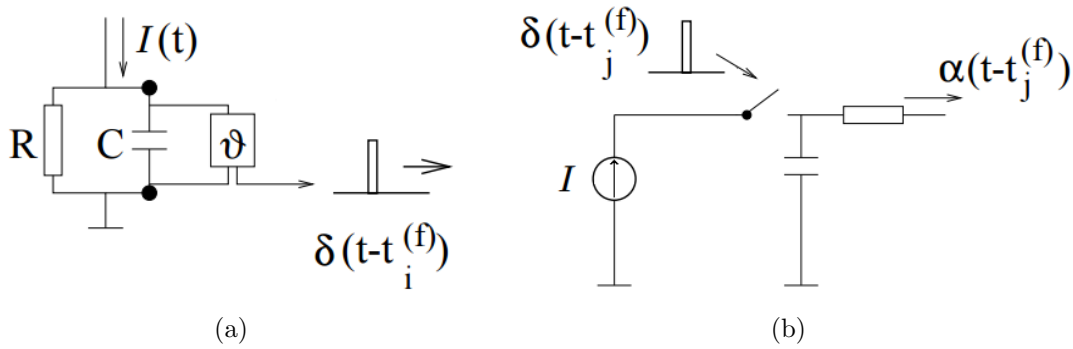


Figura 7: Diagrama esquemático de um neurônio Integra-e-Dispara. Fonte: Adaptado de (GERSTNER; KISTLER, 2002)

das sinapses inibitória e excitatória respectivamente, v_{rest} potencial de membrana de repouso (em V), i_{offset} e i_{noise} as correntes de ruído fixa e variável (em A), respectivamente. As correntes $i_e(t)$ e $i_i(t)$ (em A) correspondem, respectivamente, às correntes excitatórias e inibitórias do neurônio.

$$\frac{dv}{dt} = \frac{i_e(t) + i_i(t) + i_{offset} + i_{noise}}{c_m} + \frac{v_{rest} - v}{\tau_m} \quad (3a)$$

$$\frac{di_e}{dt} = -\frac{i_e}{\tau_{syn_e}} \quad (3b)$$

$$\frac{di_i}{dt} = -\frac{i_i}{\tau_{syn_i}} \quad (3c)$$

A primeira parcela da equação 3a corresponde às variações de tensão na membrana devido as correntes de entrada e saída, conexões excitatórias e inibitórias do neurônio, juntamente com as influências do ruído da rede. A segunda parcela corresponde à tendência da membrana em buscar o potencial de repouso v_{rest} , se $v > v_{rest}$ há um decaimento devido essa parcela, se $v < v_{rest}$ há um aumento do potencial da membrana. Esta situação ocorre após um pulso do neurônio, como pode ser visto na Figura 4(c), onde o potencial v é menor que v_{rest} após a ocorrência do pulso.

2.3.3 Machine Learning

O esquema global do Aprendizado de Máquina é simples: a rede neural, que é um filtro da função de entrada $u(t)$, gera uma saída $y(t) = F_{net}(t)$ que é comparada a uma saída esperada $y_0(t) = F_{target}(t)$ em cada *timestep*. O procedimento de treinamento consiste em adaptar todos os pesos sinápticos da rede de modo que sua saída se ajuste à saída de destino. Dessa forma, após um treinamento suficientemente longo, o filtro F_{net} deve ser capaz de reproduzir a saída de F_{target} para entradas diferentes da entrada de treinamento.

Neste trabalho, para treinar os pesos (w_1 a w_{200}) da função de leitura, foi utilizada uma regressão linear. Os Mínimos Quadrados Ordinários (OLS - *Ordinary Least Squares*) encaixam um modelo com coeficientes $w = (w_1, w_2, \dots, w_p)$ solucionando o problema em (4), onde a matriz \mathbf{X} contém os valores filtrados pela membrana passa-baixa dos pulsos do líquido e y a saída esperada.

$$\min_w \|\mathbf{X}w - y\|_2^2 \quad (4)$$

Quando os dados de entrada entre ensaios distintos são correlacionados, as colunas de \mathbf{X} são aproximadamente linearmente dependentes. Consequentemente, a estimativa dos mínimos quadrados torna-se altamente sensível a erros aleatórios, produzindo uma grande variância. A regressão de Ridge (HOERL; KENNARD, 1970) foi realizada em cada líquido para encontrar o correspondente vetor de pesos para a função de leitura da LSM. Esse método de regressão linear impõe uma penalidade ($\alpha \geq 0$) sobre o tamanho dos coeficientes, como pode ser visto em (5). Mesmo que \mathbf{X} não tenha rank completo, a regressão ainda é solucionável.

$$\min_w \|\mathbf{X}w - y\|_2^2 + \alpha \|w\|_2^2 \quad (5)$$

2.4 Controle de robô com Spiking Neural Network

Nesta seção é feita uma breve apresentação das redes de controle apresentadas em (AZAMBUJA; CANGELOSI; ADAMS, 2016), nas quais baseia-se este trabalho. A fim de deixar o objetivo e motivação do controlador de eixo Z apresentado neste trabalho mais claro.

Azambuja apresenta uma nova implementação de SNNs para controle, utilizando uma abordagem baseada em redes diversas e paralelas, com movimentos biologicamente inspirados. O trabalho utiliza o robô humanoide BAXTER para desenhar três formas diferentes em uma superfície plana. As trajetórias são inspiradas nos movimentos humanos, que conforme relatado em (FLASH; HOGAN, 1985), seguem um perfil de velocidade gaussiana. A Figura 8 mostra as três trajetórias utilizadas, o perfil de velocidade, e o efeito pode ser visto pela concentração de pontos nas arestas das formas.

As trajetórias do plano 2D são traduzidas para trajetórias angulares das quatro juntas utilizadas para o robô realizar o desenho. A Figura 9 mostra as trajetórias de cada uma das quatro juntas, bem como o movimento cartesiano X e Y, para desenhar um quadrado. As trajetórias são geradas no simulador V-REP, utilizando cinemáticas inversas (*inverse kinematics* - IK), mantendo a posição em Z constante.

No trabalho de Azambuja é apresentado a implementação de um *framework* de LSMs paralelas para o sistema de controle, as redes, distintas em suas conexões, são treinadas individualmente para seguir as trajetórias das juntas (Figura 9). Na Figura 10, tem-se um diagrama da estrutura de uma LSM de controle, onde os ângulos das juntas, apresentados anteriormente, são usados como entrada. O próximo valor angular é calculado pela rede e realimenta a mesma. Dessa forma a LSM apresenta o comportamento de um atrator para a trajetória que foi treinada. Os valores de ganho e *bias* extraídos na entrada, e aplicados na saída, são utilizados para tradução dos ângulos em radianos para um índice dos neurônios da camada de entrada. Esses valores são encontrados na etapa de treinamento. Na implementação paralela, realimenta-se as redes com a média das saídas, conforme diagrama apresentado na Figura 11. A variável *input* apresenta apenas os ângulos que devem alimentar a rede em *step* = 0.

Como este controlador não apresenta controle no eixo Z, oscilações acontecem devido à incerteza das trajetórias criadas pelas redes e ruído gerado pelas mesmas. Em simulações no simulador V-REP, as variações no eixo Z com este controlador apresentaram flutuações de $1.93mm$ entre valores mínimos e máximos da posição Z retornada pelo simulador. Porém quando utilizadas no robô real, sem uma superfície de contato ("desenhando no ar"), as variações foram aproximadamente $6,3mm$,

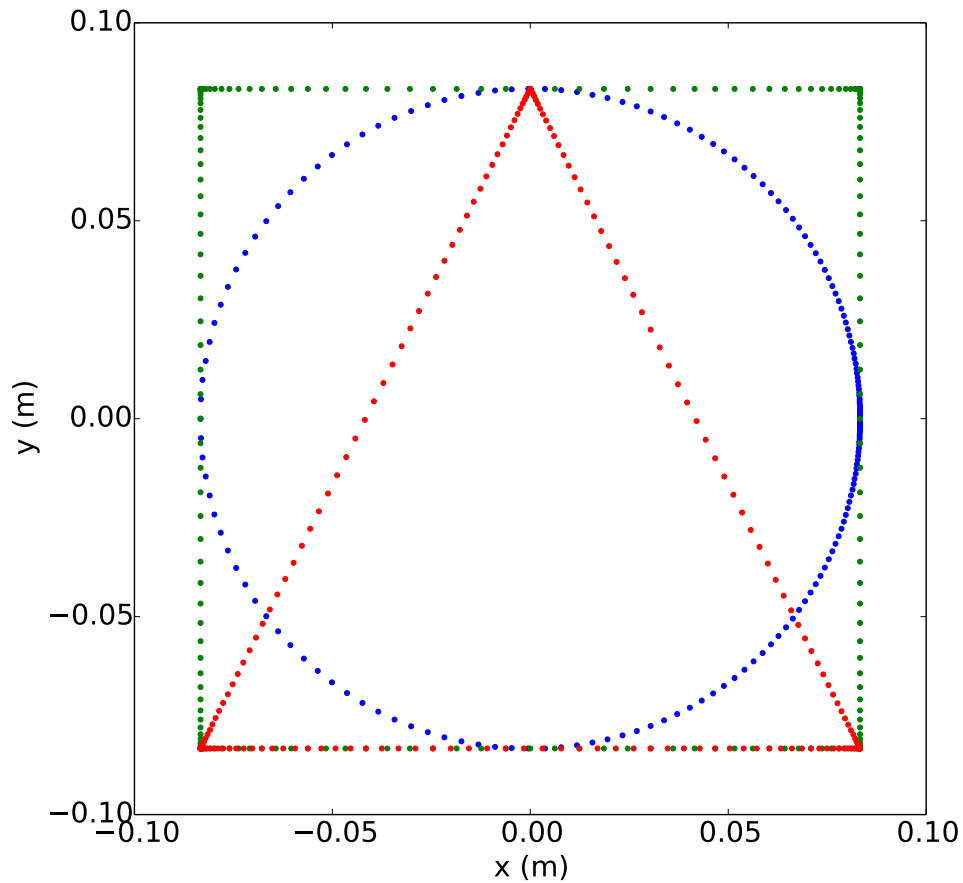


Figura 8: Trajetórias 2D utilizadas na rede de controle. Fonte: (AZAMBUJA; CANGELOSI; ADAMS, 2016)

devido a ruídos externos ao sistema, como ruído inserido pelas juntas do robô e peso do braço, que não estão modeladas no simulador. Impossibilitando o sistema de ser empregado no robô real sem um controle para o posicionamento no eixo Z.

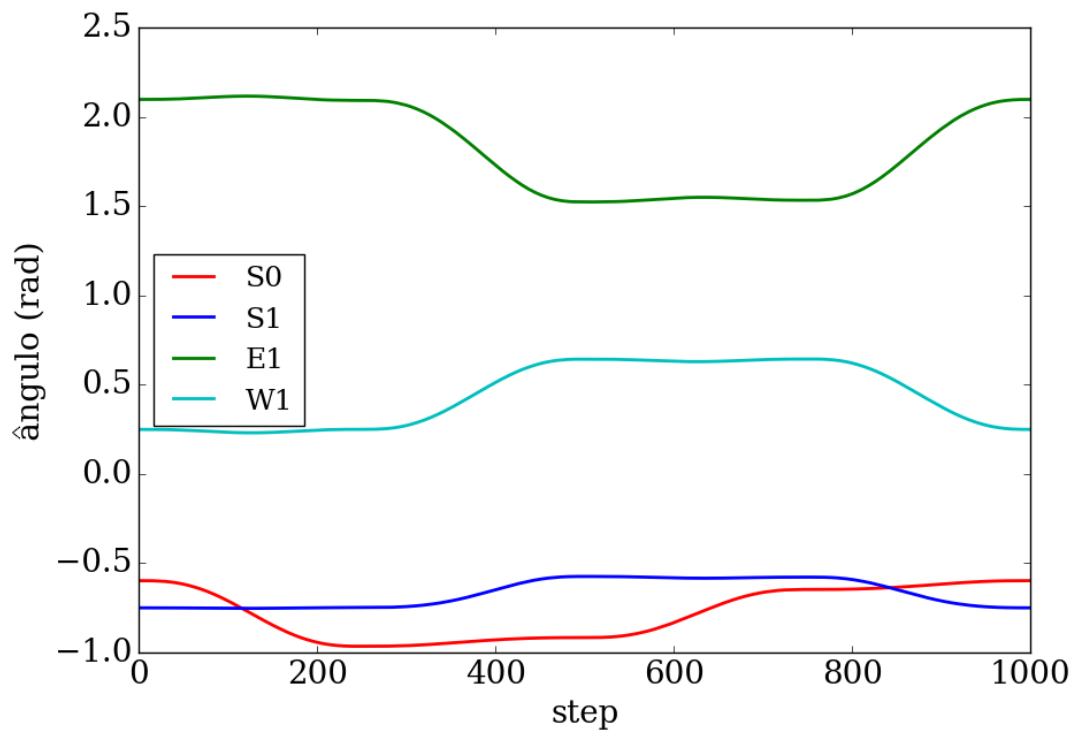


Figura 9: Ângulos das juntas utilizados para realizar a trajetória quadrada com o robô BAXTER.

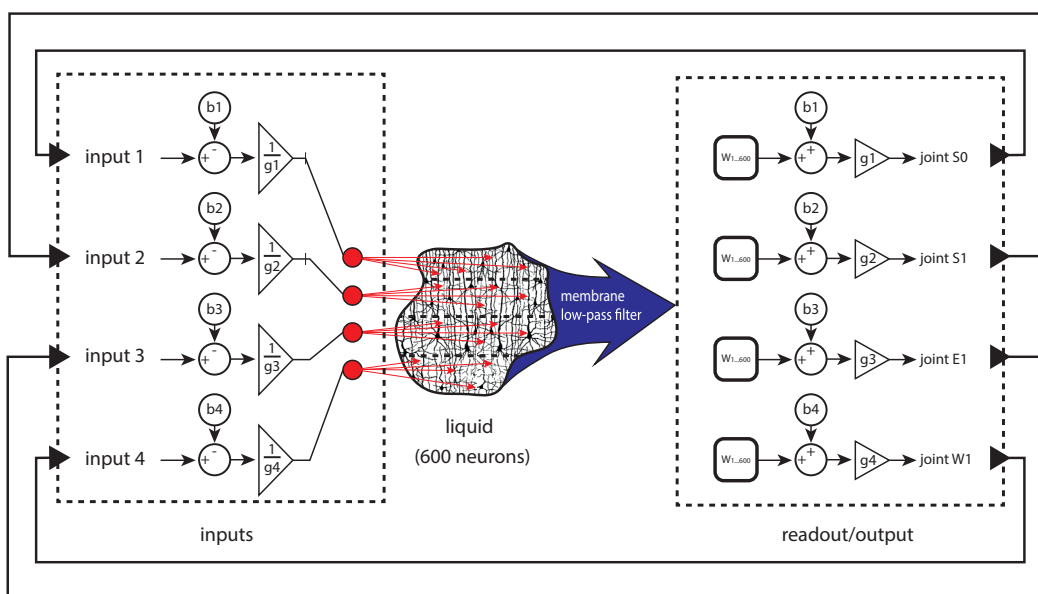


Figura 10: Disposição de uma LSM para controle de trajetória. Fonte: (AZAMBUJA; CANGELOSI; ADAMS, 2016)

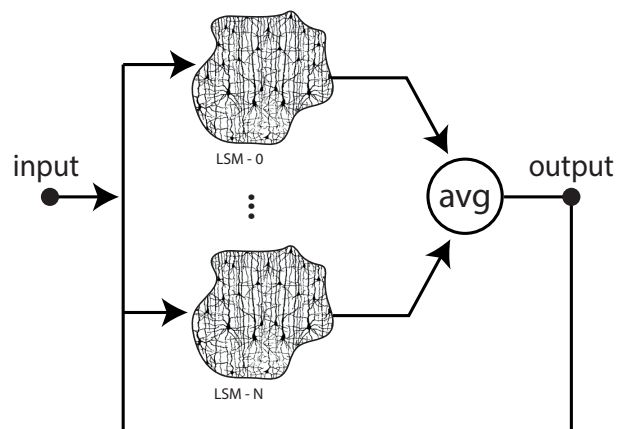


Figura 11: Disposição de LSMs paralelas. Fonte: (AZAMBUJA; CANGELOSI; ADAMS, 2016)

3 MATERIAIS E MÉTODOS

Até o momento foram apresentados os conceitos necessários para o um entendimento básico de Redes Neurais Recorrentes e Máquinas de Estado Líquido, assim como uma visão geral do funcionamento dos neurônios. Neste capítulo apresenta-se a metodologia aplicada para a criação do conjunto de treinamento, como os líquidos das redes foram configurados e a realização da sua etapa de aprendizado, assim como a escolha da melhor rede após testes iniciais. Também é feita uma breve apresentação sobre o robô BAXTER, sensores e ferramentas utilizadas.

Um breve sumário da metodologia aplicada é apresentado a seguir:

1. Criação de um conjunto de treinamento usando dois controladores PID;
2. N Líquidos com conexões aleatórios foram criados;
3. Dados do conjunto treinamento são convertidos em trens de pulsos e os líquidos criados são simulados com essas entradas;
4. A partir dos líquidos simulados são calculados os pesos da função de ativação de cada uma das N LSMs;
5. Cada rede é avaliada individualmente através de experimentos usando o robô BAXTER. A rede com os melhores resultados é escolhida.

Quanto as ferramentas de programação que foram utilizadas para implementação do sistema e realização dos experimentos:

- O robô BAXTER foi comandado usando o *Robot Operating System* (ROS) (QUIGLEY et al., 2009);
- As LSMs foram criadas e simuladas usando a biblioteca BEE ¹;
- As trajetórias das juntas foram calculadas utilizando o simulador *Virtual Robot Experimentation Platform* (V-REP) (ROHMER; SINGH; FREESE, 2013);
- Este projeto foi desenvolvido utilizando IPython Notebooks.

Todo o código fonte e conjuntos de dados de treinamento estão disponíveis online².

¹Disponível em <https://github.com/ricardodezambuja/LiquidStateMachine-Python>

²Código fonte e conjuntos de dados disponível em <https://github.com/davi-sala/I2MTC2017-LSMFusion>

3.1 Robô BAXTER

O robô colaborativo BAXTER, da Rethink Robotics Inc., é um robô projetado para trabalhar entre humanos com segurança. O *framework* LSM desenvolvido para controlar BAXTER em (AZAMBUJA; CANGELOSI; ADAMS, 2016) foi projetado usando o Controle de Posição de Juntas³. Este modo permite que todos os mecanismos de segurança do robô sejam ativados durante a execução do movimento.

O robô BAXTER utiliza os Atuadores Elásticos em Série (SEAs - Series Elastic Actuators) (PRATT; WILLIAMSON, 1995), introduzindo uma mola entre os elementos do motor e a saída do atuador. Isso resulta em maior segurança, pois as molas podem ser deformadas por entradas de nível humano, mas em contra partida, adiciona ruído à saída. Os SEAs também permitem medir a saída de torque dos atuadores, possibilitando estimar a força em cada junta e também no ponto final (caneta de feltro neste caso). Ao lado do ruído introduzido pelos atuadores, uma mola externa que suporta a junta S1 (ver Figura 12) também adiciona interferências no movimento do braço, quando esta junta é utilizada.

A Figura 12 mostra como o robô e a superfície foram dispostos para a execução das trajetórias. Na figura também estão destacadas as quatro juntas utilizadas (S0, S1, E1 e W1), com seus devidos nomes e a orientação do sistema de coordenadas. A orientação da caneta é mantida perpendicular à superfície durante a trajeteto.

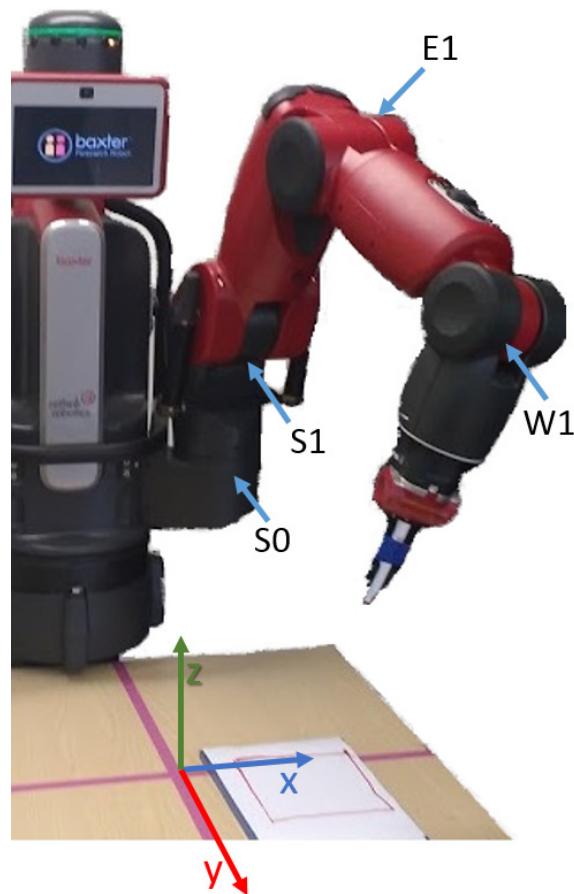


Figura 12: Disposição do robô BAXTER para executar a tarefa.

O ruído introduzido pelos atuadores do BAXTER resultam em erros de posicio-

³See http://sdk.rethinkrobotics.com/wiki/Arm_Control_Modes

namento no eixo Z ao mover o braço, a Figura 15 mostra a posição em Z do *endpoint* (ponta da caneta de feltro conectada ao *gripper*) registrada pelo robô. Este valor é estimado pelo modelo interno do robô, utilizando a posição das juntas.

Todos os robôs apresentam não linearidades na sua movimentação (SICILIANO et al., 2010), devido, principalmente, à restrições impostas pelas juntas. Espera-se que o controlador LSM seja capaz de compensar essas não linearidades a partir de um amplo conjunto de treinamento. Neste trabalho, utilizou-se quatro das sete juntas presente no braço do robô, limitando o espaço de trabalho do mesmo, ou seja, reduzindo a possibilidade de posições de braço disponíveis, porém, todas as trajetórias utilizadas estavam em posições solucionáveis no espaço de juntas.

O robô possui uma gama de sensores e estimativas de posição, força, torque, entre outros. Neste trabalho, utilizou-se os dados de distância e força por serem complementares nesta aplicação. O sensor de distância possui a informação que quantifica a distância entre a mão do robô e a superfície, enquanto a estimativa de força mostra o quão forte o robô está pressionando a caneta contra a superfície. Assim a primeira informação é importante enquanto o *endpoint* não toca a superfície, e a segunda enquanto contato é mantido.

3.1.1 Sensor de distância

O sensor de distância presente no *gripper* do robô BAXTER consiste de um sensor infra-vermelho, posicionado conforme mostra a Figura 13. A distância mínima detectada pelo sensor é de 4cm , e a máxima 40cm . A documentação do robô não disponibiliza maiores informações sobre o sensor, porém esse tipo de sensor não costuma ter um comportamento linear (OPTOELECTRONIC DEVICE, 2005), apresentando uma resolução maior em distâncias mais próximas. Ao analisar o comportamento do sensor entre as distâncias de $15 - 18\text{cm}$, intervalo de interesse utilizado neste trabalho, observou-se uma resolução de 3mm .

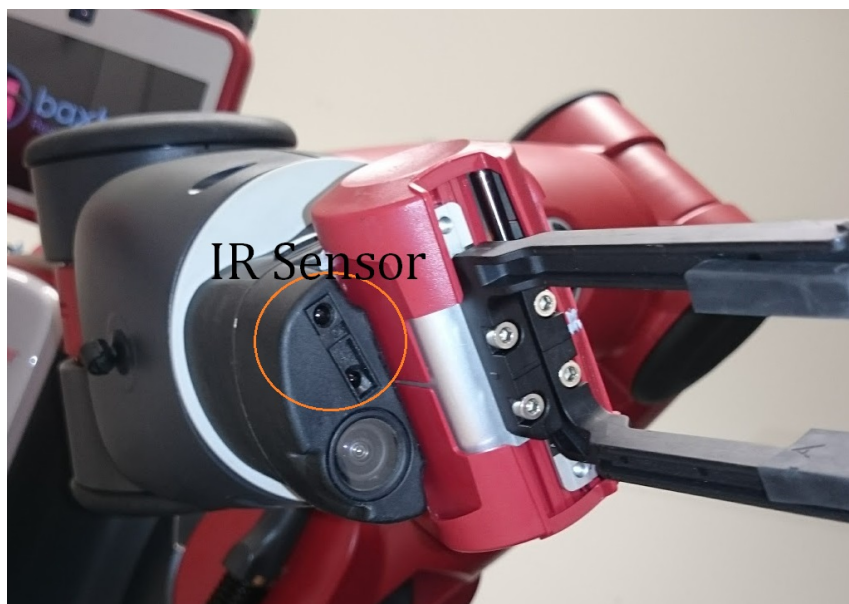


Figura 13: Localização do sensor de distância no *gripper* do robô BAXTER.

O valor deste sensor, em cm , pode ser obtido subscrevendo-se ao tópico ROS `/robot/range/<side>_hand_range`, sendo `<side>` o braço desejado, `left` ou `right`.

3.1.2 Força estimada

A estimativa de força no *endpoint* do braço do robô é realizada pelo sistema interno do robô. Esse valor é dado a partir das forças calculadas para cada junta, a partir do sensor de torque das molas presentes nos SEAs. A orientação do vetor de força lido é paralelo ao *endpoint* do robô. O braço, se mantido parado no ar, sem apoio, retorna uma estimativa de aproximadamente $1N$ no *endpoint*, esta medida é devida às forças exercidas pelo robô para compensar o peso do braço.

O valor de distância, em N , pode ser obtido subscrvendo-se ao tópico ROS `/robot/limb/<side>/endpoint_state/force`, sendo `<side>` o braço desejado, `left` ou `right`.

3.2 Trajetórias

Como trajetória, utilizou-se apenas a figura quadrada, porém, teoreticamente, o controlador não é dependente da trajetória, podendo funcionar com outras implementações. As curvas angulares das juntas foram gerada com 1000 pontos, em uma trajetória quadrada, 250 para cada arresta. Manteve-se o perfil de velocidade gaussiana para a trajetória, conforme mostrado na Figura 8. Os ângulos para as trajetórias das juntas, para executar esta forma, foram gerados usando cinemática inversa (SICILIANO et al., 2010) no simulador V-REP, enviando a posição em X e Y de cada ponto e lendo os valores das juntas do robô. Manteve-se a posição do eixo Z constante durante a geração da trajetória. Utilizou-se uma posição em Z mais elevada que as trajetórias utilizadas no trabalho de (AZAMBUJA; CANGELOSI; ADAMS, 2016), que foram executadas sem superfície, em uma posição inferior à altura das mesas que estavam disponíveis para uso.

Gerou-se cinco trajetórias quadradas, distanciadas por $5mm$ cada no eixo Z. A Figura 14 mostra as trajetórias para duas das quatro juntas, S1 e W1, ambas subtraídas do seu valor inicial ($step = 0$) para melhor visualização. Cada $step$ corresponde a um ponto, ou posição da trajetória.

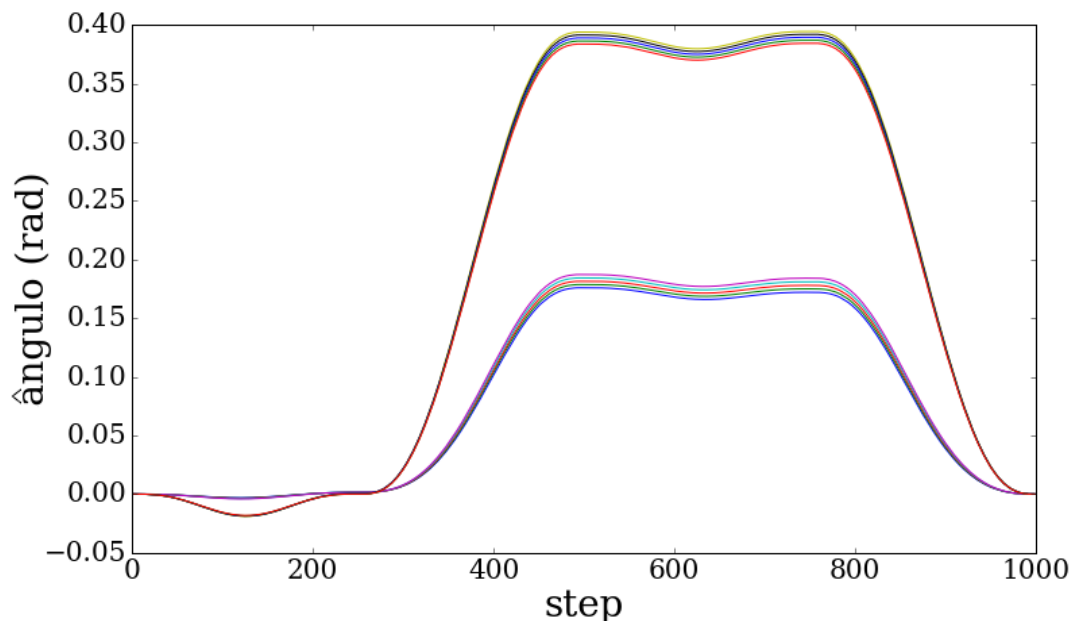


Figura 14: Cinco trajetórias das juntas juntas S1 e W1.

Ao executar a trajetória das juntas no simulador V-REP, obteve-se um valor constante para a distância no eixo Z ($160mm$), como esperado, uma vez que o Baxter simulado não possui nenhum tipo de ruído modelado. Quando a mesma trajetória foi executada utilizando o robô real, o valor médio observado, em 10 experimentos, foi de $\mu = -159mm$ com um desvio padrão de $\sigma = 1.27mm$. A figura 15 apresenta os valores de posição no eixo Z obtidos em simulação (linha pontilhada) e no robô real (linha contínua).

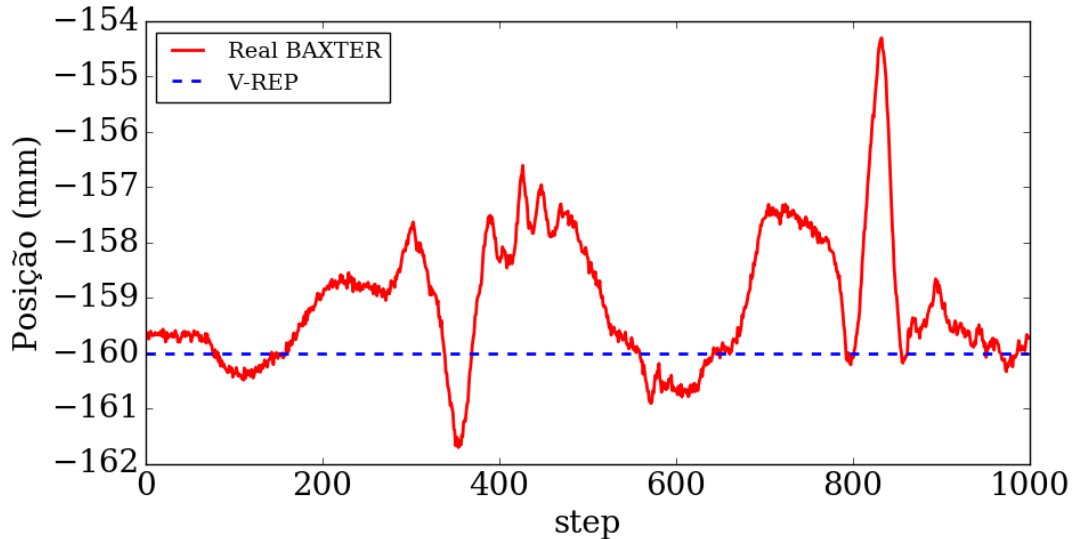


Figura 15: Distância no eixo Z-axis.

É importante notar que a diferença entre os picos de máximo e mínimo observada na Figura 15 é aproximadamente $7.5mm$, um valor maior que o observado ($6.0mm$) quando utilizando as redes de controle. Esta diferença é devido à trajetória estar sendo executada em uma altura maior, utilizando mais a junta S1, e assim aumentando a carga sobre as molas que sustentam o ombro do robô, gerando os picos observados. Assim resultando em um desafio maior para o sistema de controle de posicionamento aqui proposto.

3.3 Conjunto de treinamento

Para encontrar a função de ativação da rede é necessário um treinamento com dados que retomam o comportamento desejado. Para realizar o aprendizado do controlar aqui apresentado, optou-se trabalhar com dois controladores Proporcional Integrativo Derivativo (PID) em paralelo, realizando a fusão dos sensores. A escolha desse controlador se deve à alguns fatores: fácil implementação devido às ferramentas disponíveis nas bibliotecas do robô BAXTER, utilizando as variáveis de tempo internas; rapidez de implementação e teste; e por apresentar uma resposta suficientemente satisfatória para o controle em questão, que é manter a caneta de feltro em contato com a superfície durante toda a trajetória do desenho. Todos os dados apresentados aqui foram obtidos experimentalmente controlando o robô BAXTER.

O PID é um tipo controlador amplamente utilizado na redução de erros e resposta rápida em controle de processos. Os parâmetros do PID foram ajustados usando o método de Ziegler-Nichols (OGATA; YANG, 2010). Dois controladores PID foram

implementados para criar um conjunto de dados contendo 100 ensaios, cada um com 1000 pontos da trajetória quadrada.

Figura 16 mostra como os PIDs foram inseridos no sistema de controle: Um PID foi utilizado para manter o sensor de distância em um setpoint de 15.7cm . O segundo foi ajustado para manter a força em -2N . Um valor negativo, à medida que a força estimada aponta para o *endpoint* do robô. Os PIDs compensam o erro de deslocamento do posicionamento em Z , a saída somada dos dois controladores é compensado em cada junta como um valor de *offset*.

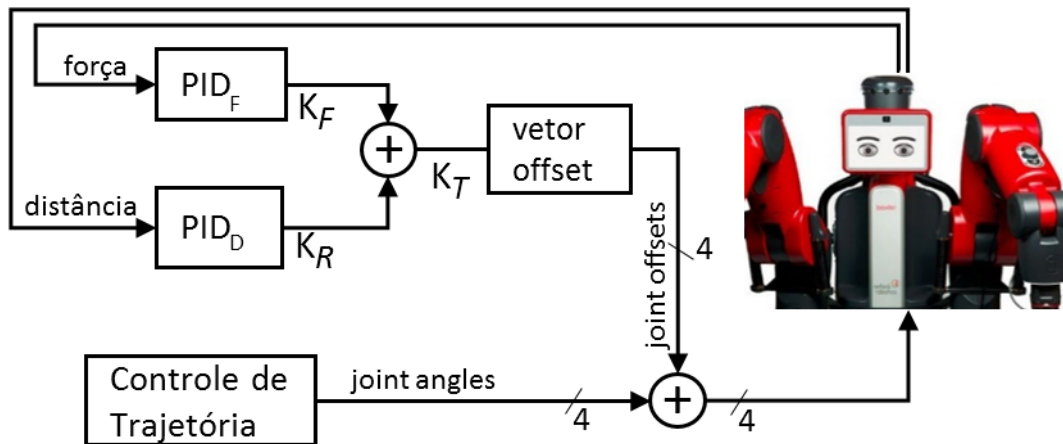


Figura 16: Diagrama do sistema utilizado para criar o conjunto de treinamento.

O conjunto de treinamento criado consiste em 100 ensaios, divididos em 50 ensaios em uma superfície à 0° , 25 ensaios com uma inclinação de aproximadamente 4° no eixo Y e 25 com a mesma inclinação no eixo X , a imagem 17 mostra, respectivamente, da esquerda para a direita, as disposições mencionadas. Todas trajetórias foram executadas usando uma caneta de feltro desenhando em uma superfície macia (um caderno de notas).

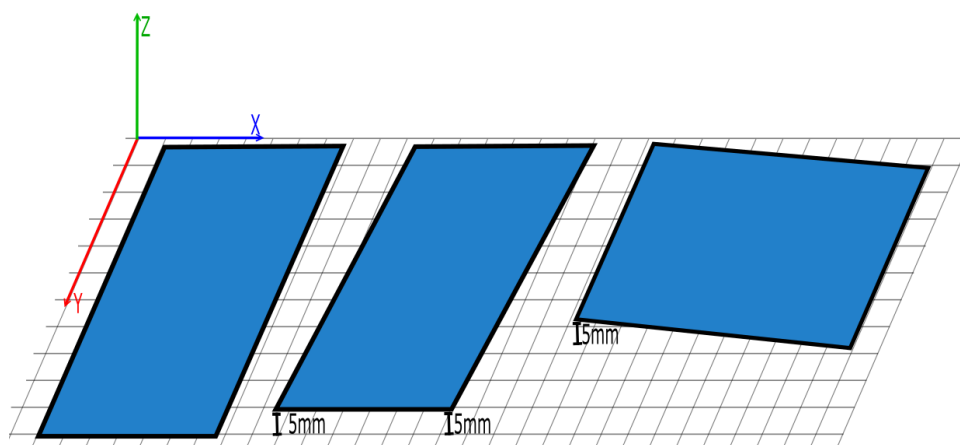


Figura 17: Disposições da superfície de desenho utilizadas.

A trajetória utilizada foi projetada no simulador V-REP 1.0cm acima do nível da superfície. Desse modo a ponta da caneta não tocava a superfície em nenhum ponto ao comandar o robô apenas com os ângulos das juntas. Ou seja, sem qualquer controle no eixo Z , não há contato entre a caneta e a superfície.

A Figura 18 mostra as médias dos valores de força (em N) obtidos em 100 ensaios, realizando a trajetória quadrada com os controladores PID. Os conjuntos PID - Set 1, 2 e 3 (linhas azul, vermelha e verde) representam os 50 ensaios em 0° , 25° em 4° no eixo X , e 25° com 4° no eixo Y , respectivamente. A linha preta pontilhada mostra o *setpoint* do controlador em $-2N$ e a linha preta contínua (*Without Control*) mostra a força média de 10 ensaios sem utilizar nenhum controle, a fim de comparação.

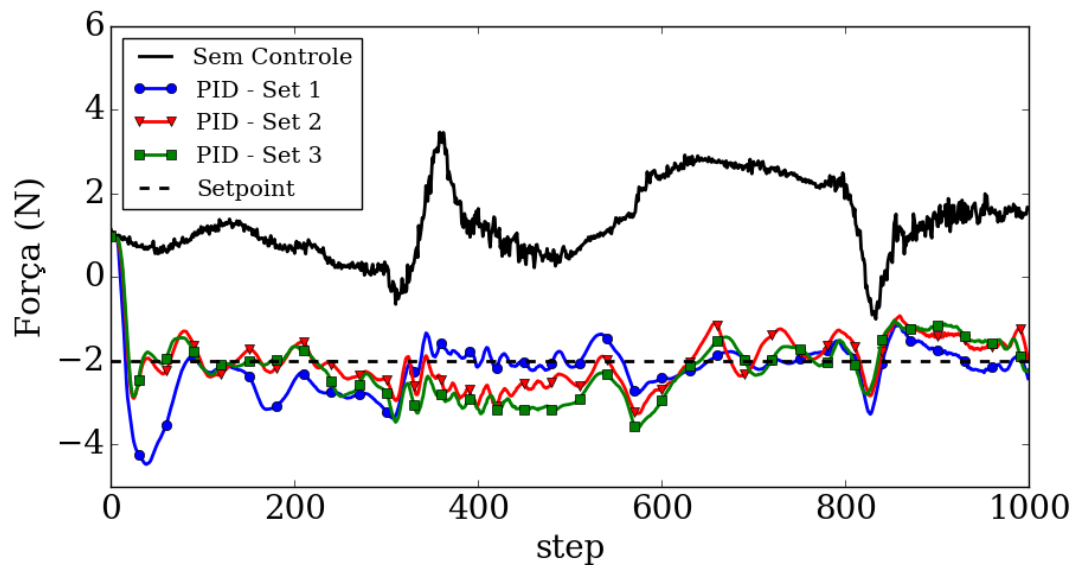


Figura 18: Valores médios de força do conjunto de treinamento.

A Figura 19 mostra as médias dos valores de distância (em cm) obtidos em 100 ensaios realizando a trajetória quadrada com os controladores PID, os conjuntos *PID - Set 1*, *2* e *3* (linhas azul, vermelha e verde) representam os 50 ensaios em 0° , 25° em 4° no eixo X , e 25° com 4° no eixo Y , respectivamente. A linha preta pontilhada mostra o *setpoint* do controlador em $15.7cm$ e a linha preta contínua (*Without Control*) mostra a força média de 10 ensaios sem utilizar nenhum controle, a fim de comparação.

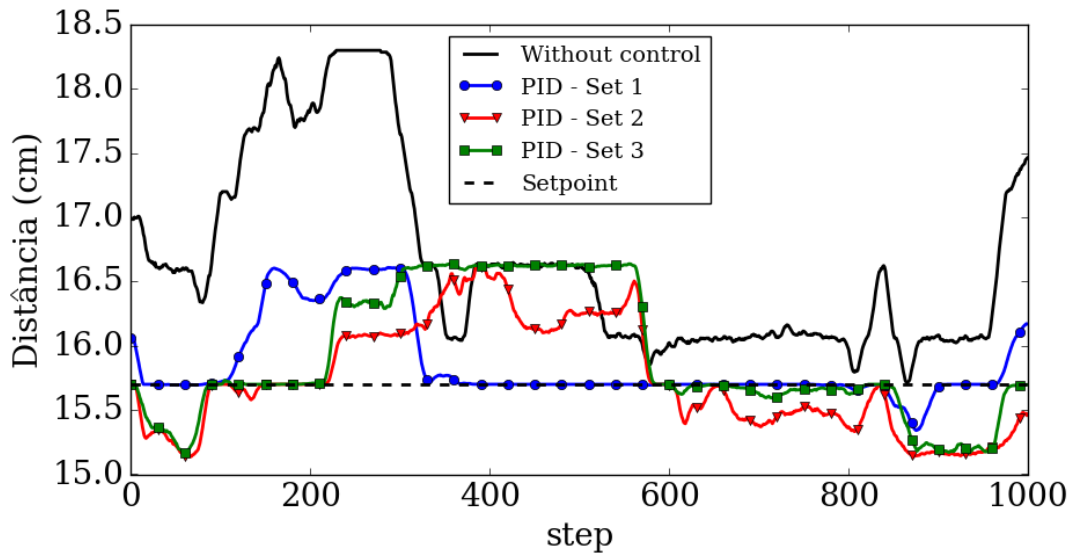


Figura 19: Valores médios de distância do conjunto de treinamento.

A Figura 20 mostra as médias dos valores de saída de controle final ($K_F + K_R$) obtidos em 100 ensaios realizando a trajetória quadrada com os controladores PID. Os conjuntos *Set 1*, *2* e *3* (linhas azul, vermelha e verde) representam os 50 ensaios em 0° , 25 em 4° no eixo X , e 25 com 4° no eixo Y , respectivamente.

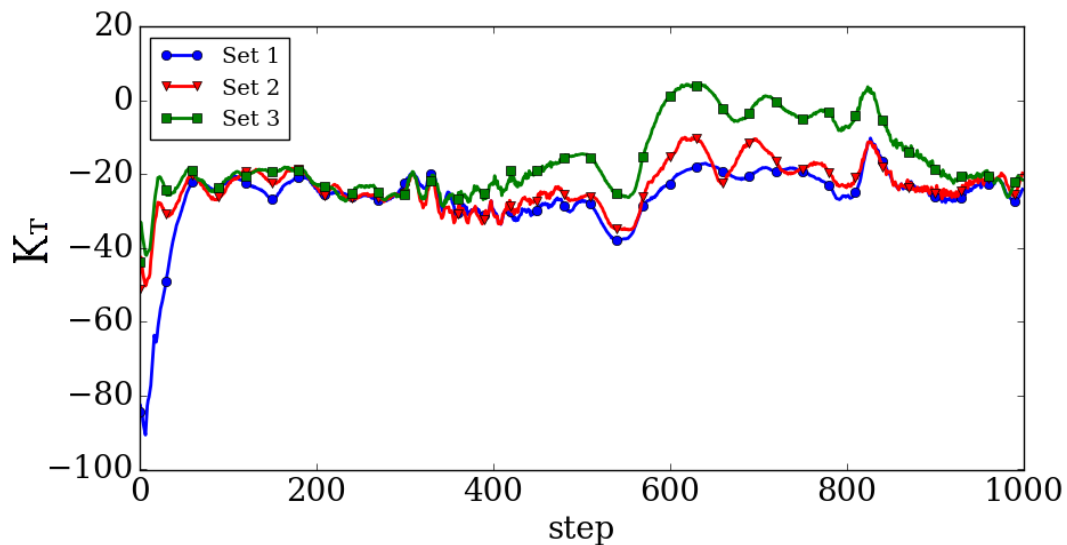


Figura 20: Valores médios da saída de controle dos PIDs do conjunto de treinamento.

3.4 Projeto e Simulação dos Líquidos

Cinco líquidos foram criados, cada um contendo 200 neurônios, 100 para cada entrada sensorial. O modelo neural e parâmetros da LSM empregados neste trabalho e no laço de controle principal (AZAMBUJA; CANGELOSI; ADAMS, 2016) basearam-se nos parâmetros apresentados em (JOSHI; MAASS, 2005). Com algumas modificações, não foram utilizadas implementações de Plasticidade de Curto Prazo (STP - *Short-Term Plasticity*) ou atrasos de transmissão forçados. Segundo

(BUONOMANO; MAASS, 2009), STP é apenas uma das propriedades geradoras de estados escondidos na rede. Como eles aumentam o tempo de simulação, ao exigir mais variáveis e cálculos, STP e atrasos de transmissão forçados não foram utilizados.

Os parâmetros utilizados foram: $c_m = 30nF$, $\tau_m = 30ms$, $\tau_{syn_i} = 6ms$, $\tau_{syn_e} = 3ms$. Cada líquido teve seu i_{offset} gerado aleatoriamente durante a sua criação, conforme descrito em (JOSHI; MAASS, 2005). A estrutura interna do líquido, ou seja, suas conexões, são mantidas após a inicialização. Dessa forma, cada execução da rede pode ser executada com o mesmo líquido. A potencial inicial da membrana e a corrente i_{noise} foram aleatoriamente gerados durante as fases de aprendizado e teste, onde novos valores para i_{noise} são gerados para cada passo de simulação. Para todas simulações, utilizou-se um passo de $2ms$.

Um código de população simplificado foi usado para gerar os pulsos de entrada, discretizando valores analógicos. O Algoritmo 1 apresenta um pseudo código demonstrando o procedimento de discretização da entrada. Como o número de neurônios é limitado, cada sinal de entrada foi condicionado entre $-10N$ a $2N$ e $15cm$ a $18cm$, para sensores de força e distância, respectivamente. Como cada líquido foi dividido igualmente para receber duas entradas, tem-se uma camada de entrada com 100 neurônios para cada sensor. O valor retornado pelos sensores são normalizados entre os limites estabelecidos e faz-se a equivalência, de maneira linear, do valor de entrada normalizado com o índice do vetor contendo 100 valores entre -1 e 1 , obtendo assim o índice dos neurônios que serão ativados. O pseudo-código que realiza essa normalização é apresentado no Algoritmo 1.

As conexões entre os neurônios do líquido e os pulsos de entrada são formatados para criar uma forma Gaussiana, cujo valor médio é o índice do neurônio de entrada, que receberá o pulso. Essa aplicação é exemplificada na Figura 21, onde a entrada é discretizada e normalizada para o correspondente índice do vetor de neurônios (pelo método exposto no Algoritmo 1). A distribuição Gaussiana, com valor máximo de $100nA$, é usada para enviar o sinal ao neurônio central e seus vizinhos.

Algoritmo 1: NORMALIZAÇÃO DA ENTRADA

```

Entrada:  $I_{sensor}$  /* Valor de entrada(Força ou Distância) */
Saída:  $index$  /* Índice para a entrada gaussiana */
1  $gain = I_{max} - I_{min}$  /* Limites do sensor */
2  $mid = (I_{max} + I_{min})/2$  /* Valor inicial em  $t_0$  */
3  $V = [-1, \dots, 0, \dots, 1]$  /* Vetor normalizado de tamanho NdN */
4 /* NdN = Número de Neurônios */
5 Função NormInput( $I_{sensor}$ )
6 | Recebe valor do sensor  $I_{sensor}$ 
7 | Subtrai  $mid$  /* A curva é deslocada para o centro */
8 | Divide por  $gain$  /* Agora o valor de  $I_{sensor}$  é unitário */
9 | Discretiza  $I_{sensor}$  /* Valor mais próximo do vetor normalizado */
10 retorna  $index$  /* índice que mais aproxima  $V[index] \approx I_{sensor}$  */

```

A quantidade de neurônios foi determinada a partir da resolução de entrada, que pode ser calculada em a partir dos limites impostos nos sensores, de modo que a

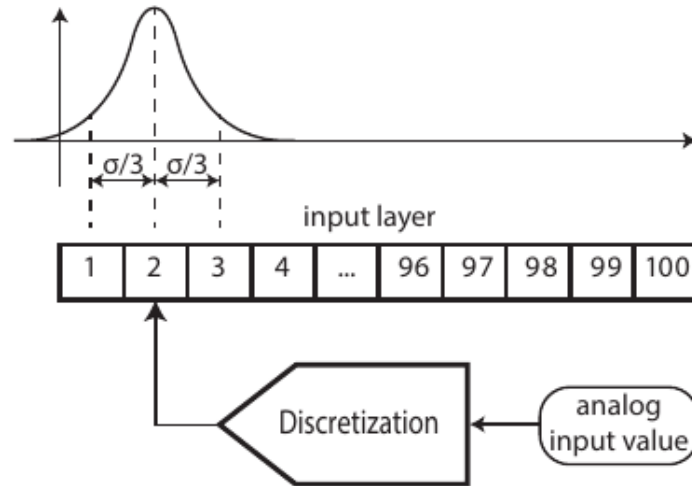


Figura 21: Normalização da entrada para uma entrada gaussiana. Fonte: Adaptado de (AZAMBUJA; CANGELOSI; ADAMS, 2016)

resolução dos pulsos de entrada

$$res_{Input} = \frac{(Input_{max} - Input_{min})}{NdN} \quad (6)$$

onde $Input_{max}$ e $Input_{min}$ são os valores máximos e mínimos de entrada e NdN é o número de neurônios destinado àquela entrada. A resolução obtida para a força foi de $res_{Force} = 0,12 \frac{N}{neuron}$, e $res_{Distance} = 0,03 \frac{cm}{neuron}$ para a distância. Esta sendo a mesma resolução do sensor de distância, já o valor de força, estimado pelo modelo interno do robô, não possui uma resolução dada por um valor exato.

Realizou-se uma análise da normalização dos dados de entrada. A Figura 22 mostra os valores de 30 amostras de um sinal de força (curva *Original*), dentro do limite imposto ($-10N$ à $2N$), deslocado e escalado para ter seus valores entre 0 e 100. Os pontos azuis mostram o valor do índice neural quando aplicado a função $NormInput(I)$ do Algoritmo 1 no sinal de força. O erro médio quadrático (MSE - *Mean Square Error*) encontrado entre as curvas foi de 0.101.

Na Figura 23, mostra-se os valores de 60 amostras de um sinal de distância (curva *Original*), dentro do limite imposto de $15cm$ à $18cm$, deslocado e escalado para ter seus valores entre 0 e 100. Os pontos azuis mostram o valor do índice neural quando aplicado a função $NormInput(I)$ do Algoritmo 1 no sinal de distância. Como ambos possuem a mesma resolução, as curvas se sobrepõem, o erro MSE calculado foi 0.

Os líquidos foram simulados, recebendo como entrada os sinais discretizados do conjunto de treinamento, contendo os dados dos ensaios dos PIDs apresentados na seção 3.3. Isto produz uma matriz de tamanho 1000×200 para cada elemento do conjunto de dados (100). As Figuras 24 e 25 apresentam a atividade neural de duas das 100 simulações realizadas, a primeira teve como dados de entrada dados conjunto de treinamento *PID - Set 1*, na segunda os dados do *PID - Set 2*. Além da atividade neural aleatória devido ao ruído natural da rede, nota-se padrões dos sinais ao longo do tempo correspondentes aos estímulos das entradas.

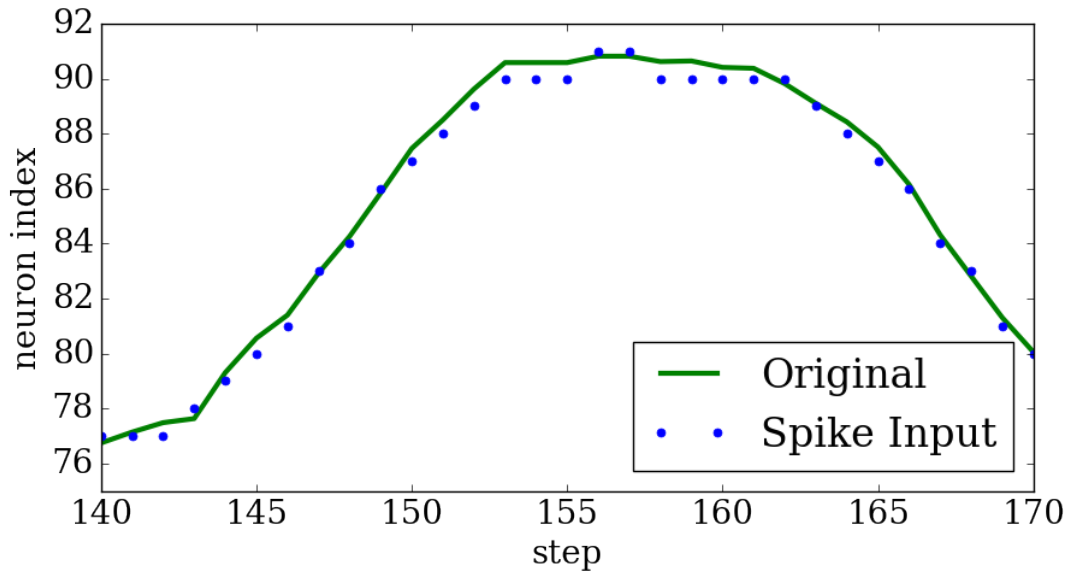


Figura 22: Normalização do sinal de força para um índice neural.

3.5 Criação do controlador

A partir dos líquidos simulados, utilizando como entrada os dados do conjunto de treinamento, tem-se 100 simulações de 1000 passos com 200 neurônios, para cada um dos 5 líquidos criados. O próximo passo é realizar o aprendizado da rede, obtendo os pesos da função de ativação. Utilizou-se o método de regressão linear apresentado na Seção 2.3.3, equação 5.

No treinamento, realizado individualmente para cada líquido, a matriz \mathbf{X} possui todos os valores da atividade da rede e no vetor y tem-se os valores de saída de controle (K_T) dos controladores PID. A partir desses dados, encontra-se o vetor de pesos w para a função de ativação da rede. Vetor este que minimiza o erro na função $\min_w \|\mathbf{X}w - y\|_2^2 + \alpha \|w\|_2^2$. Vale lembrar que existem diversas metodologias para encontrar os pesos da função de ativação, neste trabalho utilizou-se a regressão linear de Ridge.

A criação de diversos líquidos é justificada pela aleatoriedade de suas conexões, podendo não apresentar dinâmica interna suficiente para encontrar uma função de ativação que vá de encontro ao resultado esperado. Uma breve avaliação da etapa de aprendizado pode ser feita a partir dos dados simulados do conjunto de treinamento e função de ativação. Para cada líquido, obtém-se a saída de controle K_{LSM} de cada simulação realizada, multiplicando os dados utilizados em X pelo vetor w . Observa-se os valores médios de saídas de controle e o desvio dos resultados encontrados para cada rede e compara-se com a saída de controle K_T utilizados no treinamento. As redes que não apresentarem um bom treinamento são descartadas.

De forma análoga ao diagrama da implementação dos controladores PID, apresentado anteriormente na Figura 16, para os experimentos utilizando o robô BAXTER e os controladores LSM foi utilizada a estrutura apresentada na Figura 26, onde o controlador LSM recebe os valores dos sensores de distância e força e corrige o erro de posicionamento do controlador de trajetória. Um único bloco foi alterado em todo o sistema, agora o controle é realizado pelo controlador LSM, que recebe como entrada os dados de força e distância.

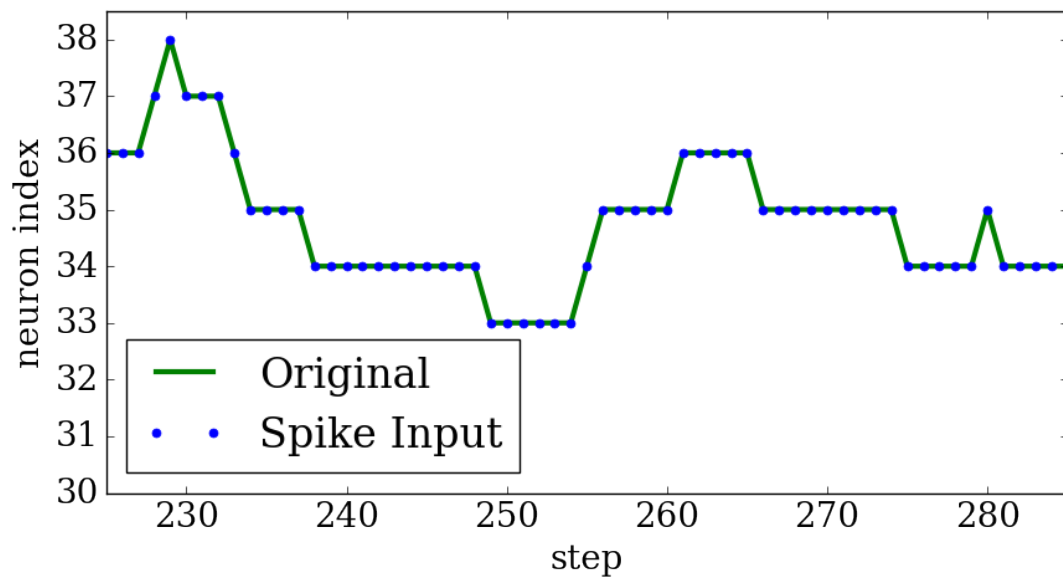


Figura 23: Normalização do sinal de distância para um índice neural.

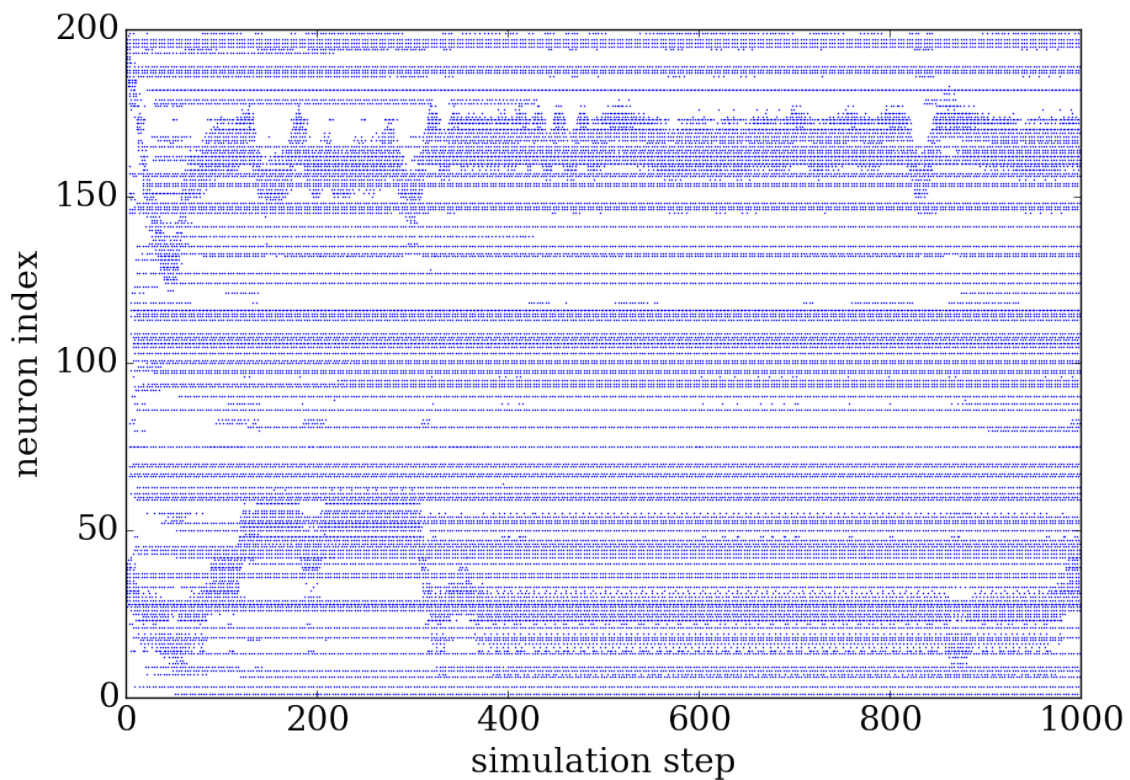


Figura 24: Atividade neural no líquido durante simulação utilizando dados do conjunto de treinamento *PID - Set 1* como entrada.

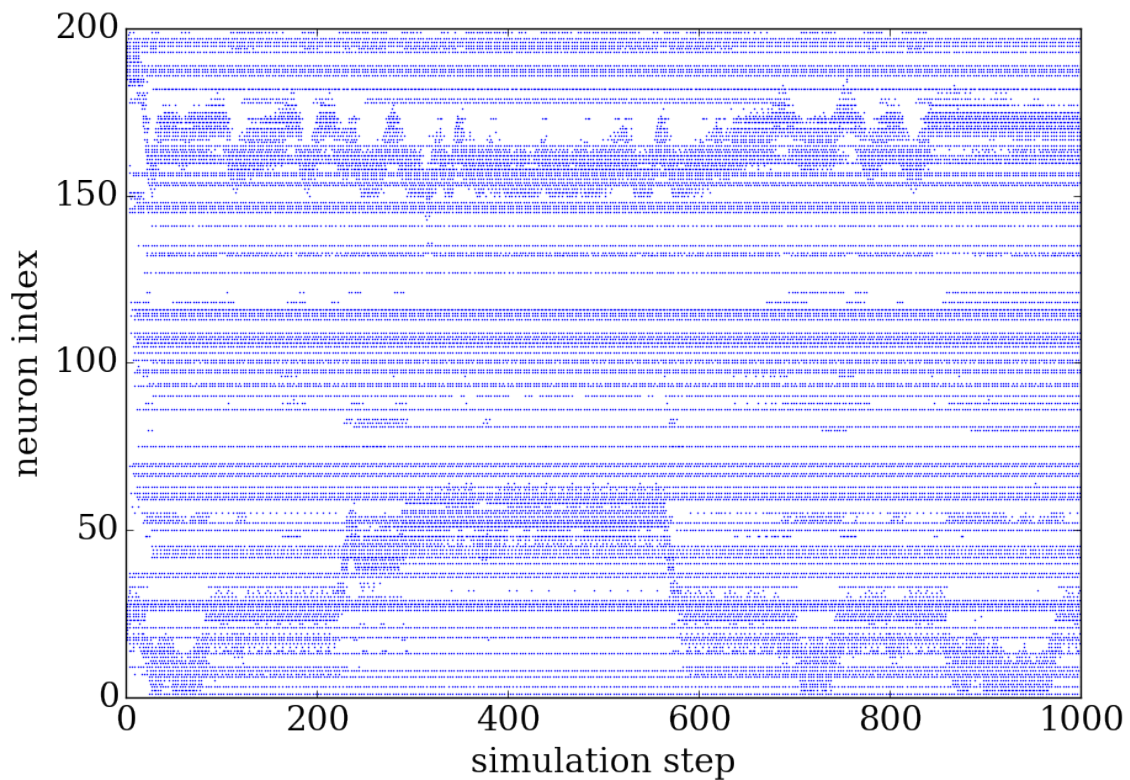


Figura 25: Atividade neural no líquido durante simulação utilizando dados do conjunto de treinamento *PID - Set 2* como entrada.

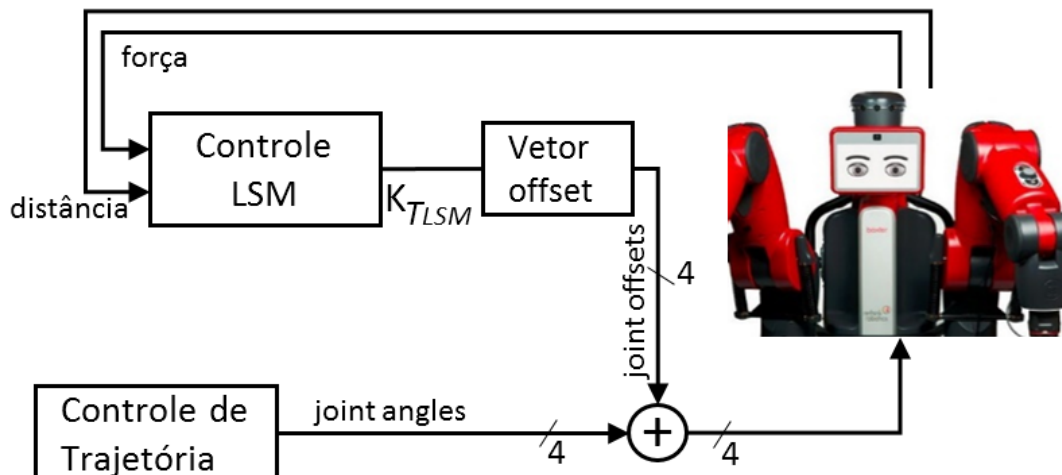


Figura 26: Diagrama do sistema para realizar os experimentos.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos com o controlador LSM implementado, conforme apresentado na seção 3.5, treinado com dados de dois controladores PID paralelos, apresentado na seção 3.3. Primeiramente são detalhados os experimentos, e a forma como eles foram divididos. Depois são apresentados os resultados iniciais e uma breve avaliação de todas as redes treinadas, e na sequência são exibidos os resultados finais das redes selecionadas. Os experimentos aqui apresentados foram executados no robô BAXTER, durante um intercâmbio de pesquisa de dois meses, na *Plymouth University, Centre for Robotics and Neural Systems*, em conjunto com os autores do trabalho em que este estudo é baseado.

O controlador LSM deve minimizar a variabilidade no *endpoint* do robô, mantendo a força aplicada na superfície de desenho constante, reduzindo o erro de posição da caneta de feltro. O controlador LSM foi treinado com dados de um controlador PID. Espera-se que o controlador LSM aprenda o comportamento do PID, reduzindo o erro de posição Z, mantendo valores de força e distância em um *setpoint* fixo, e naturalmente mantendo o padrão da informação de entrada na forma de spikes, conforme já relatado nesse trabalho.

O conjunto de treinamento, apresentado na seção 3.3, foi criado utilizando uma superfície macia (caderno de notas) em três posições distintas (ver Figura 17). A partir destes dados, treinou-se o controlador para posição no eixo Z. Os experimentos utilizando o *framework* LSM desenvolvido foram divididos conforme mostra a Tabela 1. A execução foi feita em oito etapas, seguindo a ordem (Quantidade-Experimento): 50-A; 25-B; 25-C; 25-D; 50-A; 25-B; 25-C; 25-D. A primeira etapa foi realizada utilizando as cinco redes treinadas, realizou-se uma análise preliminar dos resultados, as três redes com os melhores resultados foram utilizadas na execução de todas as etapas restantes. No Experimento A utilizou-se a mesma superfície que foi utilizada com os controladores PID para gerar o conjunto de treinamento, sem inclinação. Os Experimentos B, C e D foram realizados com uma superfície diferente, um quadro branco (ver Figura 1), neste último, o a superfície foi disposta com o ângulo no lado oposto ao conjunto de treinamento.

Os resultados apresentados em (AZAMBUJA; CANGELOSI; ADAMS, 2016) mostraram uma diferença de $1,93mm$ entre as leituras de distância mínima e máxima do eixo Z, ao executar a trajetória quadrada no simulador V-REP. No entanto, realizou-se experimentos utilizando uma implementação das mesmas redes, porém no robô real, os resultados de posição no eixo Z apresentaram flutuações três vezes maiores. A Figura 27 exibe os resultados obtidos em 25 experimentos, com uma implementação idêntica às redes apresentadas por (AZAMBUJA; CANGELOSI; ADAMS, 2016), utilizando o robô BAXTER. Aqui, o valor médio entre os

Tabela 1: Definição dos experimentos.

Nome do experimento	Tipo da superfície	Posição da superfície	Quantidade de experimentos
Experimento A	Macia	0°	100
Experimento B	Dura	0°	50
Experimento C	Dura	4° no eixo Y	50
Experimento D	Dura	-4° no eixo X	50

picos de mínimo e máximo nas variações de posição foi de $6,36mm$. A rede utilizada foi uma duplicação do sistema de controle de trajetória com LSMs paralelas, feita a partir das mesmas trajetórias apresentadas no trabalho citado e utilizando os mesmos parâmetros e sementes para criação das redes, tendo assim a mesma estrutura interna. Essa rede também foi testada no simulador V-REP, onde verificou-se resultados semelhantes à rede original.

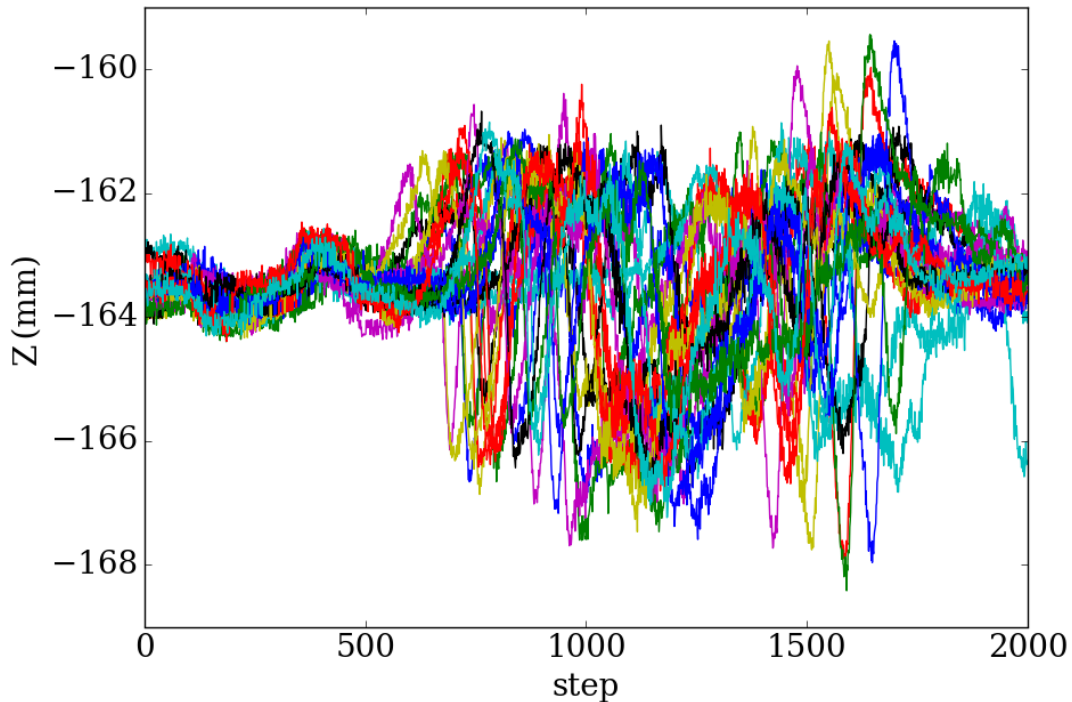


Figura 27: Posição no eixo Z utilizando a LSM de controle de trajetória.

A trajetória usada para validar o sistema apresentou uma diferença de aproximadamente $7,6mm$ (ver Fig. 15). As LSMs apresentadas foram avaliadas no robô real, executando a mesma trajetória quadrada em diferentes situações, alterando parâmetros como tipo de superfície e inclinação. Inicialmente todas as cinco LSMs foram submetidas a 50 experimentos em uma superfície macia (caderno), a mesma utilizada para criação do conjunto de treinamento. A partir de uma breve análise dos resultados desses experimentos, foram selecionados três líquidos para realizar o restante dos experimentos. Esta decisão foi tomada para tornar possível a realização de mais experimentos no mesmo espaço de tempo, utilizando as melhores redes, para uma avaliação mais rigorosa.

4.0.1 Avaliação inicial

Após a etapa de treinamento, realizou-se uma validação inicial das redes de controle de posição Z criadas, utilizando os valores dos pulsos filtrados pela membrana passa-baixa do líquido simulado com os dados do conjunto de treinamento. O resultado é usado para avaliar se a função encontrada pela regressão linear se encaixa no modelo esperado. Na figura 28 mostra-se os valores médios de saída de controle de 100 experimentos do controlador PID (presentes no conjunto de treinamento) e os valores médios para cada uma das cinco LSMs treinadas, quando submetidas às entradas do conjunto de treinamento, também 100 simulações para cada rede. A partir de uma breve inspeção visual, nota-se que as redes, apesar de possuírem conexões distintas, e portanto, funções de ativação também distintas, apresentaram resultados semelhantes durante o teste com o conjunto de treinamento.

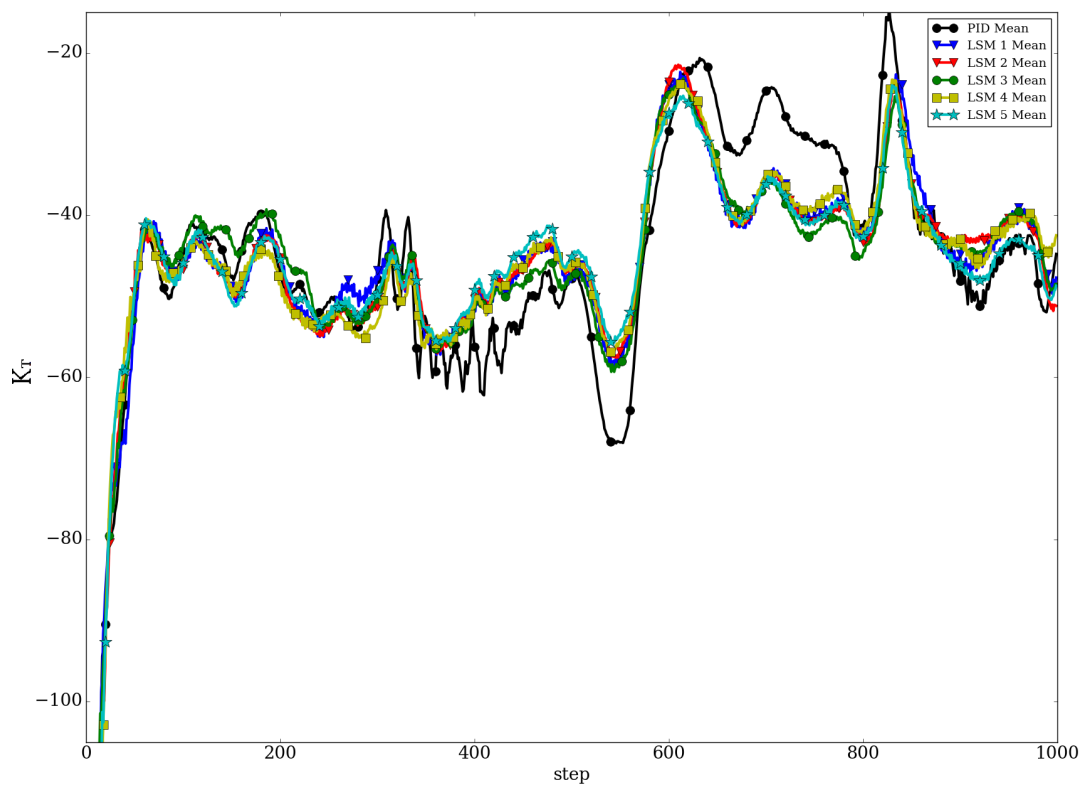


Figura 28: Teste inicial das LSMs, utilizando dados do conjunto de treinamento.

Conforme a tabela de experimentos (Tabela 1), a primeira etapa foi a realização da mesma trajetória, 50 vezes, sobre uma superfície macia, a mesma disposição utilizada no conjunto de treinamento. As cinco redes foram avaliadas e os resultados da força são apresentados na Figura 29, onde tem-se os valores médios de força dos 50 experimentos para cada rede, bem como o valor de $setpoint(-2N)$ utilizado no conjunto de treinamento. Valores acima de $0N$, são medidos quando a caneta não está encostando na superfície, já valores abaixo de, aproximadamente, $-4N$ começam a exercer muita força sobre a superfície. Nota-se, nesta primeira etapa, que as LSMs 1 e 2 exerceram muita força em parte da trajetória, a LSM 3 manteve-se próxima ao $setpoint$, enquanto as redes LSM 4 e 5 mantiveram-se acima do $setpoint$, porém abaixo de $0N$ na maior parte do trajeto. A partir deste experimento preliminar, selecionou-se as redes de número 3, 4 e 5 para as posteriores etapas.

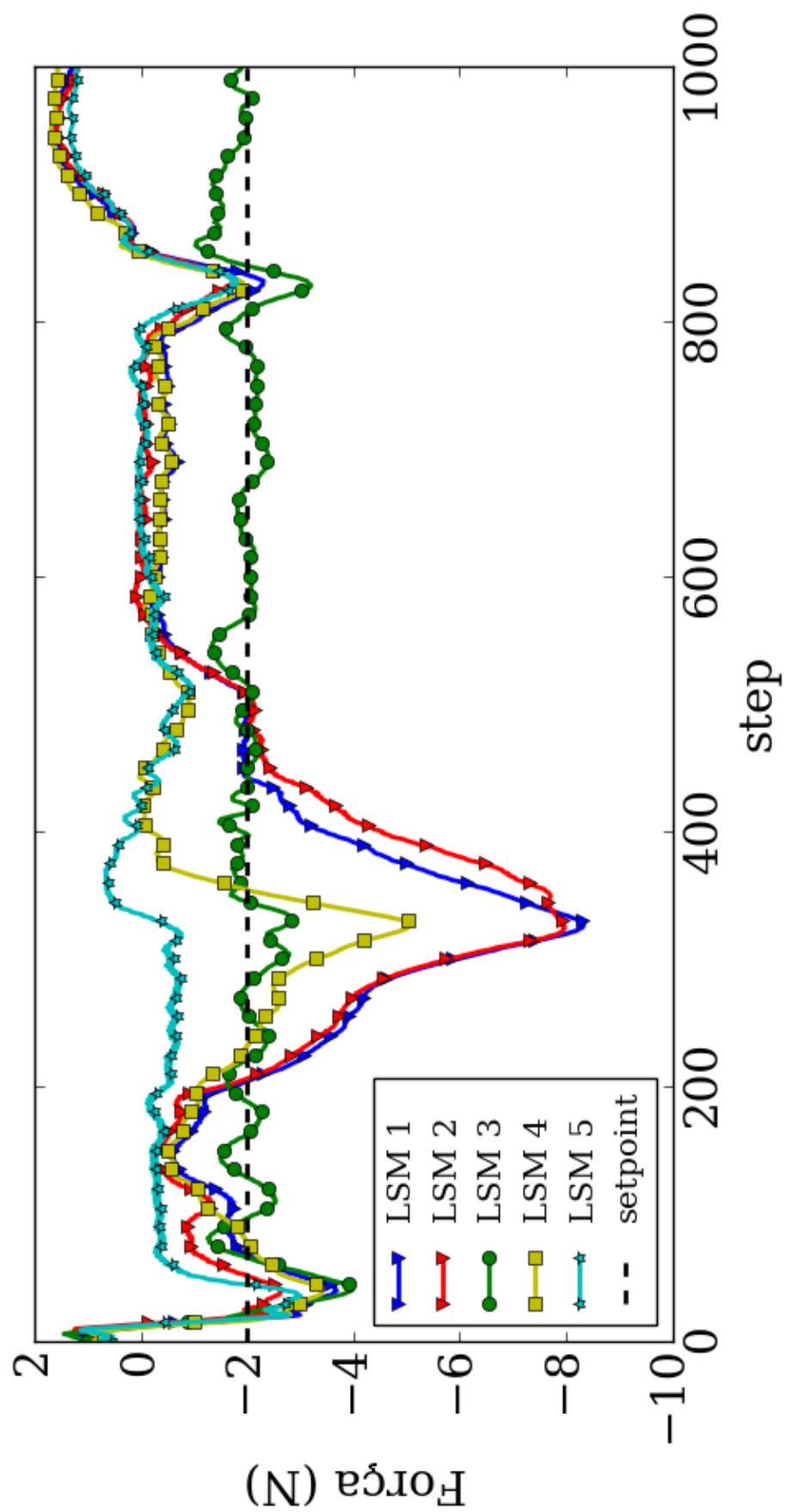


Figura 29: Força resultante das LSMs no conjunto de experimentos A.

4.0.2 Dados experimentais

Realizou-se os restantes dos experimentos A, B, C e D com as Liquid State Machines de números 3, 4 e 5. As médias dos resultados de força e distância, obtidos com cada LSM, separados por tipo de experimentos são mostrados nas Figuras 30, 31, 32 e 33, para os experimentos A, B, C e D, respectivamente.

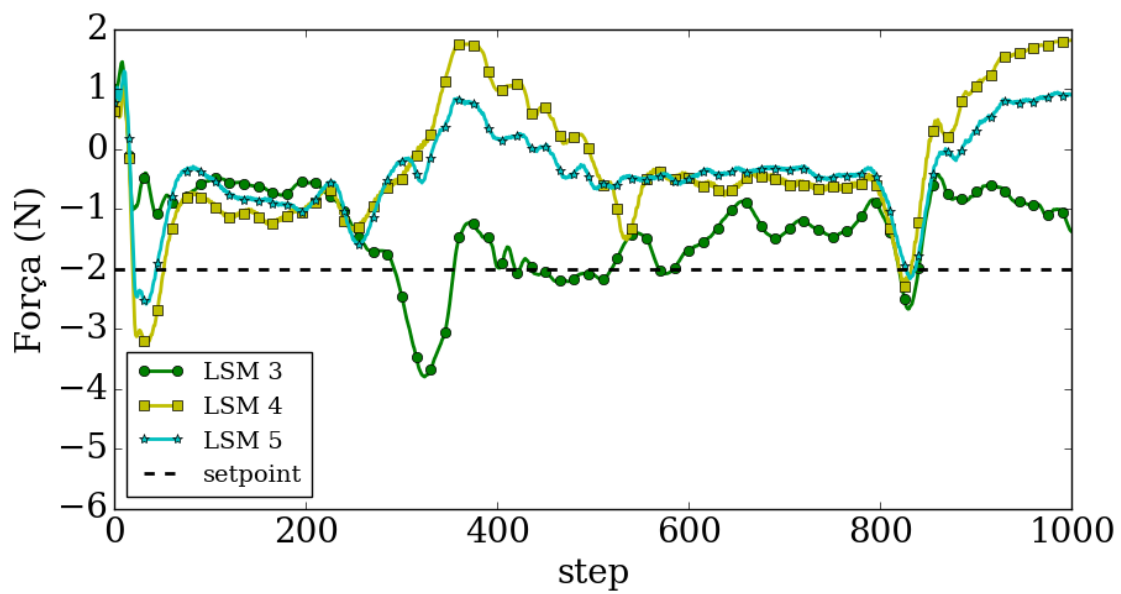
Na figura 30(a) apresenta-se os resultados de força média das LSM 3, 4 e 5 no experimento A. Onde utilizou-se a mesma superfície que no conjunto de treinamento, sem inclinação. Observa-se que os controladores LSM 4 e 5 apresentaram resultados semelhantes, porém mais distantes do *setpoint*, na maior parte da trajetória, em comparação com a LSM 3. Valores acima de $0N$ não são desejados, pois apontam momentos em que a caneta de feltro não estava em contato com a superfície. Os resultados de distância média para este experimento são apresentados na figura 30(a), onde resultados similares entre todas as redes são observados.

A figura 31(a) apresenta os resultados médios de força das LSM 3, 4 e 5 no experimento B. Onde utilizou-se uma superfície diferente (rígida) do conjunto de treinamento, sem inclinação. Observa-se, novamente, os controladores LSM 4 e 5 ficaram distantes do *setpoint*, na maior parte da trajetória, enquanto o LSM 3 manteve-se próximo ao *setpoint* durante toda trajetória, aponta-se também para o valor mínimo da LSM 4, próximo à *step* = 350, onde o valor de força abaixo de $-4N$ apresenta força excessiva utilizada na superfície. Outro ponto importante a ser apontado neste gráfico são os altos valores de força média para as LSM 4 e 5 no final da trajetória, mantendo a caneta de feltro sem contato com a superfície. Os resultados médios de distância para este experimento são apresentados na figura 31(a), onde resultados similares entre as redes LSM 4 e LSM 5 são observados, e a LSM 3 manteve valores menores.

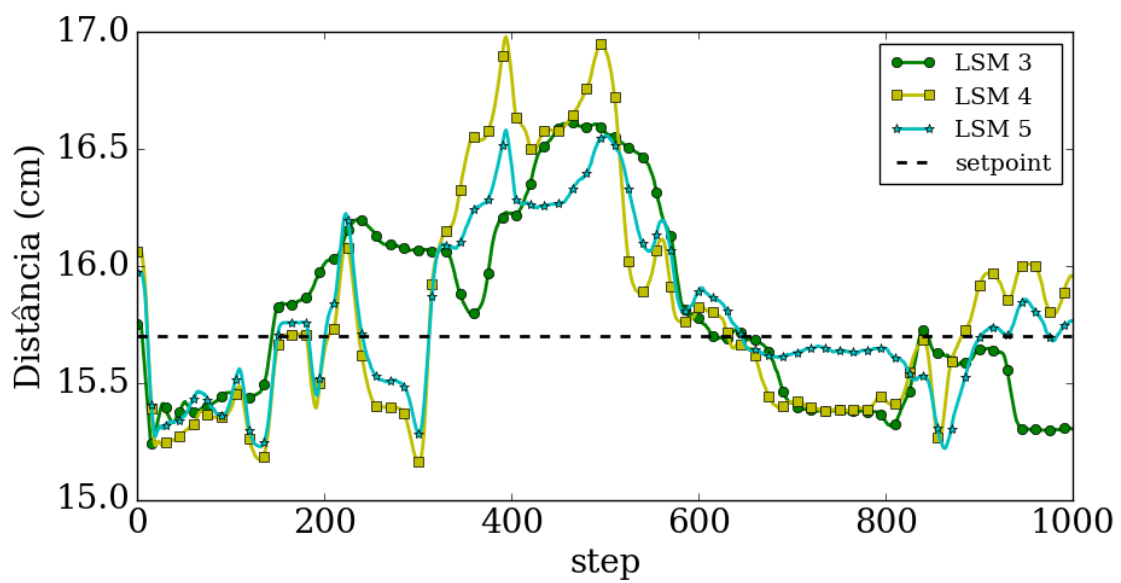
Os resultados de força média das LSM 3, 4 e 5 obtidos no experimento C são apresentados na figura 32(a). Utilizou-se a superfície rígida, com inclinação de 4° no eixo Y, inclinação também presente no conjunto de treinamento. Observou-se um comportamento distinto dos experimentos anteriores, onde a LSM 3 manteve, na média, um valor constante de força, porém com *bias* negativo (próximo de $-3,88N$), resultando em uma maior força aplicada sobre a superfície. A LSM 4 apresentou, na média, forças altas durante o início, mas foi capaz de alcançar o *setpoint* após certo tempo (*step* \approx 400). Os resultados de distância média para o Experimento C são apresentados na figura 32(a), onde a LSM 3 manteve valores menores de distância média, e resultados similares entre as redes LSM 4 e LSM 5 são observados.

Na figura 33(a) apresenta-se os resultados médios de força das LSM 3, 4 e 5, para o experimento D. Onde foi utilizado a superfície rígida, com inclinação de -4° no eixo X, inclinação no sentido oposto ao utilizado no conjunto de treinamento. É importante notar que neste experimento, a inclinação é do mesmo lado que o início da trajetória, ou seja, a caneta está mais próxima à superfície no início, se comparado com os outros experimentos. Observa-se um comportamento similar entre as redes LSM 3 e 4 no início da trajetória, aplicando uma força menor no início, já a rede LSM 5 aplica muita força no começo da trajetória. A situação entre a LSM 4 e LSM 5 se invertem após *step* \approx 200, onde a LSM 5 começa a aplicar força exagerada e a LSM 4 estaliza-se próxima ao *setpoint*. A LSM 3 permanece próxima ao *setpoint* durante quase toda trajetória. Os resultados de distância média para o Experimento D são apresentados na figura 32(a), onde as 3 redes apresentam resultados similares, já que estas, na média, mantiverem o contato entre a caneta e superfície (de acordo

com o gráfico de força média).

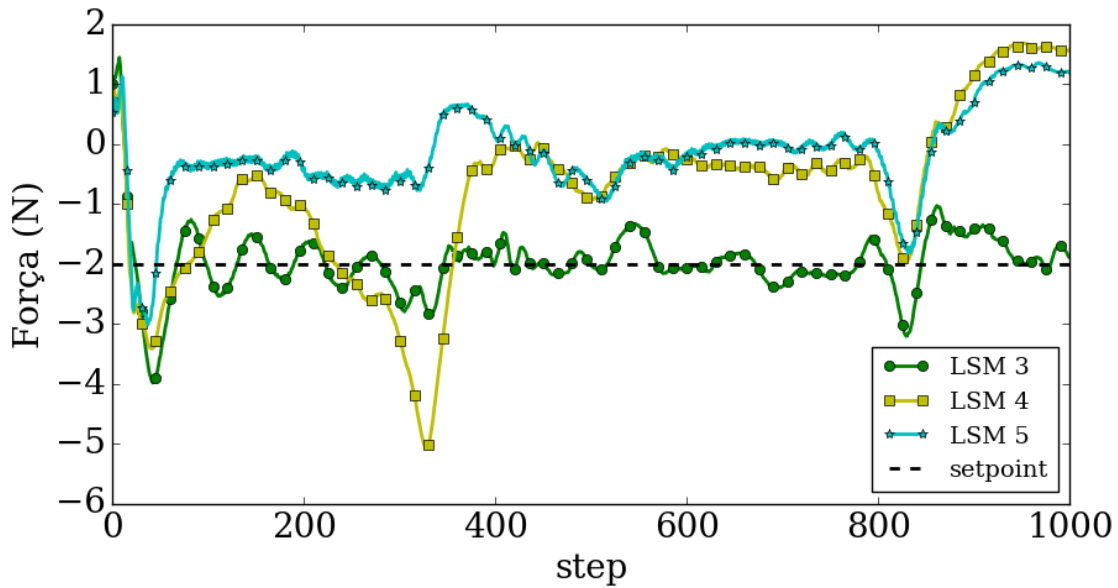


(a)

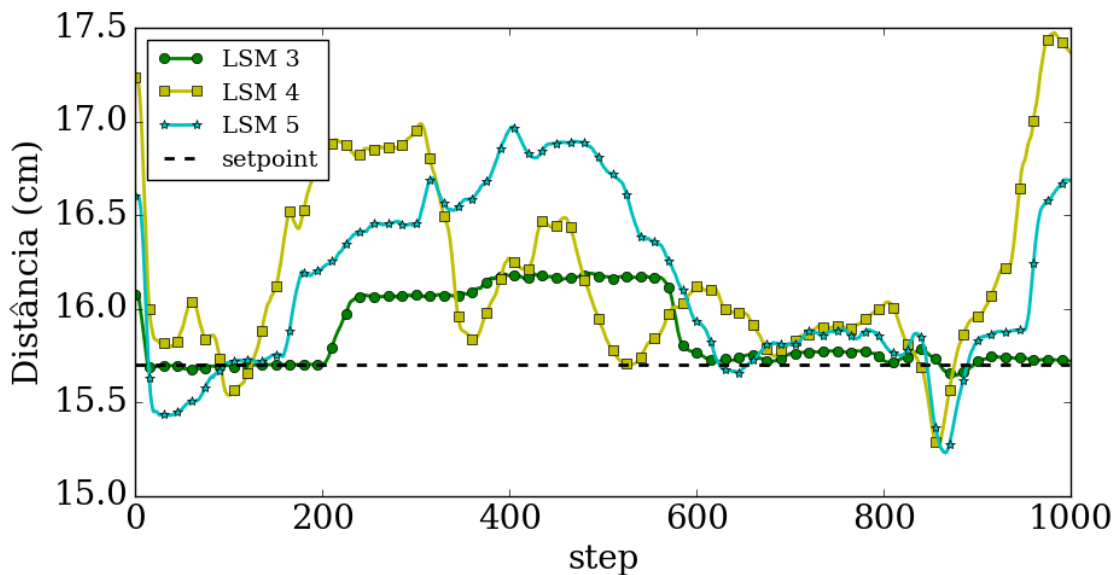


(b)

Figura 30: Resultados obtidos no Experimento A.



(a)



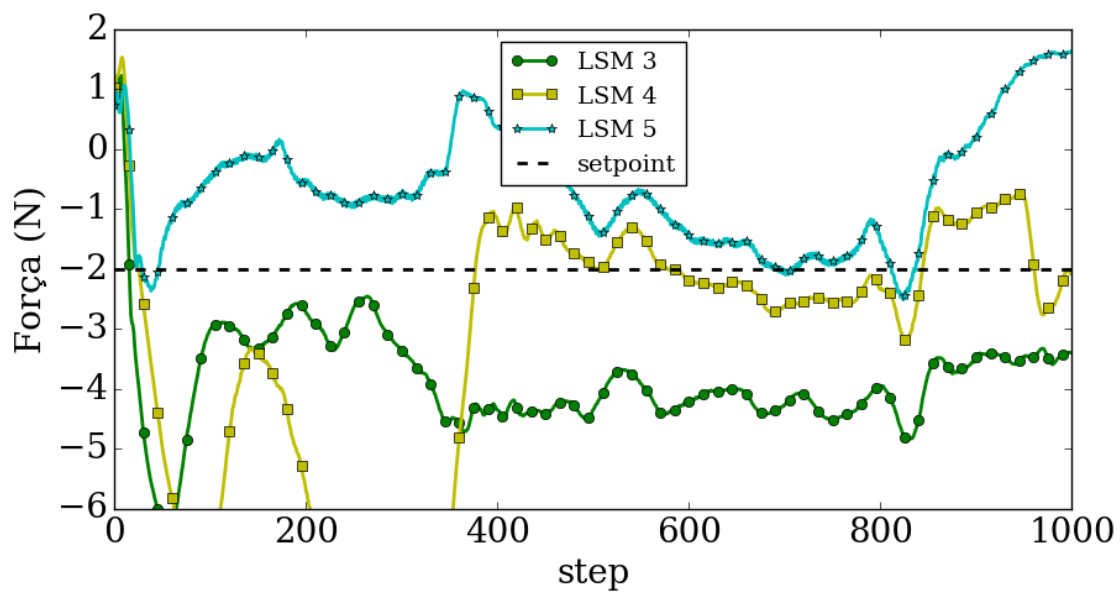
(b)

Figura 31: Resultados obtidos no Experimento B.

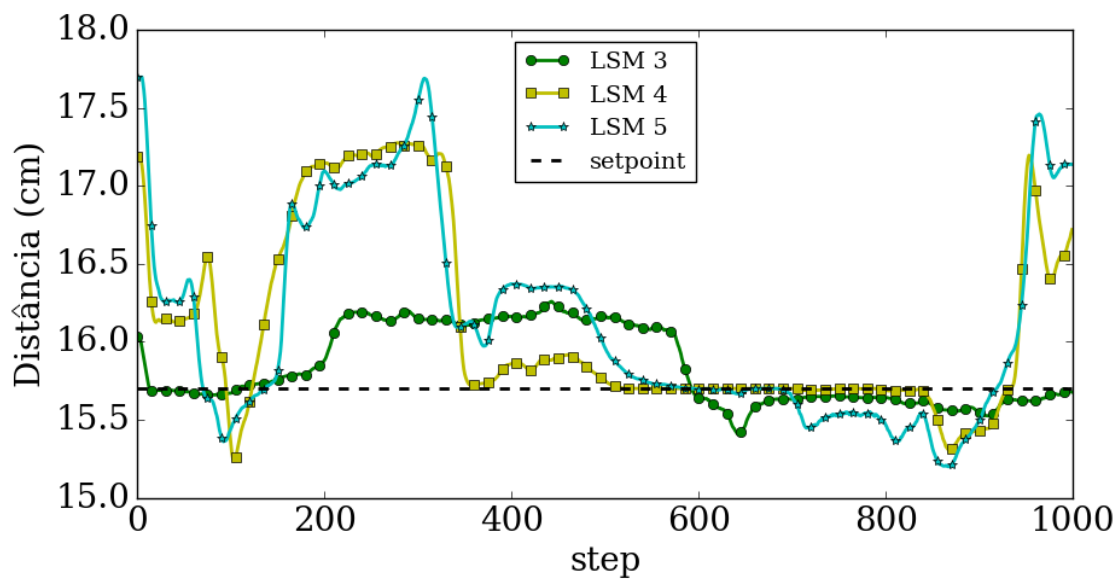
Os valores combinados, para força e distância, de todos os experimentos são apresentados na Figura 34. Na média geral, a LSM 3 manteve-se mais próxima do *setpoint* de $-2N$ ao longo da trajetória, como pode ser observado no gráfico da força média exposto na Figura 34(a). Na Figura 34(b) são mostrados os valores média de distância para cada rede, os resultados mostram-se similares entre as três redes. Uma diferença no final da trajetória ($step > 800$) é observado para as redes LSM 4 e LSM 5, que apresentaram um valor maior que a LSM 3, provocando falhas na trajetória, onde a caneta de feltro não mantém contato com a superfície.

De modo geral, a LSM 3 apresentou melhores resultados em cada um dos quatro experimentos realizados. Manteve-se mais próxima do *setpoint* de força nos gráficos das Figuras 30(a), 31(a) e 33(a). Demonstrou um *bias* negativo no experimento C (Figura 32(a)), significando que na maioria dos experimentos deste conjunto, foi

exercida uma maior força sobre a superfície. Calculou-se os valores de Raiz do Erro Quadrático Médio (RMSE - *Root-Mean-Square Error*), erro das curvas de todos os experimentos em relação aos *setpoints*. Os resultados RMSE para distância e força são apresentados na Tabela 2. A LSM 3 apresentou o menor valor RMSE em ambos os casos.

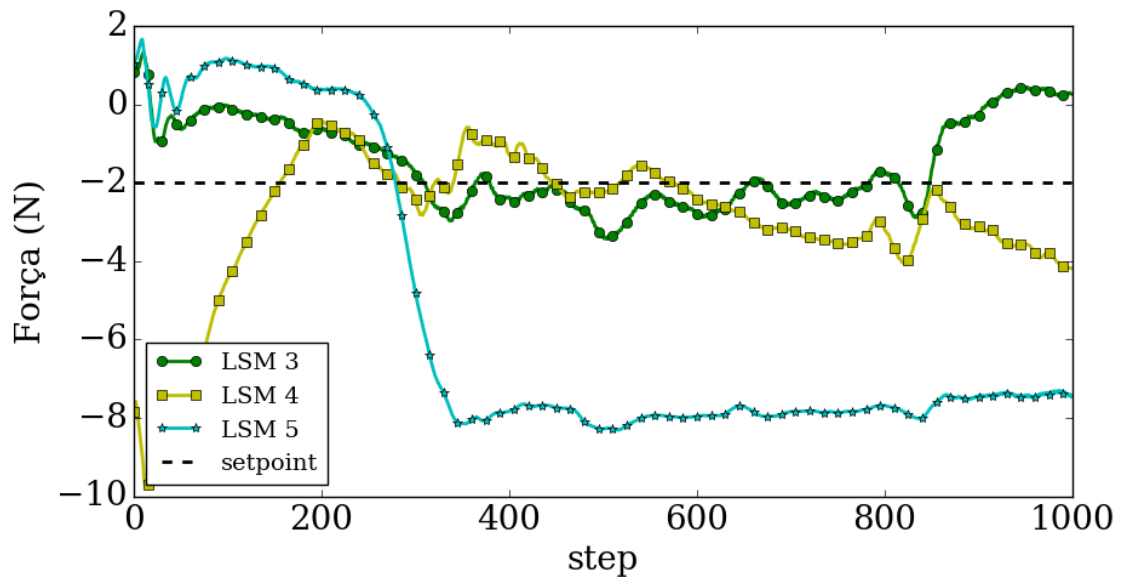


(a)

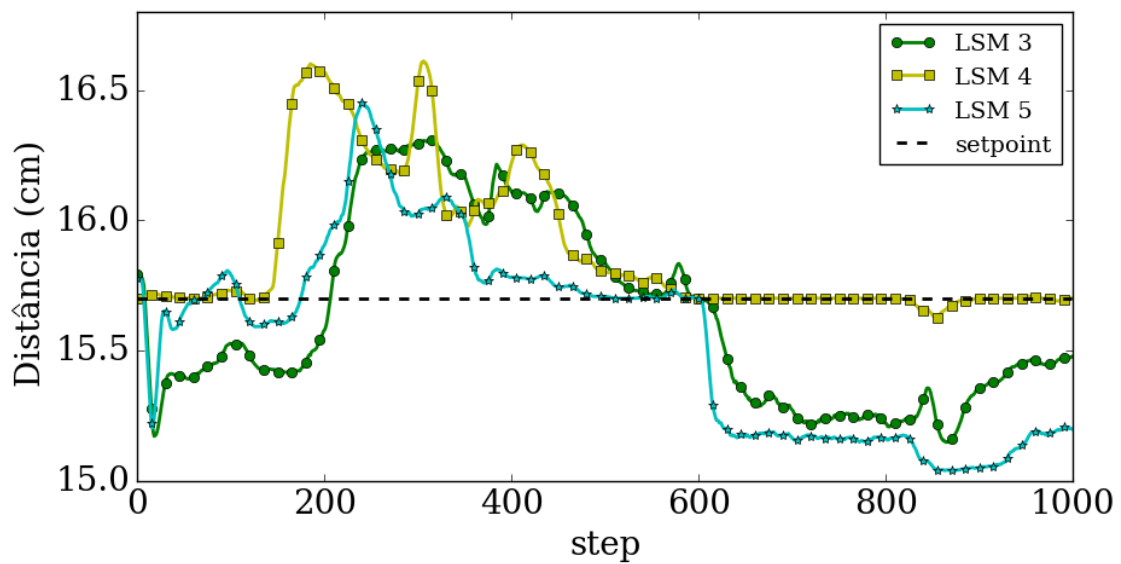


(b)

Figura 32: Resultados obtidos no Experimento C.

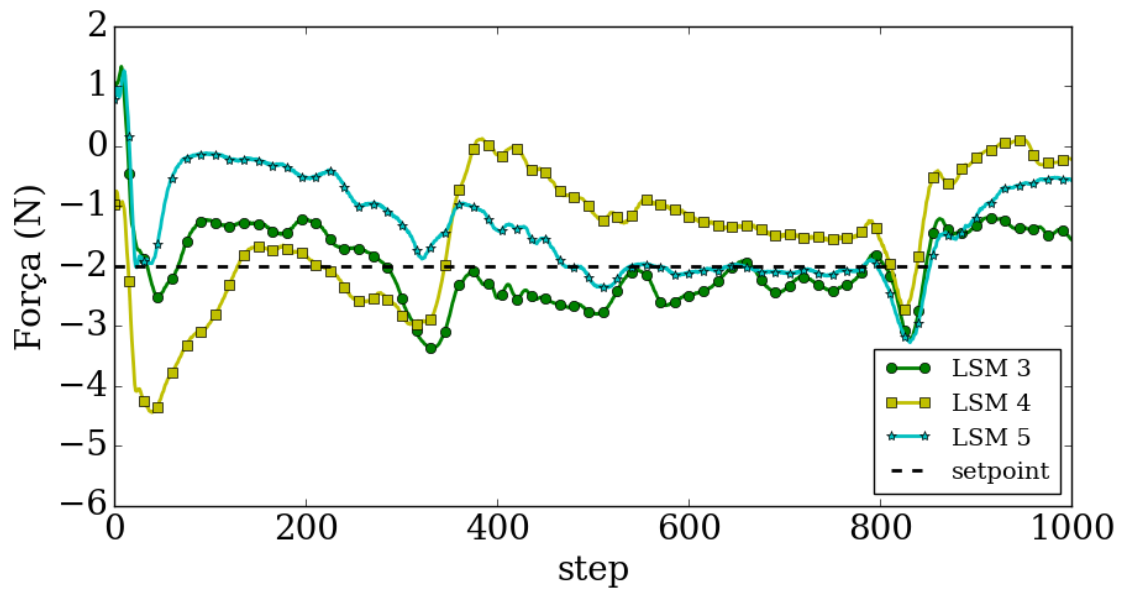


(a)

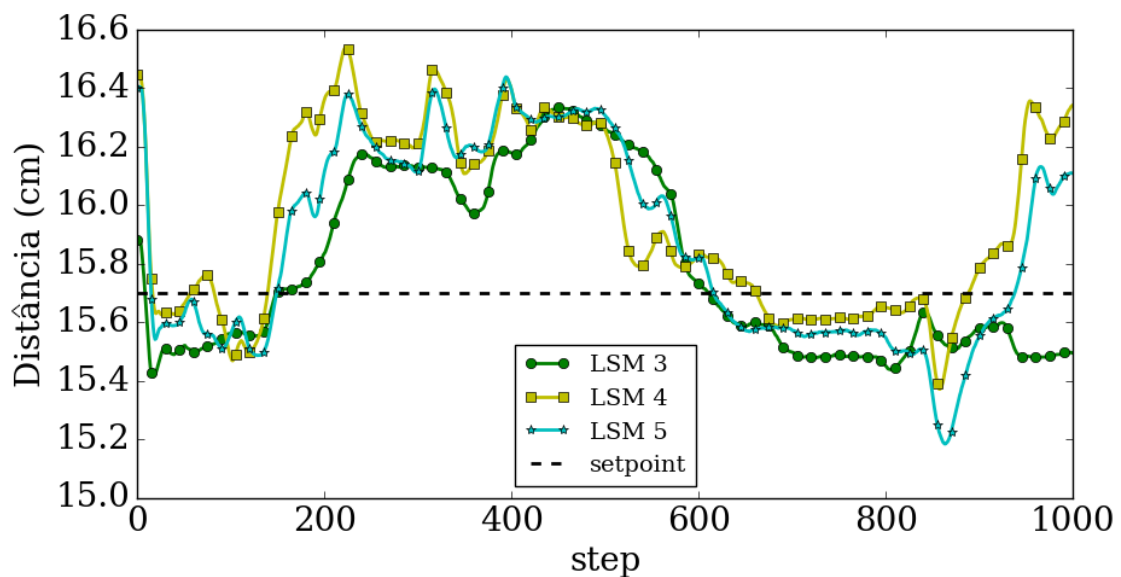


(b)

Figura 33: Resultados obtidos no Experimento D.



(a)



(b)

Figura 34: Resultados combinados das quatro etapas experimentais (A, B, C e D).

Tabela 2: Erro quadrático médio das medidas de força.

Rede Neural	RMSE Força	RMSE Distância
LSM 3	0.4467	0.3155
LSM 4	1.1728	0.3997
LSM 5	1.0181	0.3755

A partir das cinco LSMs criadas, três foram selecionadas para a realização de 250 experimentos cada, a LSM de número 3 mostrou um melhor desempenho de modo geral e menor erro RMSE. Uma análise mais aprofundada dos resultados dessa rede, comparando com os valores obtidos com o controlador PID, é apresentada a seguir.

Os valores de média e variância da saída do controlador LSM 3 foram comparados aos resultados do controlador PID.

Na Figura 35 mostra-se o valor médio de força obtido em 50 experimentos usando o controlador PID e o valor médio de 50 ensaios com o controlador LSM proposto (experimento B), ambos em superfícies sem inclinação. Valores médios de 10 ensaios da trajetória sem qualquer controle também são apresentados para comparação. Através de uma simples inspeção visual, é possível observar que a solução LSM proposta neste trabalho foi capaz de seguir o comportamento de um controlador PID.

Os resultados individuais de cada conjunto de dados foram analisados individualmente. Na Tabela 3 tem-se os valores médios e desvio padrão da força resultante de todos os ensaios realizados usando o controlador LSM escolhido. A última coluna da tabela mostra a porcentagem de valores que foram encontrados dentro do Intervalo de Confiança (IC) de 95.44% dos dados de força do controlador PID, considerando uma distribuição gaussiana. Isso compreende todos os valores de força medidos entre $-3.763N$ e $-0.567N$. A última linha apresenta os resultados de todos os ensaios combinados.

O histograma na Figura 37, apresenta a distribuição de probabilidade dos valores obtidos no sensor de força, nos 250 experimentos com a LSM 3 e 100 experimentos com o PID. Observa-se que as curvas se sobrepõem na maior frequência, porém a LSM apresentou um desvio padrão maior. O pico observado em $2N$ se deve ao fato da sinal de força ter sido condicionado entre os limites de $-10N$ e $2N$. Considerando que a caneta de feltro não mantém contato a superfície quando a força estimada está acima de $0N$. De todos os ensaios, 11,12% de valores de força foram encontrados acima de $0N$, ou seja, não tocando a superfície. A LSM, na média, produziu resultados semelhantes aos do controlador PID, porém apresentou uma maior variância nas leituras de força. Enquanto o PID mostrou um valor de força médio de $\mu_{PID_f} = -2.165N$ e o desvio padrão de $\sigma_{PID_f} = 0.799N$, o controlador LSM exibiu $\mu_{LSM_f} = -2.162N$ e $\sigma_{LSM_f} = 2.418N$.

Tabela 3: Resultados do sensor de força do controlador LSM 3.

Conjunto Experimental	Valor Médio μ_i	Desvio Padrão σ_i	Valores dentro do IC do PID (95%)
Experimento A	$-1.392N$	$1.666N$	73.52%
Experimento B	$-1.982N$	$1.566N$	87.42%
Experimento C	$-3.881N$	$3.348N$	62.14%
Experimento D	$-2.648N$	$1.773N$	75.38%
Combinado	$-2.162N$	$2.418N$	74.15%

Ambos os controladores LSM e PID realizaram a trajetória similarmente, na Figura 36, tem-se os valores médios de leituras do sensor de distância em três situações: sem qualquer controle (curva azul), utilizando o controlador PID (em vermelho), e o controlador LSM (curva verde), ambos sem qualquer inclinação. O *setpoint* utilizado no PID é representado pela linha tracejada azul. Nota-se novamente que o controlador LSM mostrou um comportamento similar ao PID.

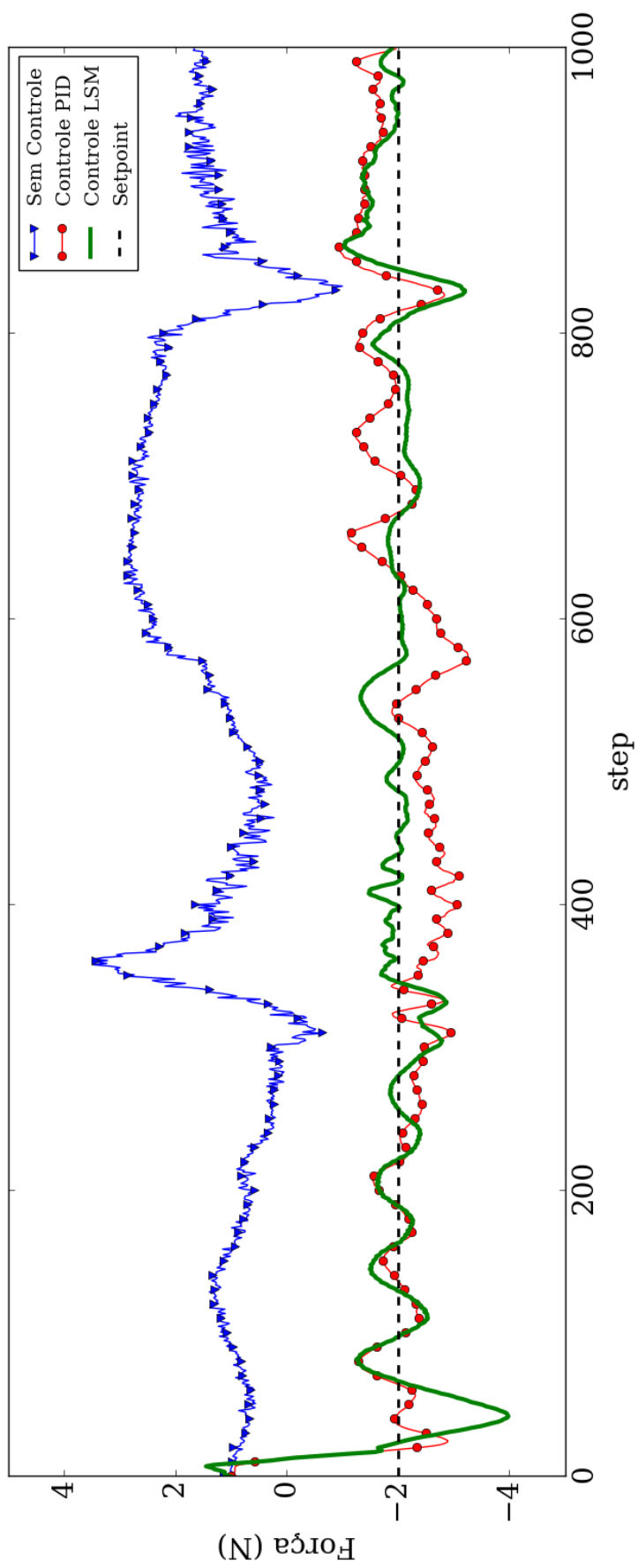


Figura 35: Comparação entre os valores médios de força dos controladores LSM 3 e PID.

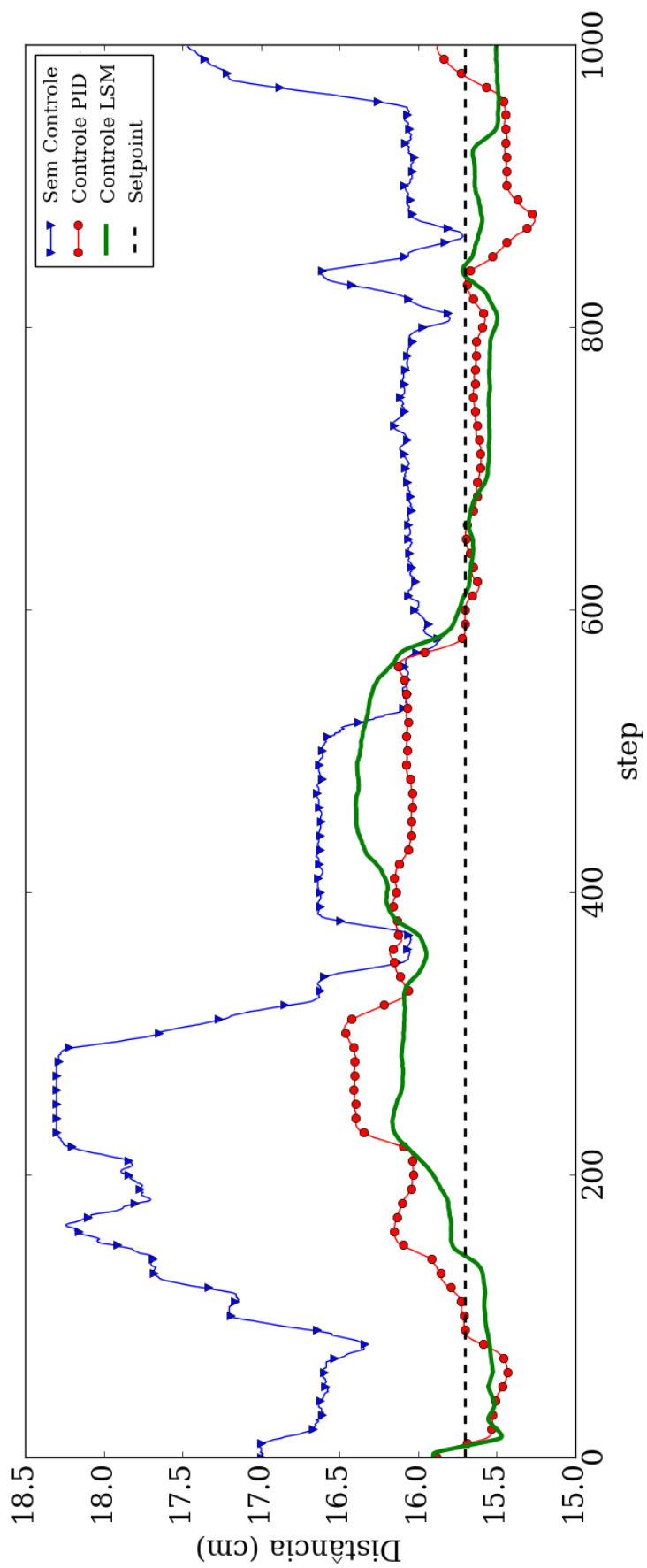


Figura 36: Comparação entre os valores médios de distância de controladores LSM 3 e PID.

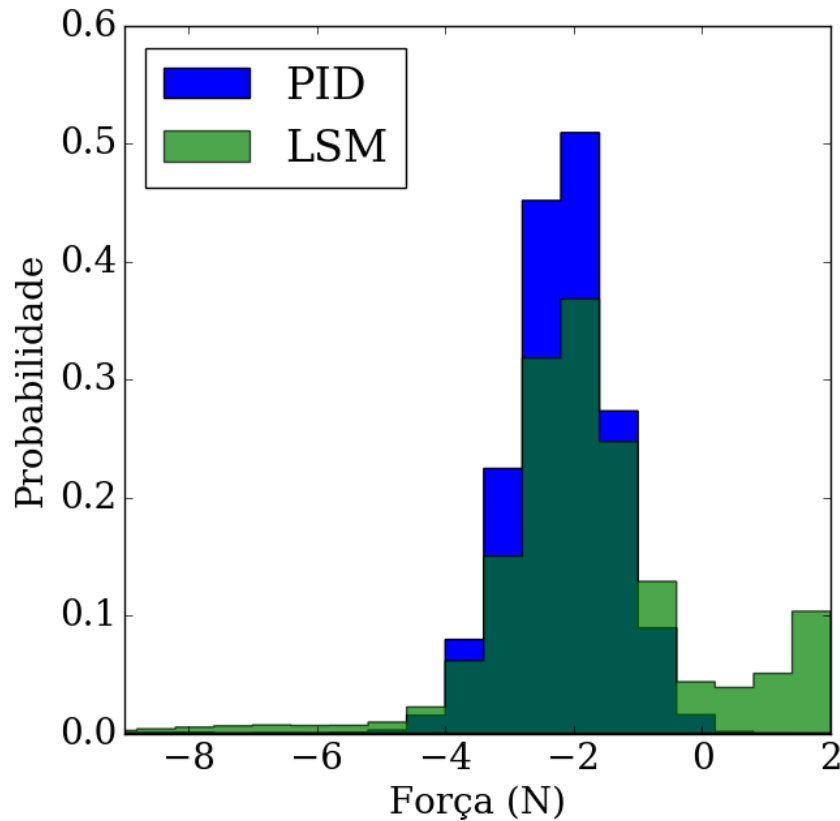


Figura 37: Distribuição de probabilidade das leituras do sensor de força.

A Tabela 4 apresenta os valores médios e desvios padrão da distância medida pelo sensor de distância infravermelho. A última coluna também mostra a porcentagem de valores encontrados entre os 95.44% IC dos dados do controlador PID, considerando uma distribuição gaussiana. Isso inclui todos os valores de distância variando entre $15,43cm$ e $16,21cm$.

Tabela 4: Resultados do sensor de distância do controlador LSM 3.

Conjunto Experimental	Valor Médio μ_i	Desvio Padrão σ_i	Valores dentro do IC do PID (95%)
Experimento A	$15,79cm$	$0,51cm$	97,09%
Experimento B	$15,87cm$	$0,28cm$	97,87%
Experimento C	$15,83cm$	$0,35cm$	97,82%
Experimento D	$15,88cm$	$0,33cm$	97,56%
Combinado	$15,83cm$	$0,41cm$	97,49%

Na Figura 38, apresenta-se o histograma com a distribuição de probabilidade dos valores obtidos no sensor de distância, nos 250 experimentos com a LSM 3 e 100 do PID. Ambos controladores exibiram a mesma média (diferença menor que 0,01%), e um desvio padrão muito similar (diferença de aproximadamente 5%). As leituras de ambas as abordagens foram semelhantes, o controlador PID apresentou uma distância média de $\mu_{PID_d} = 15,82cm$ e desvio padrão de $\sigma_{PID_d} = 0,39cm$ enquanto o controlador LSM exibiu $\mu_{LSM_d} = 15,83cm$ e $\sigma_{LSM_d} = 0,41cm$.

Em uma análise geral, a partir dos dados obtidos nas comparações entre o controlador PID e LSM, apresentados nos gráficos de força e distância (Figuras 35 e

36); nos histogramas dos dados de ambos sensores (Figuras 37 e 38); e dados de média, desvio padrão e sobreposição com o PID (Tabelas 3 e 4). O controlador baseado em Liquid State Machines para controle de distância e força no eixo Z apresentou valores médios, e desvio padrão, de distância muito próximos ao controlador PID. Analisando os valores de força, nota-se que apesar da média próxima, o desvio mostrou-se três maior que o controlador PID. No caso da distância, a variância é pequena porque, enquanto a caneta toca na superfície, as variações mais significativas se devem às mudanças no ângulo entre a superfície e caneta. Já a força pode variar em qualquer momento durante a trajetória, assim apresentando um desvio padrão maior no seu valor. Em observações durante a realização dos experimentos, notou-se que valores de força dentro da faixa $-4N$ e $0N$ apresentavam o resultado esperado do sistema de controle, onde o atrito entre a caneta de feltro e superfície não era nem exacerbado ou inexistente. Considerando-se este intervalo de força, o controlador LSM manteve-se 74,25% do tempo, dentro deste intervalo, contra 98,51% do PID (desconsiderando as primeiras 50 amostras de cada sinal, para eliminar o tempo de estabilização dos controladores). Na parcela acima de $0N$ de força, o conjunto de treinamento (PID) apresentou 0,22% dos valores, enquanto o controlador LSM 11,12%.

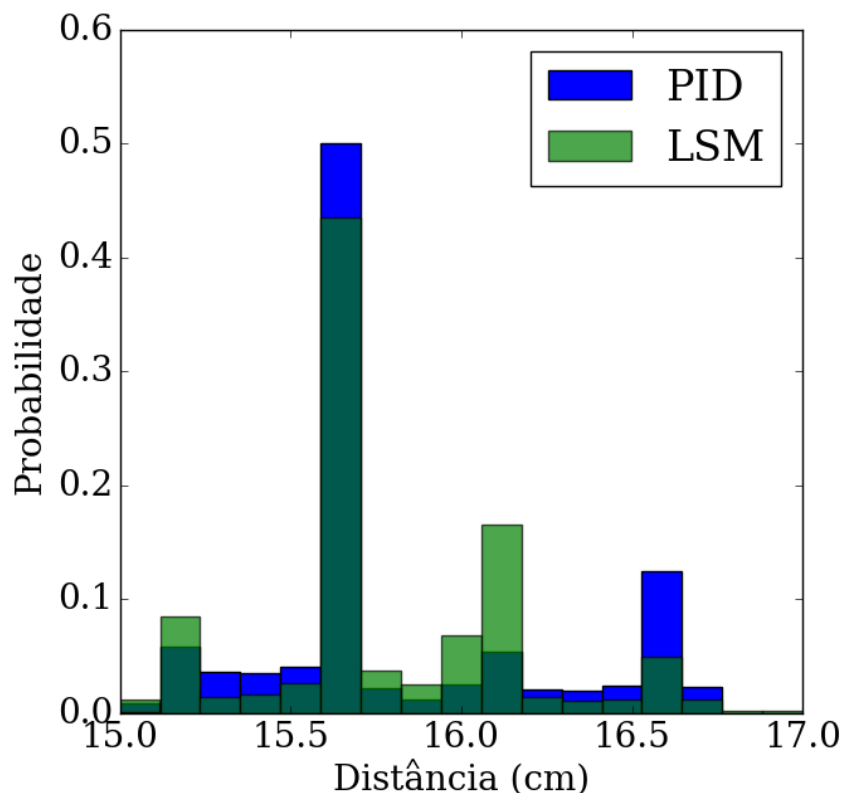
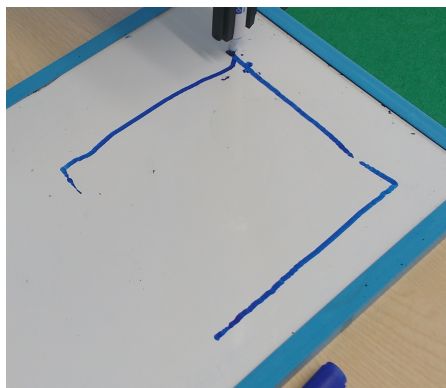


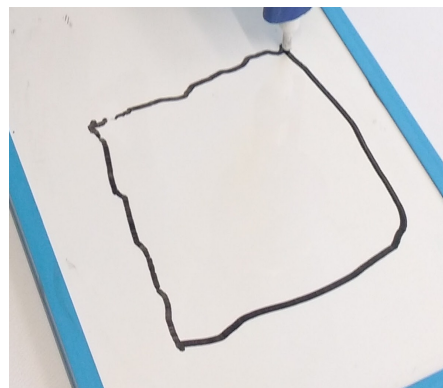
Figura 38: Distribuição de probabilidade das leituras do sensor de distância.

As figuras 39(a) e 39(b) apresentam resultados de dois quadrados desenhados sem o sistema de controle de posicionamento proposto, com o robô BAXTER realizando a trajetória quadrada com uma caneta de feltro sobre um quadro branco. Na primeira imagem, a caneta de feltro não toca a superfície em alguns momentos da trajetória, resultando em falhas, ou seja, pouca força é aplicada. Na segunda, é aplicada uma força maior que a necessária, causando uma distorção no desenho devido ao atrito

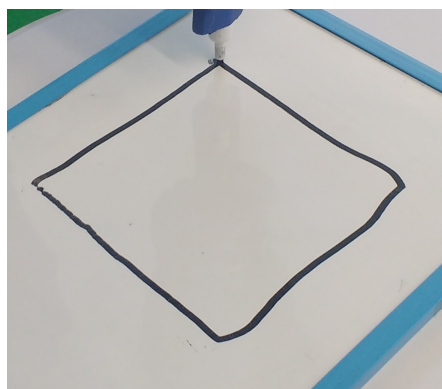
da caneta na superfície. Na figura 39(c) mostra-se a trajetória quadrada utilizando o sistema de controle proposto aqui proposto, utilizando o controlador LSM 3. Neste caso, a caneta de feltro não flutuou e nem arrastou, o que é o resultado esperado.



(a) Baixa força (sem controle)



(b) Alta força (sem controle)



(c) Força Estável (controle LSM)

Figura 39: Exemplo do Robô BAXTER desenhando em um quadro branco com e sem controle no eixo Z

4.0.3 Nova trajetória

Nos experimentos expostos até o momento, foram utilizadas diferentes superfícies, porém todas com a mesma trajetória. A seguir são apresentados resultados preliminares da mesma rede apresentada nesse trabalho para uma nova trajetória. Observa-se que não é feito nenhum treinamento, apenas os valores de *offset* são alterados para acomodar a nova trajetória. Experimentos utilizando a LSM 3 foram executados em uma superfície mais alta. As novas trajetórias são observadas na Figura 40. Em tracejado (T1 - Trajetória 1) estão os ângulos das juntas utilizadas no treinamento e nos experimentos previamente exibidos. As linhas contínuas (T2 - Trajetória 2) pertencem a nova trajetória, executada em uma superfície rígida, posicionada aproximadamente $5,8\text{cm}$ acima do nível da anterior. Realizou-se 25 experimentos nesta etapa, uma quantidade menor em relação ao número de experimentos realizados nas etapas anteriores, a questão de como uma mesma rede se comportaria sob uma nova trajetória surgiu durante a realização dos outros experimentos, devido ao curto espaço de tempo restante, poucos experimentos puderam ser realizados para essa investigação.

A Figura 41 apresenta os valores de força obtidos em 25 experimentos utilizando

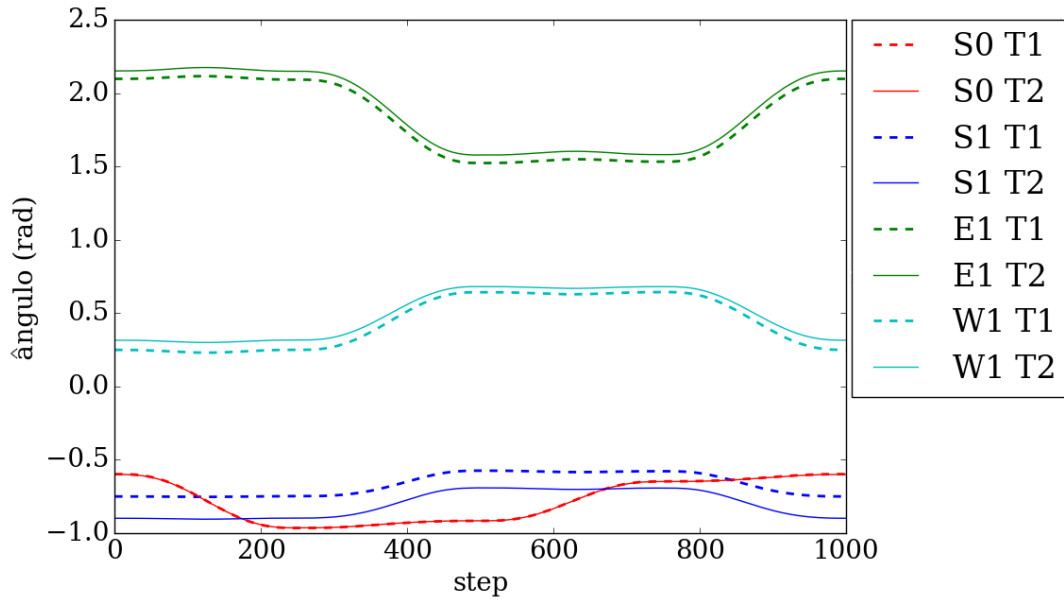


Figura 40: Comparação entre as Trajetórias 1 e 2.

a segunda trajetória, em uma superfície mais alta. O valor de força médio foi de $\mu_{LSM_f} = -2.293N$ e desvio padrão $\sigma_{LSM_f} = 2.186N$. Observa-se que estes valores são próximos aos encontrados nos experimentos anteriores, utilizando a mesma LSM (ver Tabela 3), apresentando uma diferença de $0,124N$ no valor médio, e um desvio padrão menor ($2.186N$ vs $2.418N$) que o valor dos experimentos na superfície anterior combinados.

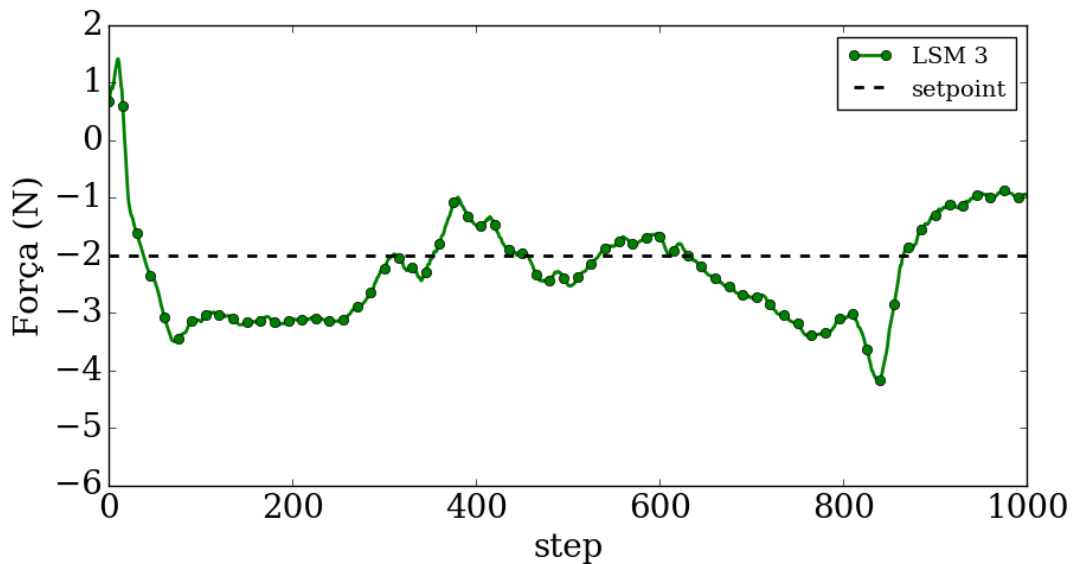


Figura 41: Média dos valores de força, utilizando a LSM 3 na Trajetória 2.

Na Figura 42, exibe-se a média dos valores do sensor de distância obtidos nos mesmos 25 experimentos. Observa-se que os valores de distância estão distantes do *setpoint* desejado, devido ao fato que o sensor é sensível à reflectância da superfície, que foi alterada, e pequenas alterações no ângulo entre mesa e braço pode ter ocorrido durante a criação da nova trajetória. A distância média encontrada foi

$\mu_{LSM_d} = 16.71cm$ e desvio padrão $\sigma_{LSM_d} = 0.71cm$. Observa-se que, apesar do *setpoint* utilizado no PID ser inalcançável, o sistema foi capaz de mantê-lo no valor mais baixo possível, enquanto mantinha a força próxima do *setpoint* de $2N$, utilizado no conjunto de treinamento.

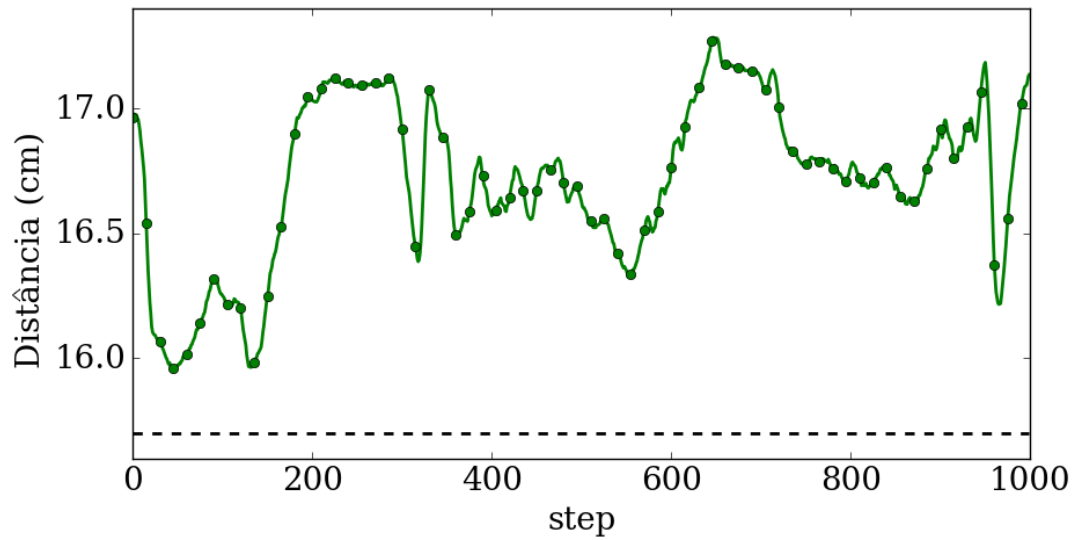


Figura 42: Média dos valores de distância, utilizando a LSM 3 na Trajetória 2.

5 CONCLUSÃO

Neste trabalho foi apresentado e avaliado um *framework*, baseado em *Liquid State Machines*, para controlar a distância e a força aplicada no eixo Z, durante a execução de uma trajetória bidimensional no plano XY. O principal objetivo deste trabalho, foi o desenvolvimento de um sistema de controle para redução de erro de posicionamento, utilizando o paradigma de *Spiking Neural Networks*, capaz de ser exportado para um computador neuromórfico. O sistema foi projetado, para posteriormente, trabalhar em paralelo com um controlador de trajetórias, também baseado em LSM. Foram criados cinco líquidos com conexões distintas, com 200 neurônios cada, utilizando os mesmos parâmetros da rede de controle de trajetória.

Foi proposto o treinamento das redes utilizando dados gerados a partir de um controlador PID, executados diretamente no robô BAXTER, sem a utilização de simuladores. O conjunto de treinamento criado possui 100 experimentos executados com um controlador PID, em três posições distintas, sobre uma superfície macia. O controlador foi programado para manter os valores de dois sensores, força estimada no *endpoint* do robô e distância retornada por um sensor posicionado no *pulso* do robô, em determinados *setpoints*.

Na etapa de aprendizado, cinco redes distintas foram treinadas com os dados do conjunto de treinamento. A avaliação da rede foi feita utilizando a mesma superfície macia sem inclinação, e outra rígida, em três posições: sem inclinação, com 4° no eixo Y e -4° no eixo X, a última sendo do lado oposto ao utilizado no conjunto de treinamento. Inicialmente as cinco redes criadas foram avaliadas em 50 experimentos, das quais selecionou-se as três melhores para realização dos 200 experimentos, variando tipo e posição da superfície, totalizando 250 execuções para cada rede. A melhor rede também foi avaliada em uma breve investigação, alterando a trajetória, em 25 experimentos utilizando uma superfície mais elevada.

O sistema proposto foi capaz de aprender o comportamento de um controlador PID e mostrou-se capaz de auxiliar o robô a executar uma trajetória em cima de uma superfície plana, reduzindo o erro, seguindo sinais de força e distância constantes. A trajetória usada para avaliar o sistema foi um caso de pior cenário, onde a caneta de feltro nunca tocaria na superfície de desenho, isto é, um erro de 100%. Neste caso a força estimada no *endpoint* pelo robô mantêm-se acima de $0N$, no conjunto de treinamento criado com o controlador PID, 0,22% dos valores encontram-se acima deste limite. O controlador LSM 3 apresentou 11,12% dos valores acima de $0N$.

Os controladores PID e LSM apresentaram valores médios de força semelhantes, com uma diferença de 1,38% ($\mu_{PID_f} = -2.165N$ e $\mu_{LSM_f} = -2.162N$), porém o controlador LSM apresentou um desvio padrão 3,02 vezes maior ($\sigma_{PID_f} = 0.799N$ e $\sigma_{LSM_f} = 2.418N$). Quando testado em uma trajetória distinta à utilizada no trei-

namento, o controlador LSM foi capaz de manter a força próximo ao valor médio de $\mu_{LSM_f} = -2.293N$, com desvio padrão de $\sigma_{LSM_f} = 2.186N$. Valores estes próximos dos encontrados nos experimentos com a trajetória original. Mostrando capacidade da rede de funcionar em superfícies distintas, com um simples ajuste de *offset*.

Considerando uma distribuição Normal dos resultados obtidos com o controlador PID em força e distância. O intervalo de confiança de 95% deste encontra-se entre os limites de $-3.763N$ e $-0.567N$. Analisando como os resultados de força se encaixam nesta distribuição, encontrou-se dentro deste intervalo 74.15% dos valores de força, entre todos os experimentos executados com o controlador LSM. Para o dado do sensor de distância, encontrou-se 97.49% dos valores do controlador LSM dentro do mesmo intervalo de confiança do PID, valores entre $15,43cm$ e $16,21cm$.

5.1 Trabalhos Futuros

A abordagem aqui apresentada foi projetada com a intenção de trabalhar em um computador neuromórfico, em paralelo com o sistema de controle principal, também baseado em LSM.

Trabalhos futuros incluem a combinação de ambos os sistemas e a migração para o sistema neuromórfico SpiNNaker, já disponível para uso na Plymouth University, onde os experimentos deste trabalho foram executados.

Apesar do sistema não ter alcançado resultados idênticos ao controlador PID utilizado para o treinamento, acredita-se que os resultados aqui gerados, podem ser melhorados a partir de um estudo mais aprofundado, com melhorias no conjunto de treinamento, alterações nas escolhas dos sensores, ou até mesmo inclusão de mais sensores ao sistema.

Investigação de como a rede neural trata as não linearidades existentes no comportamento do robô, métodos tradicionais de controle, geralmente não levam em conta estas não linearidades, enquanto a rede neural, quando treinada com um conjunto de dados gerado no robô real, provavelmente vai compensar a influência destes dados.

O sensor de distância não gerou bons resultados porque pequenas alterações no ângulo entre o braço do robô e a superfície influenciam no valor de distância obtido. Portanto está sendo estudada a possibilidade de uso de outros sensores.

Acredita-se que a precisão do sistema pode ser melhorada através da inclusão de diferentes elevações de superfície, e trajetórias de desenho no conjunto de treinamento, aumentando a diversidade dos dados para a etapa de aprendizado. Por exemplo, podemos utilizar dados com o próprio sistema de controle de trajetórias por LSM para treinamento. Além disso, podem ser feitas outras investigações, como a escolha de outros tipos de controladores para o aprendizado, e como as mesmas podem afetar o desempenho do sistema proposto.

REFERÊNCIAS

- AZAMBUJA, R. de; CANGELOSI, A.; ADAMS, S. V. Diverse, noisy and parallel: a new spiking neural network approach for humanoid robot control. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 2016, Vancouver, BC. **Proceedings...** New York: IEEE, 2016. p.1134–1142.
- BEBIS, G.; GEORGIOPOULOS, M. Feed-forward neural networks. **IEEE Potentials**, New York, v.13, n.4, p.27–31, Oct. 1994.
- BROOKS, R. R.; IYENGAR, S. S. **Multi-sensor Fusion: fundamentals and applications with software**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998.
- BUONOMANO, D. V.; MAASS, W. State-dependent computations: spatiotemporal processing in cortical networks. **Nature Reviews Neuroscience**, New York, v.10, n.2, p.113–125, 2009.
- DRUBACH, D. **The brain explained**. Ann Arbor: Prentice Hall Health, 2000. 168 p. ISBN 978-01-379-6194-8.
- FERNANDO, C.; SOJAKKA, S. Pattern Recognition in a Bucket. In: ADVANCES IN ARTIFICIAL LIFE, 2003, Berlin. **Proceedings...** Springer Berlin Heidelberg, 2003. p.588–597.
- FITZGERALD, C. Developing baxter. In: IEEE CONFERENCE ON TECHNOLOGIES FOR PRACTICAL ROBOT APPLICATIONS (TEPRA), 2013, Woburn. **Proceedings...** New York: IEEE, 2013. p.1–6.
- FLASH, T.; HOGAN, N. The coordination of arm movements: an experimentally confirmed mathematical model. **The journal of Neuroscience**, Washington, v.5, n.7, p.1688–1703, 1985.
- FONTANELLI, D. et al. Collaborative localization of robotic wheeled walkers using interlaced Extended Kalman Filters. In: IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE PROCEEDINGS, 2016, Taipei. **Proceedings...** New York: IEEE, 2016. p.1–6.
- FURBER, S. et al. Overview of the SpiNNaker System Architecture. **IEEE Trans. Computers**, New York, v.62, n.12, p.2454–2467, 2013.
- GARCIA, F. et al. Data fusion for overtaking vehicle detection based on radar and optical flow. In: INTELLIGENT VEHICLES SYMPOSIUM, 4, 2012, Madrid. **Proceedings...** New York: IEEE, 2012. p.494–499.

GERSTNER, W.; KISTLER, W. M. **Spiking neuron models: single neurons, populations, plasticity**. Cambridge: Cambridge University Press, 2002. 480 p. ISBN 978-05-218-9079-3.

HOERL, A. E.; KENNARD, R. W. Ridge regression: biased estimation for nonorthogonal problems. **Technometrics**, Abingdon, v.12, n.1, p.55–67, 1970.

JAEGER, H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. **GMD Technical Report**, Bonn, Germany, v.148, p.34, 2001.

JOSHI, P.; MAASS, W. Movement generation with circuits of spiking neurons. **Neural Computation**, Massachusetts, v.17, n.8, p.1715–1738, 2005.

KANDILLI, I. et al. Online Monitoring Of Tool Wear In Drilling and Milling By Multi-Sensor Neural Network Fusion. In: INTERNATIONAL CONFERENCE ON MECHATRONICS AND AUTOMATION, 2007, Harbin. **Proceedings...** New York: IEEE, 2007. p.1388–1394.

MAASS, W. Networks of spiking neurons: the third generation of neural network models. **Neural Networks**, Amsterdã, v.10, n.9, p.1659–1671, 1997.

MAASS, W.; NATSCHLÄGER, T.; MARKRAM, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. **Neural computation**, Massachusetts, v.14, n.11, p.2531–2560, 2002.

MEROLLA, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. **Science**, Washington, v.345, n.6197, p.668–673, 2014.

MONROE, D. Neuromorphic Computing Gets Ready for the (Really) Big Time. **Commun. ACM**, New York, NY, USA, v.57, n.6, p.13–15, June 2014.

MORAIS BORGES, G. de; BRUSAMARELLO, V. Bayesian fusion of multiple sensors for reliable heart rate detection. In: IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE PROCEEDINGS, 2014, Montevideo. **Proceedings...** New York: IEEE, 2014. p.1310–1313.

O’CONNOR, P. et al. Real-Time Classification and Sensor Fusion with a Spiking Deep Belief Network. **Frontiers in Neuroscience**, Lausanne, v.7, n.178, p.115–128, 2013.

OGATA, K.; YANG, Y. **Modern Control Engineering Fifth Edition**. Upper Saddle River, NJ, USA: Prentice-Hall, 2010. 976 p. ISBN: 978-01-330-0225-6.

OPTOELECTRONIC Device. Washington: SHARP MICROELECTRONICS, 2005. n.GP2Y0A21YK. Rev. 3.

PALUMBO, F. et al. Multisensor data fusion for activity recognition based on reservoir computing. In: INTERNATIONAL COMPETITION ON EVALUATING AAL SYSTEMS THROUGH COMPETITIVE BENCHMARKING, 2013, Valencia. **Proceedings...** Springer Berlin Heidelberg, 2013. p.24–35.

PFEIL, T. et al. Six networks on a universal neuromorphic computing substrate. **Frontiers in Neuroscience**, Lausanne, v.7, n.11, 2013.

PRATT, G. A.; WILLIAMSON, M. M. Series elastic actuators. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 1995, Pittsburgh. **Proceedings...** New York: IEEE, 1995. v.1, p.399–406.

QUIGLEY, M. et al. ROS: an open-source robot operating system. In: ICRA WORKSHOP ON OPEN SOURCE SOFTWARE, 2009, Kobe, Japan. **Proceedings...** New York: IEEE, 2009. v.3, n.3.2, p.5.

ROHMER, E.; SINGH, S. P. N.; FREESE, M. V-REP: a versatile and scalable robot simulation framework. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2013, Tokyo. **Proceedings...** New York: IEEE, 2013. p.1321–1326.

SHEN, J. et al. Darwin: a neuromorphic hardware co-processor based on spiking neural networks. **SCIENCE CHINA Information Sciences**, Valencia, v.59, n.2, p.1–5, 2016.

SICILIANO, B. et al. **Robotics: modelling, planning and control**. Springer-Verlag London: Springer Science & Business Media, 2010. 632 p. ISBN 978-1-84628-641-4.

STIEBER, M. E.; PETRIU, E.; VUKOVICH, G. Instrumentation architecture and sensor fusion for systems control. **IEEE Transactions on Instrumentation and Measurement**, New York, v.47, n.1, p.108–113, Feb. 1998.

VERSTRAETEN, D. et al. Isolated word recognition with the Liquid State Machine: a case study. **Information Processing Letters**, Washington, v.95, n.6, p.521 – 528, 2005. Applications of Spiking Neural Networks.

VILLANUEVA, J. M. M. et al. Partial discharge measurement and uncertainty analysis based on fuzzy data fusion. In: IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE, 2012, Graz, Austria. **Proceedings...** New York: IEEE, 2012. p.1033–1038.