INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS GERAIS - *CAMPUS* BETIM

BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Arthur Carlos de Faria

**Modelagem e Controle de Robô Quadrúpede Utilizando Redes Neurais preditivas e Adaptativas**

Betim

2022

ARTHUR CARLOS DE FARIA

**Modelagem e Controle de Robô Quadrúpede Utilizando Redes Neurais preditivas e Adaptativas**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Leandro Freitas de Abreu

Coorientador: Virgil Del Duca Almeida

Betim

2022

**LISTAS DE FIGURAS**

| Figura 1 - | Palavra | 09 |
| --- | --- | --- |
| Figura 2 - | Palavra palavra palavra | 09 |
| Figura 3 - | Palavra palavra palavra palavra palavra palavra palavra palavra palavra | 10 |
| Figura 4 - | Palavra palavra palavra palavra palavra | 10 |
| Figura 5 - | Palavra palavra palacra palavra palavra | 11 |
| Figura 6 - | Palavra palavra palavra palavra palavra | 11 |
|  |  |  |
|  |  |  |
|  |  |  |

**LISTAS DE TABELAS**

| Tabela 1 - | Modelos palavra palavra palavra | 08 |
| --- | --- | --- |
| Tabela 2 - | Modelos palavra palavra palavra | 10 |
| Tabela 3 - | Modelos palavra palavra palavra | 12 |
| Tabela 4- | Palavra palavra palavra palavra palavra | 12 |
| Tabela 5 - | Palavra palavra palavra palavra palavra | 12 |

**SUMÁRIO**

[**1. INTRODUÇÃO**](#_heading=h.30j0zll) **7**

[1.1. Justificativa](#_heading=h.929cqt8lanqa) 13

[1.2. Colocação do Problema](#_heading=h.9yna9y22jjw9) 13

[1.3. Objetivos](#_heading=h.bf58co3a1yur) 13

[1.3.1 Objetivos impostos ao robô quadrúpede](#_heading=h.1v3knfpmw6mi) 14

[1.4 Organização do Trabalho](#_heading=h.c7ia4btid0c8) 15

[**2 FUNDAMENTAÇÃO TEÓRICA**](#_heading=h.8eh42d4xsvzp) **16**

[2.1 Métodos de controle](#_heading=h.u157zwgptka0) 17

[2.1.1 Controle Preditivo](#_heading=h.et4xb9jo7x2e) 17

[2.1.1 Método do Modelo Interno](#_heading=h.wrqgri3ani0g) 18

[2.3 Redes Neurais Artificiais](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.gsem6jwpz6l) [18](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.gsem6jwpz6l)

[2.1.1 Redes Neurais Preditivas em sistemas desconhecidos](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.9gs8ob4h74e) [18](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.9gs8ob4h74e)

[**3 METODOLOGIA**](#_heading=h.vcpem2ehjj08) **19**

[3.1 CRONOGRAMA](#_heading=h.w3lxgd28ao5o) 19

[3.2 RECURSOS](#_heading=h.2p4c0b2li65x) 21

[3.2.1 Materiais](#_heading=h.hjgp3197ry5c) 21

[3.2.2 Ferramentas computacionais](#_heading=h.5houcbcosbds) 21

[3.2.3 Equipamentos Utilizados](#_heading=h.r54y6zeds2ol) 21

[3.3 MONTAGEM DO SISTEMA](#_heading=h.widr0fk1n63x) 21

[3.3.1 Modelo em 3D](#_heading=h.mov5j0u6o6df) 21

[3.3.2 Passo a Passo de Montagem](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.yufzyvcmisfl) [21](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.yufzyvcmisfl)

[3.3.3 Testes e validação da montagem](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.lu4z3jcjl6el) [21](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.lu4z3jcjl6el)

[3.4 Estudo e Modelagem do Sistema](#_heading=h.3fb78m4wddri) 21

[3.4.1 Coleta de Dados do Comportamento do Sistema frente a diferentes entradas](#_heading=h.4olrxm5qwoaq) 21

[3.4.2 Definição de um modelo matemático do Sistema](#_heading=h.23c9lu23cy7x) 21

[3.4.3 Estudo do modelo matemático utilizando ferramentas computacionais](#_heading=h.ie7iiqz4337m) 21

[3.5 MODELAGEM DAS REDES NEURAIS](#_heading=h.2dj00oypwm8p) 21

[3.5.1 Descrição do sistema de controle e o algoritmo a ser utilizado](#_heading=h.3strcerbjq8g) 21

[3.5.2 Definição dos Critérios de Desempenho](#_heading=h.l8g56awxsezp) 21

[3.5.3 Definição dos modelos de Rede Neural](#_heading=h.3wgxyvse57v8) 21

[3.5.4 Algoritmo de propagação e retropropagação](#_heading=h.pbab29z7cz7z) 21

[3.5.5 Análise de Desempenho e Validação do Sistema](#_heading=h.cqlyzt26n27i) 22

[3.6 APLICAÇÃO DAS REDES NEURAIS NO ROBÔ QUADRUPEDE](#_heading=h.9op8jq3g7bcy) 22

[3.6.1 Aplicação das Redes Neurais pré treinadas no robô quadrúpede](#_heading=h.nqwxtcaphxc) 22

[3.6.2 Análise dos Critérios de Desemepenho Na Rede Pré Treinada](#_heading=h.dgwljg9w8504) 22

[3.6.3 Continuidade do Treinamento no Ambiente Real](#_heading=h.56pv3xsge7v5) 22

[3.6.4 Continuidade do Treinamento no Ambiente Real](#_heading=h.ec6hsidcyksv) 22

[3.7 OTIMIZAÇÃO DAS REDES NEURAIS](#_heading=h.3zmpbn9oi03v) 22

[3.7.1 Técnicas para aperfeiçoamento das Redes Neurais](#_heading=h.e6cyy5ro1pf) 22

[3.7.2 Interpretação e conclusão dos resultados da otimização](#_heading=h.i1bmu5e4r2ld) 22

[3.7.3 Registro de Desempenho](#_heading=h.n47jv8ortjg7) 22

[**4.0 RESULTADOS E DISCUSSÃO**](#_heading=h.nnlsqdp3ej3z) **22**

[**5.0 CONCLUSÃO**](#_heading=h.xmzngkpkf87i) **22**

[**6.0 REFERÊNCIAS**](#_heading=h.1t3h5sf) **23**

# 1. INTRODUÇÃO

A utilização de robôs com controlador embutido, ou como são conhecidos, os sistemas embarcados, é crescente, e atualmente existem diversas tentativas de fazer com que alcancem a inteligência artificial, para que sejam capazes de analisar e aprender com o seu ambiente para atingir um determinado objetivo. Neste trabalho, pretende-se investigar e aplicar técnicas de modelagem de redes neurais a fim de produzir um robô quadrúpede que seja capaz de realizar as seguintes ações:

* Prever as consequências de suas ações, ou seja predizer os estados futuros de seus sensores com base nas decisões tomadas.
* Comandar os motores a fim de atingir um objetivo(Permanecer em equilíbrio e evitar obstáculos)

Dessa forma, este trabalho terá como foco principal desenvolver um modelo de inteligência artificial em um sistema embarcado, mostrado na imagem abaixo, investigar os resultados obtidos e documentar o desempenho do controlador preditivo para tal aplicação.

## 1.1. Justificativa

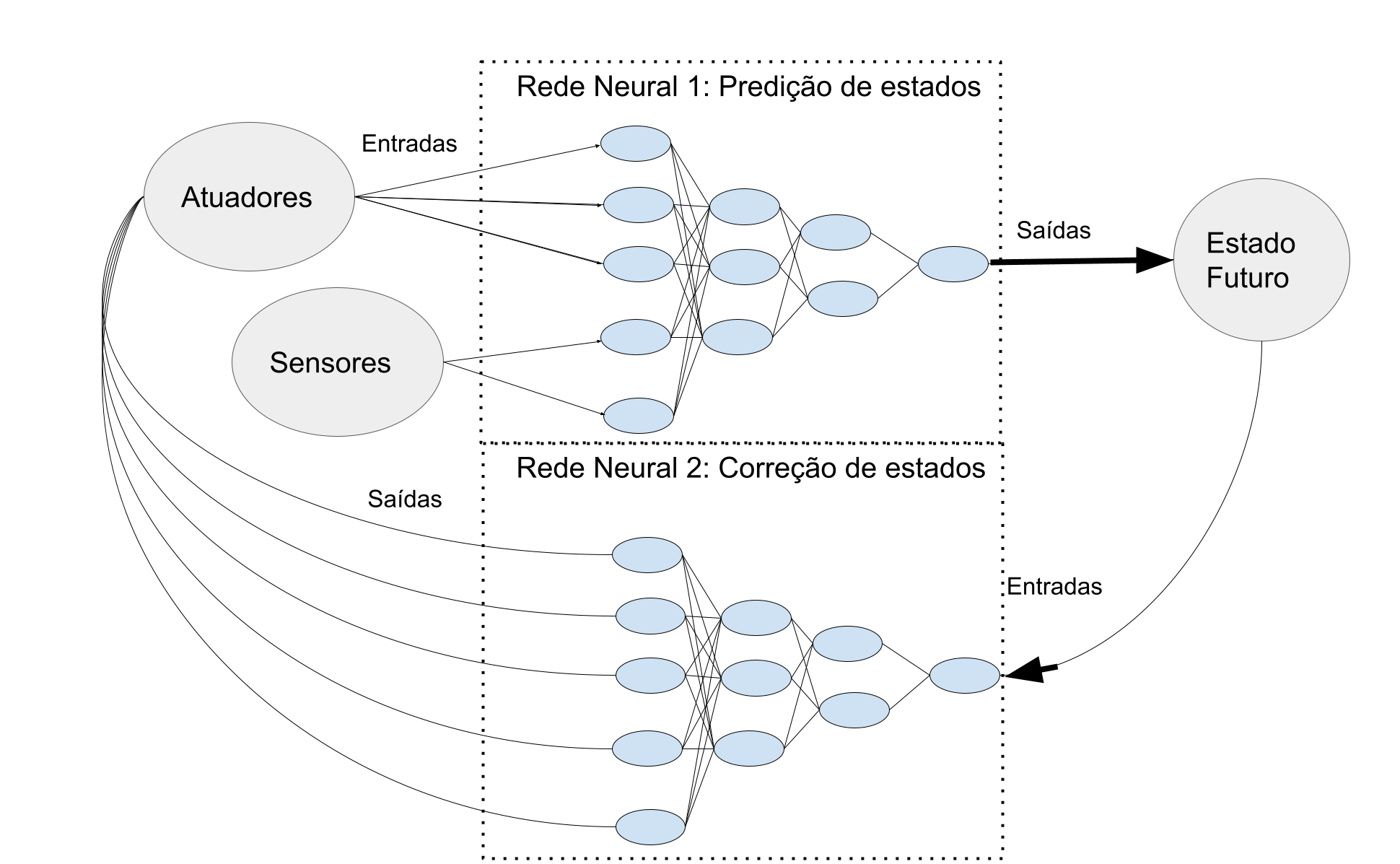
Reproduzir técnicas de controle, associadas a redes neurais artificiais, para projetar e aplicar um controlador não linear, tal que seja capaz de prever o comportamento do robô quadrúpede com base no estado atual e nas escolhas tomadas pelo robô. A utilização de redes neurais para este problema é devido ao fato de serem naturalmente adaptativas, o que as tornam essencialmente vantajosas em relação ao controlador convencional.

## 1.2. Colocação do Problema

Pretende-se projetar um controlador baseado em redes neurais para solucionar o problema descrito e avaliar o desempenho do robô ao longo de seu treinamento utilizando o microcontrolador Arduino Mega.

## 1.3. Objetivos

O trabalho está voltado para a construção de dois modelos de redes neurais, responsáveis pelo controle do robô. Um modelo capaz de prever os próximos N estados, com base no estado inicial dos sensores e um modelo capaz de prever os próximos comandos a serem dados para atingir os objetivos impostos no ítem 1.3.1.



Com isso, pode-se resumir o objetivo do trabalho em testar a seguinte característica do sistema que define comportamento de uma inteligência artificial:

“A aptidão ou capacidade de um dispositivo para desempenhar funções que são normalmente associadas à inteligência humana, tais como raciocínio, aprendizagem e auto-aperfeiçoamento”.

http://www.ppgee.eng.ufba.br/teses/3fa84f4664183cab6c0c9022d582c762.pdf

### **1.3.1 Objetivos impostos ao robô quadrúpede**

O robô tem os seguintes objetivos:

I: Manter-se em equilíbrio, mesmo que hajam perturbações ;

II: Minimizar as vibrações e impactos, a fim de preservar as conexões, ou seja, manter a aceleração nos três eixos o mais próximo de zero possível ;

III:Manter-se virado para o lado que não haja obstáculos, ou seja, sempre que detectar um objeto em uma distância menor do que a distância máxima DM, ele deverá virar o sensor para outra direção.

IV: Prever as consequências de suas ações, ou seja, estimar qual serão os estados resultantes dos comandos dados aos motores para os próximos N passos.

## 1.4 Organização do Trabalho

O Trabalho está dividido nos seguintes temas:

**I- Fundamentação Teórica:** Revisão Bibliográfica acerca de trabalhos anteriores envolvendo Controle Preditivo, Redes Neurais e sistemas embarcados baseados que se aproximam do robô utilizado nos testes. Introdução aos assuntos contidos no trabalho.

**II- Cronograma:** Apresenta as datas limites para realização de cada tarefa, na forma de tabela a partir do início do desenvolvimento até a defesa do TCC.

**III- Descrição e modelagem do Sistema a ser controlado:** Descreve de forma matemática e conceitual, o sistema embarcado a ser controlado e o ambiente ao qual é submetido, além disso, justifica as escolhas de cada componente com base nas variáveis que devem ser medidas e controladas pelo microcontrolador utilizado. Também define as entradas e saídas do modelo.

**V- Descrição do sistema de controle e o algoritmo a ser utilizado:** Descreve a lógica a ser utilizada pelo algoritmo, além do microcontrolador a ser utilizado, baseado no sistema descrito no item III.

**VI- Definição dos Critérios de Desempenho:** Descreve os critérios a serem analisados ao longo do treinamento de forma a possibilitar a comparação dos resultados obtidos com os de controladores convencionais utilizados em sistemas lineares.

**VII - Definição dos modelos de Rede Neural:** Define de forma lógica, possível modelo a ser utilizado para rede neural, tanto para predição de estados, quanto para correção de estados.

**VIII- Algoritmo de propagação e retropropagação:** Descrição matemática da relação entre a entrada e saída de cada modelo, definição do erro a ser utilizado no treinamento e atualização dos pesos de cada camada na retropropagação do erro.

**IX- Treinamento da rede neural:** Descrição do Treinamento do modelo e da obtenção dos dados.

**X- Teste da rede neural:** Valida o modelo com valores colhidos do sistema.

**XI- Montagem do sistema Embarcado e Aplicação dos modelos de treinamento e teste:** Descreve a forma de construção, conexão e utilização de todo o sistema.

**XII - Análise dos critérios de desempenho:** Analisa os valores de forma gráfica, a respeito do desempenho da rede neural utilizada a fim de validar o sistema.

**XIII - Otimização da rede neural:** Aperfeiçoamento da rede neural após a sua validação.

**XIV- Interpretação e conclusão dos resultados:** Comparação, descrição e detalhamento dos resultados obtidos.

# 2 FUNDAMENTAÇÃO TEÓRICA

A teoria utilizada para compreender as técnicas utilizadas para compreensão deste trabalho está nos tópicos a seguir:

* Redes Neurais Artificiais preditivas
* Método do Modelo Interno

## 2.1 Métodos de controle

### **2.1.1 Controle Preditivo**

O controle preditivo utiliza técnicas para que seja possível prever o comportamento de um determinado sistema a partir de um modelo matemático, este modelo pode ser tanto uma Rede Neural quanto um modelo matemático da saída em relação às variáveis de estado. As vantagens deste tipo de Controle é poder prever o erro de determinada a ação e tomar a melhor ação de controle a partir do erro previsto, portanto não é preciso que o erro aconteça.

No artigo ***Controle Preditivo Baseado em Modelo para Conversores Formadores de Rede com Operação,***é feito o controle por chaveamento de forma preditiva para minimizar o erro em Conversores seguindo os seguintes passos:

*(1) Medir a variável de controle;*

*(2) Aplicar o estado de chaveamento calculado no período*

*de amostragem anterior;*

*(3) Estimar a variável de controle para o instante k + 1;*

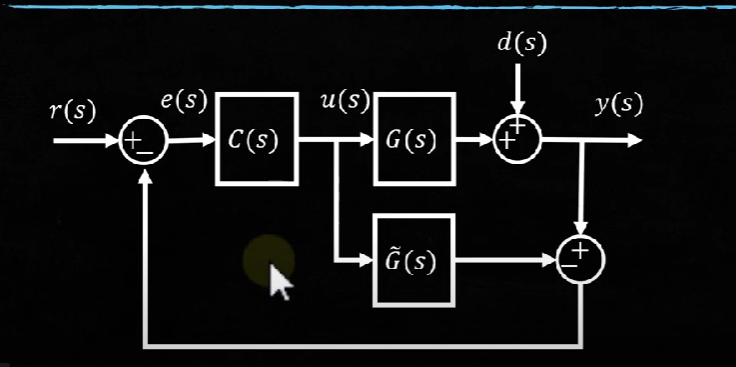
*(4) Prever a variável de controle para o instante k + 2 para todos os estados de chaveamento do conversor;*

*(5) Avaliar a função custo;*

*(6) Escolher o estado de chaveamento que minimiza a função custo o qual será aplicado no próximo período de amostragem.*

### **2.1.1 Método do Modelo Interno**

Modelo Interno

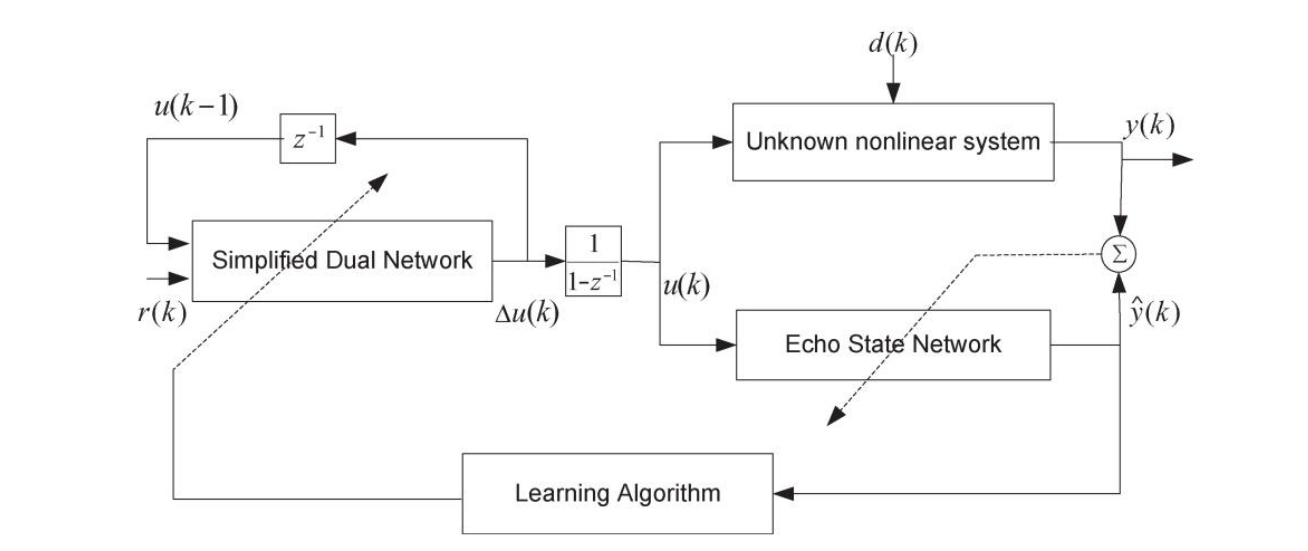


[controlautomaticoeducacion.com](https://controlautomaticoeducacion.com/control-realimentado/imc-internal-model-control/)

No diagrama, o controle do processo é feito através do erro entre o sinal real do sistema e o seu modelo matemático, a fim de obter um controlador tolerante a erros de modelagem. O modelo matemático é obtido a partir da resposta ao degrau do sistema em malha aberta.

### **2.1.1 Redes Neurais Preditivas em sistemas desconhecidos**

No Artigo “***Model Predictive Control of Unknown Nonlinear Dynamical Systems Based on Recurrent Neural Networks***”, utilizado como uma das bases para o controle, foi utilizado duas redes neurais, a rede de estado de eco (ESN) e a rede dupla simplificada (SDN). A rede ESN é utilizada para identificação do sistema e a rede SDN utilizada para otimização. A abordagem baseada em RNN proposta tem muitas propriedades desejáveis como convergência global e baixa complexidade.



## 2.3 Controle inteligente do caminhar de robôs móveis utilizando algoritmos genéticos e redes neurais artificiais

## 2.4 Controle adaptativo de sistemas dinâmicos: uma proposta para o relaxamento da condição de excitação persistente

https://downloads.editoracientifica.org/articles/210705383.pdf

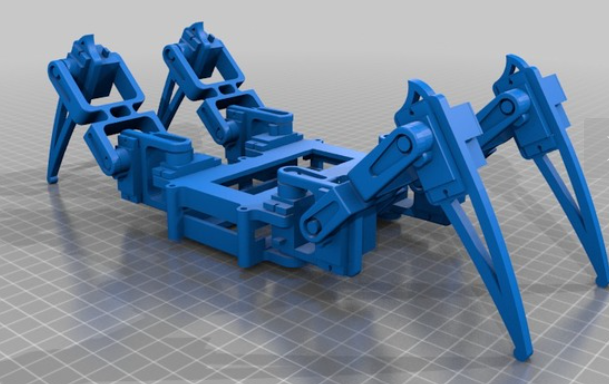
**2.5 Controle de Robô quadrúpede utilizando Redes Neurais**

**2.6 Controle Adaptativo Por Modelo de Referência**

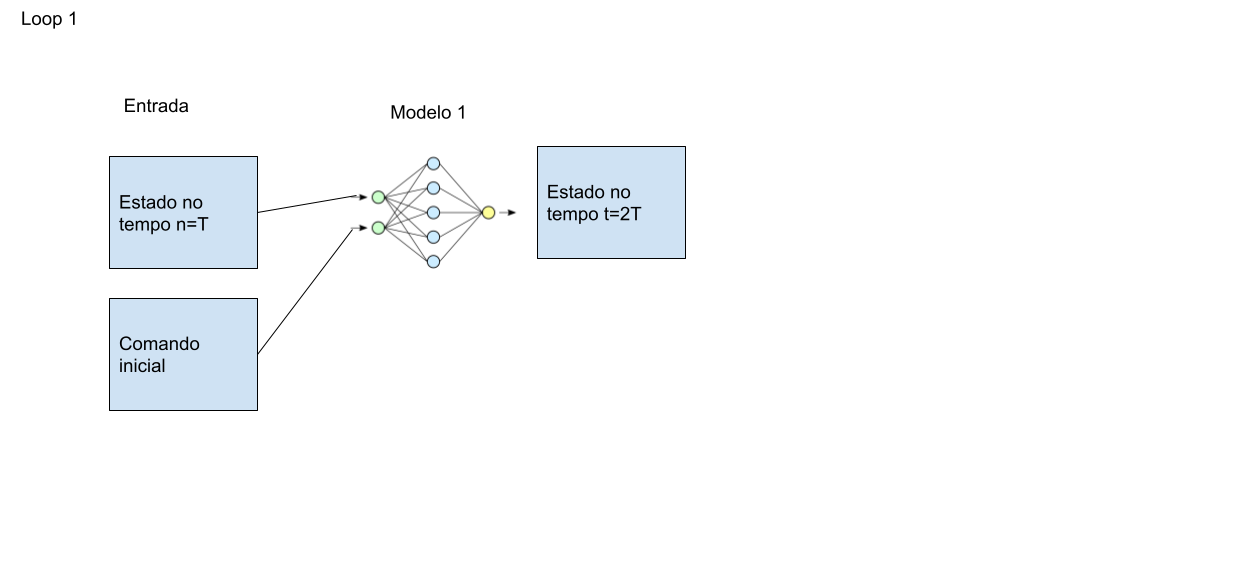
# 3 METODOLOGIA

Para testar e colher os resultados da rede neural preditiva em um modelo não linear, pretende-se utilizar um microcontrolador em um sistema embarcado baseado em um robô quadrúpede:

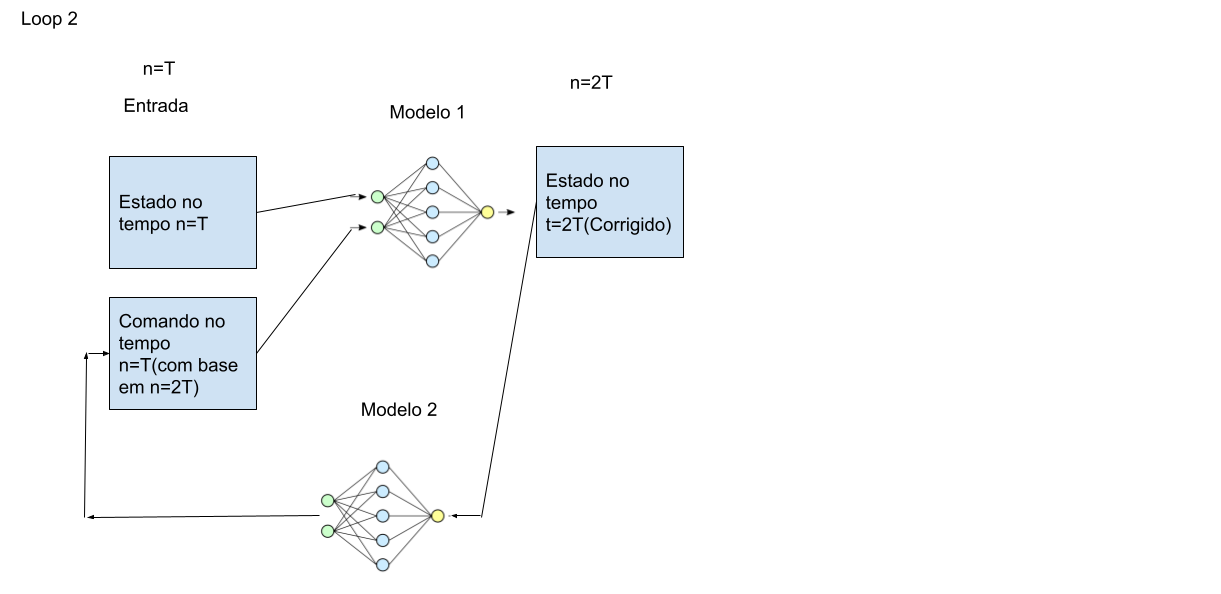
Figura 1: Spider robot, quad robot, quadruped



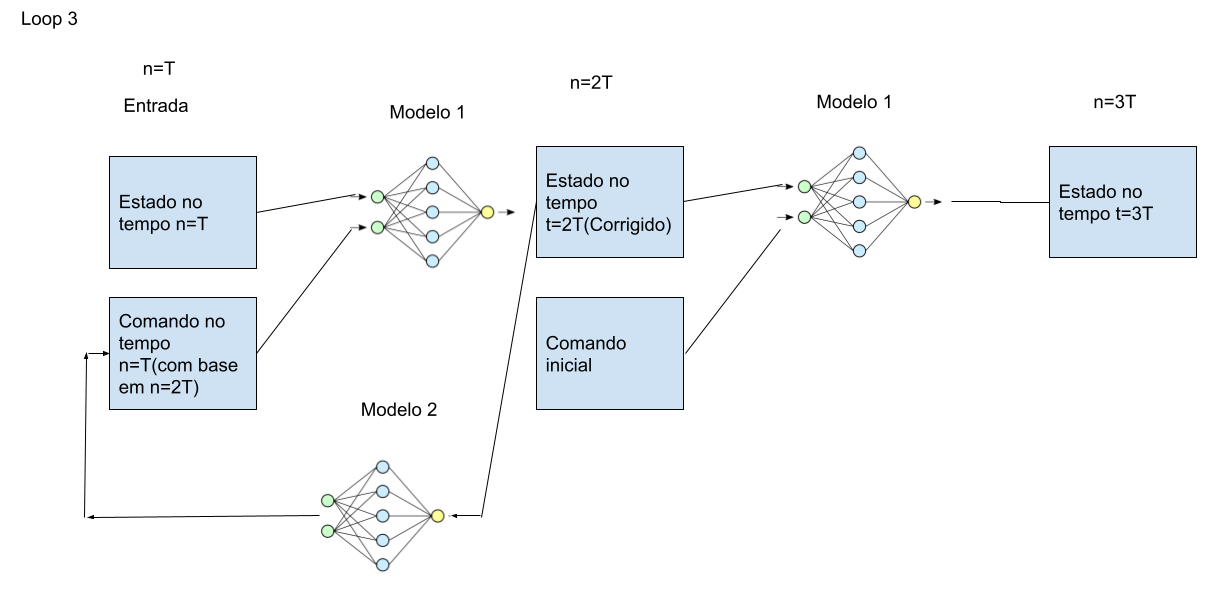
Fonte: <https://www.thingiverse.com/thing:2796820/files>

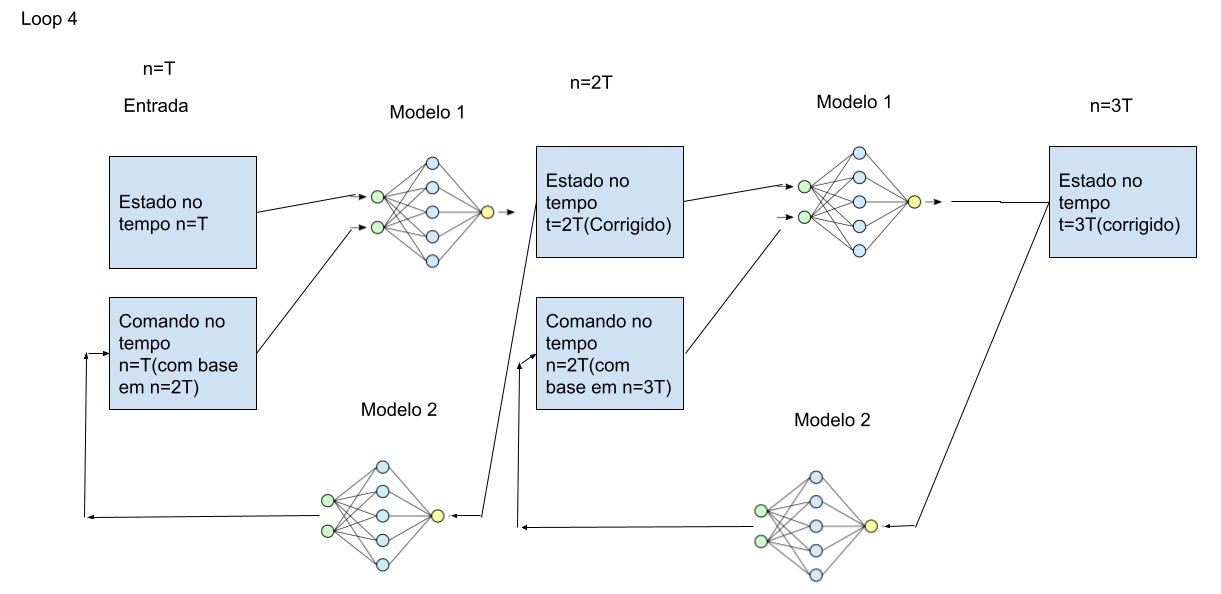


No segundo loop, o modelo 2 irá corrigir os estados, com base na trajetória prevista, por fim o modelo 1 irá prever novamente os estados com base nas alterações dadas nos comandos previstos.



Com isso, repetir o processo para as demais saídas.





A aprendizagem do modelo 1 será feita com base no quão certo ele previu o estado, utilizando um erro médio quadrático entre o vetor E[T], estado no tempo T, e o vetor Eh[T], estado previsto para o tempo T. E a aprendizagem do modelo 2 será feita com base no quanto o estado resultante do comando aproximou-se do estado desejado, em que o erro será baseado em uma pontuação dada de acordo com os objetivos que devem ser atingidos. O resultado deste processo será um vetor contendo a trajetória dos estados do sistema, este vetor é dinâmico, e seus valores são constantemente alterados e corrigidos pelos modelos trabalhando em conjunto.

Erro do modelo 1:

| Vetor resultante | | | | | |
| --- | --- | --- | --- | --- | --- |
| tempo(n) | n=T | n=2.T | n=3T | ... | n=N.T |
| Estados | E1[T] | E1[2T] | E1[3T] | ... | E1[NT] |
| E2[T] | E2[2T] | E2[3T] | ... | E2[NT] |
| E3[T] | E3[2T] | E3(T) | ... | E3(T) |
| E3[T] | E3[2T] | E3[3T] | ... | E3[NT] |
| Comandos | C1[T] | C1[2T] | C1[3T] | ... | C1[NT] |
| C2[T] | C2[2T] | C2[3T] | ... | C2[NT] |
| C3[T] | C3[T] | C3[3T] | ... | C3[NT] |

Modelo 1:

O modelo 1 terá como entrada os estados atuais do robô quadrúpede e os comandos dados no tempo atual e, terá como saída o próximo estado do robô. Considerando como entrada o estado e comandos medidos no tempo n=T, a saída do sistema será o estado no tempo n=(2T), onde T é o intervalo de medições feito pelo microcontrolador.

Modelo 2:

O modelo 2 irá determinar os comandos a serem dados para robô no tempo n=T, de acordo com os estados previsto para o tempo n=(2T) e o estado atual, considerando que o primeiro comando dado será um vetor nulo.

Portanto, o modelo de controle preditivo pode ser simplificado da seguinte maneira:

Os estados e comandos do sistema são:

| Variáveis De Estado | | | | |
| --- | --- | --- | --- | --- |
| Sensores | | | | |
| Nome | Descrição | Range | Tipo | unidade |
| EM1 | Ângulo do Motor 1 | 0-180 | int | graus |
| EM2 | Ângulo do Motor 2 | 0-180 | int | graus |
| EM3 | Ângulo do Motor 3 | 0-180 | int | graus |
| EM4 | Ângulo do Motor 4 | 0-180 | int | graus |
| EM5 | Ângulo do Motor 5 | 0-180 | int | graus |
| EM6 | Ângulo do Motor 6 | 0-180 | int | graus |
| EM7 | Ângulo do Motor 7 | 0-180 | int | graus |
| EM8 | Ângulo do Motor 8 | 0-180 | int | graus |
| EM9 | Ângulo do Motor 9 | 0-180 | int | graus |
| EM10 | Ângulo do Motor 10 | 0-180 | int | graus |
| EM11 | Ângulo do Motor 11 | 0-180 | int | graus |
| EM12 | Ângulo do Motor 12 | 0-180 | int | graus |
| EIX | Inclinação do robô no eixo x | 0-180 | int | graus |
| EIY | Inclinação do robô no eixo y | 0-180 | int | graus |
| EIZ | Inclinação do robô no eixo z | 0-180 | int | graus |
| EAX | Aceleração do robô no eixo x | 0-1 | int | - |
| EAY | Aceleração do robô no eixo y | 0-1 | int | - |
| EAZ | Aceleração do robô no eixo z | 0-1 | int | - |
| EV | Vibração do robô, medido indiretamente através da variação de EIX e EIY | 0-180 | float | graus/período |
| ED | Distância medida pelo sensor ultrassônico | 0-10 | float | metros |
| Comandos | | | | |
| Nome | Descrição | Range | Tipo | unidade |
| CM1 | comanda motor M1 | 0-180 | int | graus |
| CM2 | comanda motor M2 | 0-180 | int | graus |
| CM3 | comanda motor M3 | 0-180 | int | graus |
| CM4 | comanda motor M4 | 0-180 | int | graus |
| CM5 | comanda motor M5 | 0-180 | int | graus |
| CM6 | comanda motor M6 | 0-180 | int | graus |
| CM7 | comanda motor M7 | 0-180 | int | graus |
| CM8 | comanda motor M8 | 0-180 | int | graus |
| CM9 | comanda motor M9 | 0-180 | int | graus |
| CM10 | comanda motor M10 | 0-180 | int | graus |
| CM11 | comanda motor M11 | 0-180 | int | graus |
| CM12 | comanda motor M12 | 0-180 | int | graus |
| Critérios de desempenho do controle utilizados como treinamento do modelo 2(Modelo Preditivo) | | | | |
| Nome | Descrição | | Tipo | unidade |
| D\_TS | Tempo de subida medido em número de períodos | | float | T |
| D\_TA | Tempo de acomodação medido em número de períodos | | float | s |
| D\_E | Erro em regime permanente | | float | % |
| D\_EC | Esforço de controle | | float | % |

## 3.1 CRONOGRAMA

| **TAREFA** | **INÍCIO** | **TÉRMINO** |
| --- | --- | --- |
| **TAREFAS PRELIMINARES** | Maio | Junho |
| Aquisição dos Materiais | 1/5/22 | 15/5/22 |
| Montagem do robô | 15/5/22 | 29/5/22 |
| Montagem do circuito | 30/5/22 | 13/6/22 |
| Teste e validação do sistema | 14/6/22 | 28/6/22 |
| Estudo e Modelagem do Sistema | 14/6/22 | 28/6/22 |
| **ESTUDO E MODELAGEM DO SISTEMA** | Junho | Julho |
| Coleta de Dados do Comportamento do Sistema frente a diferentes entradas | 29/6/22 | 9/7/22 |
| Definição de um modelo matemático do Sistema | 10/7/22 | 20/7/22 |
| Estudo do modelo matemático utilizando ferramentas computacionais | 20/7/22 | 30/7/22 |
| **MODELAGEM DAS REDES NEURAIS** | Julho | Setembro |
| Descrição do sistema de controle e o algoritmo a ser utilizado | 30/7/22 | 6/8/22 |
| Definição dos Critérios de Desempenho | 7/8/22 | 14/8/22 |
| Definição dos modelos de Rede Neural | 15/8/22 | 22/8/22 |
| Algoritmo de propagação e retropropagação | 23/8/22 | 30/8/22 |
| Treinamento e Testes das Redes Neurais em Ambiente Simulado | 31/8/22 | 7/9/22 |
| **APLICAÇÃO DAS REDES NEURAIS NO ROBÔ QUADRUPEDE** | Setembro | Outubro |
| Aplicação das Redes Neurais pré treinadas no robô quadrupede | 8/9/22 | 15/9/22 |
| Análise dos Critérios de Desemepenho Na Rede Pré Treinada | 14/9/22 | 21/9/22 |
| Continuidade do Treinamento no Ambiente Real | 21/9/22 | 28/9/22 |
| Análise de Desempenho e Validação do Sistema | 28/9/22 | 5/10/22 |
| **Otimização Das Redes Neurais** | Outubro | Outubro |
| Técnicas para aperfeiçoamento das Redes Neurais | 6/10/22 | 13/10/22 |
| Interpretação e conclusão dos resultados | 13/10/22 | 20/10/22 |
| Registro de Desempenho | 21/10/22 | 28/10/22 |

## 

## 3.2 RECURSOS

### **3.2.1 Materiais**

Os componentes utilizados para este projeto são divididos em atuadores, sensores e outros, como descritos na tabela a seguir:

| Lista de Materiais | | | | |
| --- | --- | --- | --- | --- |
| Atuadores | | | | |
| Número | Componente | Modelo | Quantidade | Função |
| 1 | Servo Motor |  | 12 | Controle do ângulo das patas |
| Sensores | | | | |
| Número | Componente | Modelo | Quantidade | Função |
| 2 | Sensor Ultrassônico |  | 1 | Medição da distância linear de objetos no caminho |
| 3 | Módulo Sensor de Inclinação | MPU6050 | 1 | Medir a Inclinação e aceleração em x, y e z do sistema |
| Auxiliares | | | | |
| Número | Componente | Modelo | Quantidade | Função |
| 4 | Arduino Mega | - | 1 | Utilizado para controle do Sistema |
| 5 | Jumpers | - | 28 | Utilizado para conexão dos materiais |
| 6 | Bateria | - | 1 | Para alimentar o sistema e diminuir as variações de tensão da fonte |
| 7 | Fonte 5v-3A | - | 1 | Alimentar o sistema |
| 8 | Filamento de impressão 3D | - | 200g | Para impressão das peças |
| 9 | Espaguete termo retrátil | - | 1M | Para isolar o circuito |
| 10 | Estanho para Solda | - | 50g | Para soldar os componentes |
|  |  |  |  |  |

Para fazer o controle, foi definido a utilização do Arduino Mega, pela praticidade, custo e quantidade de saídas digitais pwm. A fonte a ser utilizada foi dimensionada pelo consumo de corrente do circuito. Utilizando uma fonte de bancada e um multímetro, foi aplicada uma tensão de 5V no circuito e os motores foram comandados para ângulos aleatórios a fim de definir qual o pico de consumo do sistema. O resultado foi que o circuito teve um consumo de 3A.

### **3.2.2 Ferramentas computacionais**

Para auxiliar nos testes e modelagem, tanto do sistema a ser controlado quanto dos modelos de redes neurais, foram utilizados as seguintes ferramentas:

* Jupyter Notebook: Utilizado para desenvolvimento das redes neurais a serem utilizadas no projeto
* Visual Studio Community: Programação do Arduino e simulação das Redes Neurais
* Mathworks MATLAB R2020a: Utilizado para simular o modelo matemático e obter os critérios de desempenho idealizados para o projeto
* Tinkercad: Utilizado para testar o código em c++ a ser utilizado no arduino mega

### **3.2.3 Equipamentos Utilizados**

Para montagem e testes, foram utilizados os seguintes equipamentos:

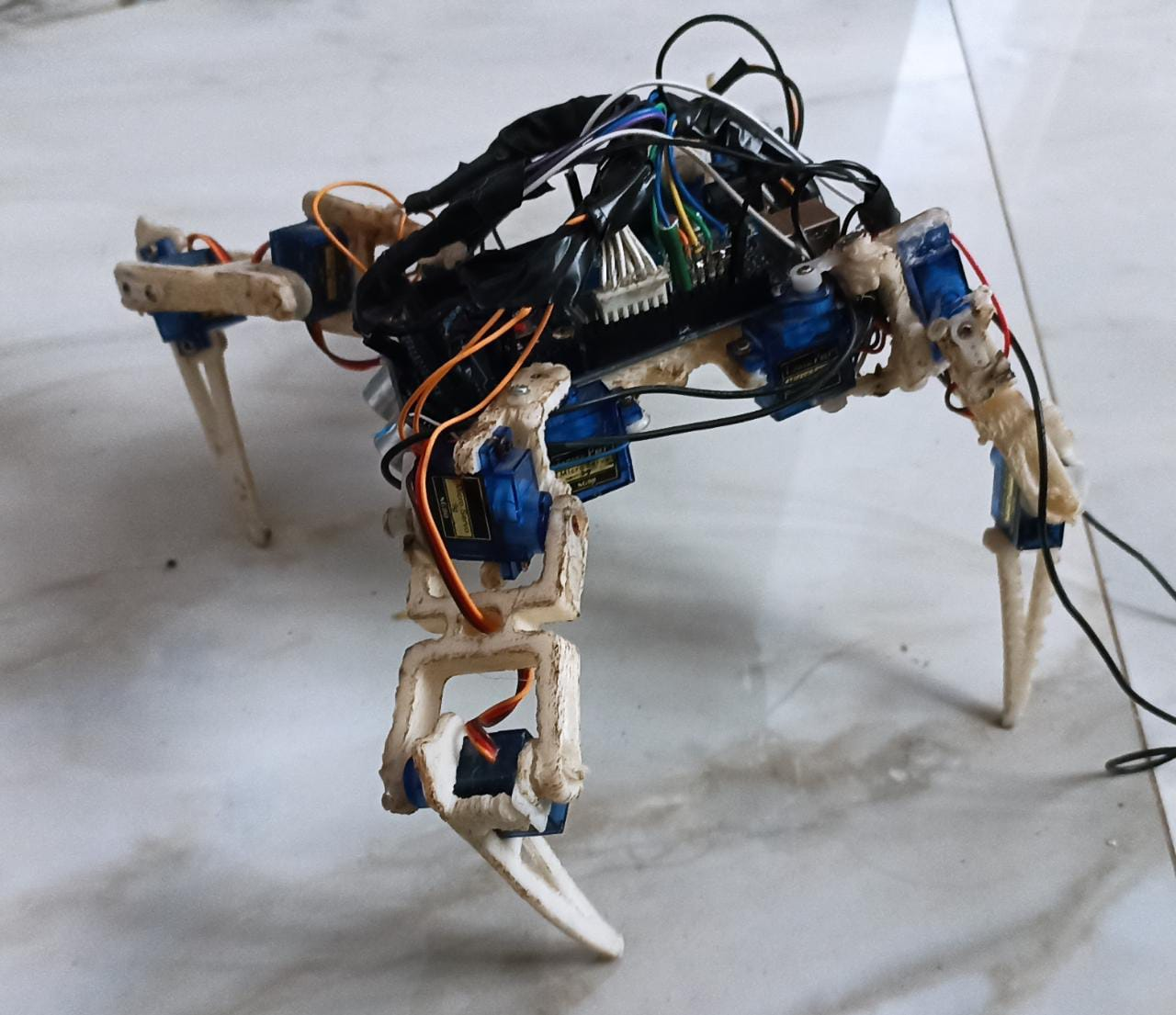
* Multímetros;
* Osciloscópio;
* Ferro de solda;
* Computador;
* Fonte de bancada;

## 3.3 MONTAGEM DO SISTEMA

### **3.3.1 Modelo em 3D**

O modelo em 3D foi obtido no site thingiverse, criado pelo usuário Monhambe em 16 de Fevereiro de 2018. Foi utilizada uma impressora 3D para obter as peças do sistema a ser controlado, na figura 2, que foi montado utilizando parafusos para fixar os servos motores. Por fim, os componentes foram conectados ao controlador com o auxílio de jumpers e de solda com estanho

FIGURA 2 - Robô Quadrúpede Montado



Após a Montagem, foram feitos testes de continuidade e a alimentação do sistema, para testar os componentes e validar a montagem do sistema.

## 3.4 Estudo e Modelagem do Sistema

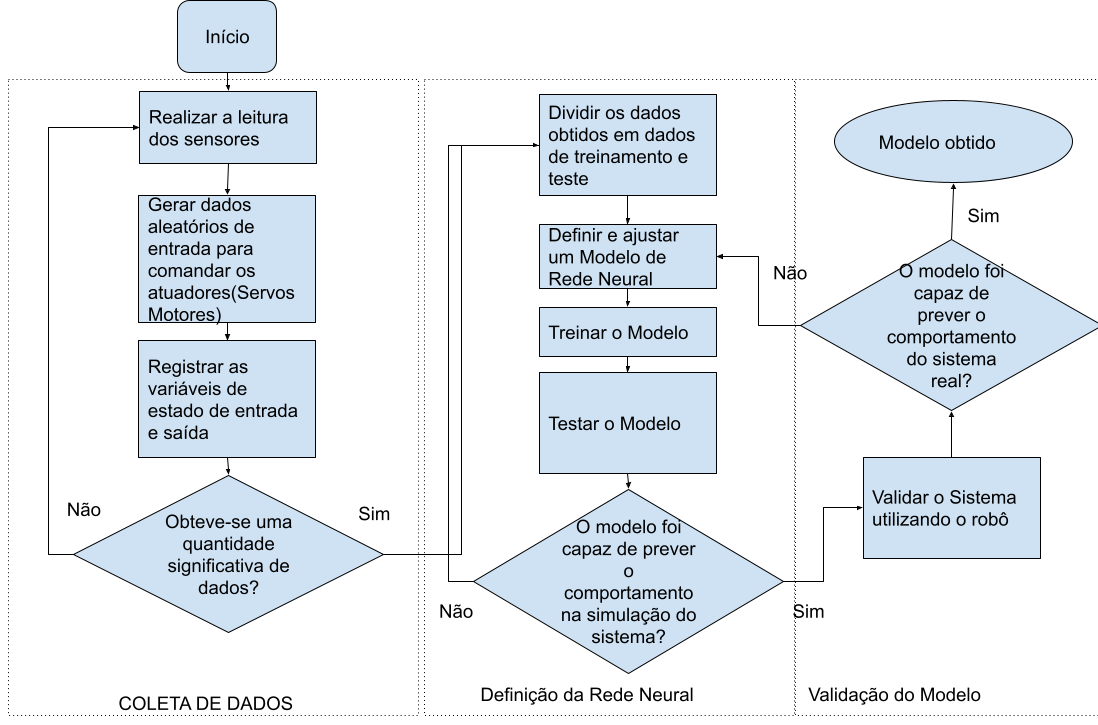
Para modelar um sistema é preciso descrever matematicamente o seu comportamento de forma a prever os estados de suas variáveis em função de perturbações e da própria dinâmica existente no sistema, isto é cria um bloco capaz de relacionar de forma aproximada uma entrada e uma saída.

Para o robô quadrúpede, pode-se considerar como entrada, os estados dos sensores e comandos a serem dados aos atuadores. Os estados são os ângulos atuais dos motores, a inclinação nos três eixos(x,y e z), a aceleração nos três eixos, e a distância medida pelo sensor ultrassônico. Os comandos dados são os ângulos inseridos na função Servo.write(), da biblioteca “Servo.h”, que determina para qual ângulo o motor deverá se mover.

Como saída, pode-se considerar os estados resultantes, ou seja, a inclinação, a aceleração e a distância da próxima leitura do sistema.

A técnica de modelagem utilizada é descrita no fluxograma a seguir:.

FIGURA 3: Modelagem do Sistema



Com esta técnica, pretende-se obter um modelo capaz de representar matematicamente o comportamento do sistema, prevendo os estados futuros com base nas variáveis de entrada do sistema com o menor erro possível. O modelo preditivo resultante será a base para a construção da rede neural que controlará o sistema, com base no erro previsto pelo modelo obtido neste método.

### **3.4.1 Coleta de Dados do Comportamento do Sistema frente a diferentes entradas**

Falar sobre como foi feita a coleta de dados, quais dados foram coletados, descrever o algoritmo e qual a importância de cada dado para a modelagem.

Falar sobre Persistência de excitação.

Descrever o sinal de entrada no sistema.

Descrever o sinal de saída do sistema.

Registrar graficamente os resultados.

### **3.4.2 Definição de um modelo matemático do Sistema Utilizando Redes Neurais**

Introduzir o método a ser utilizado para obtenção do modelo matemático

Utilização do matlab para obter a forma de onda e resposta ao impulso

### **3.4.3 Estudo do modelo matemático utilizando ferramentas computacionais**

Obter as respostas do modelo a diferentes entradas

Testar o modelo no simulador

Registrar as características do sistema de forma gráfica e conceitual;

## 3.5 Síntese do Controlador Neural

### **3.5.1 Descrição do sistema de controle e o algoritmo a ser utilizado**

Utilizando o modelo criado no item anterior, descrever o controlador ideal e características do controlador utilizando teorias de controle por redes neurais.

Descrever o algoritmo a ser utilizado para pré treino;

Descrever o algoritmo a ser utilizado para controle e tempo em ambiente real;

### **3.5.2 Definição dos Critérios de Desempenho**

Definir os critérios de desempenho a serem atingidos;

### **3.5.3 Definição dos modelos de Rede Neural**

Definir os modelos de redes neurais, tipos de camadas e suas características com base nos resultados

### **3.5.4 Algoritmo de propagação e retropropagação**

Descrever de forma matemática o erro a ser utilizado da saída das redes neurais

Descrever de forma matemática a propagação da rede neural e a retropropagação do erro.

Construção dos algoritmos de propagação e retropropagação em python

### **3.5.5 Análise de Desempenho e Validação do Sistema**

Analisar as redes neurais e o aprendizado no modelo computacional;

Validar o aprendizado;

### **3.5.5 Algoritmo genético para seleção da Rede Neural**

Descrever como o algoritmo será utilizado para seleção da melhor rede neural, com melhor estrutura para alcançar os objetivos.

## 3.6 APLICAÇÃO DAS REDES NEURAIS NO ROBÔ QUADRÚPEDE

### **3.6.1 Aplicação das Redes Neurais pré treinadas no robô quadrúpede**

Utilizar a rede neural que teve o melhor desempenho no simulador para construir a rede neural no robô quadrúpede, utilizando os pesos do pré treinamento

### **3.6.2 Análise dos Critérios de Desempenho Na Rede Pré Treinada**

Descrever os resultados obtidos ao longo do treinamento e medições no robô em simulação e ajustar os critérios de desempenho para o ambiente real

### **3.6.3 Continuidade do Treinamento no Ambiente Real**

Iniciar o treinamento no ambiente

### **3.6.4 Teste no Ambiente Real**

Colocar o robô quadrúpede em diferentes ambientes e analisar a capacidade de adaptação/

## 3.7 OTIMIZAÇÃO DAS REDES NEURAIS

### **3.7.1 Técnicas para aperfeiçoamento das Redes Neurais**

Descrever como serão feitos os ajustes finais para melhorar os resultados obtidos

### **3.7.2 Interpretação e conclusão dos resultados da otimização**

### **3.7.3 Registro de Desempenho**

Registrar os gráficos de desempenho ao longo do treinamento, tanto no simulador quanto no ambiente real

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 4.0 RESULTADOS E DISCUSSÃO

# 5.0 CONCLUSÃO

# 6.0 REFERÊNCIAS

[MODELAGEM PREDITIVA DE LINHA DE COSTA UTILIZANDO REDES NEURAIS ARTIFICIAIS](https://www.scielo.br/j/bcg/a/DJGyBq7LJm39Ggmj5KypQ4P/?format=pdf&lang=pt)

[Análise preditiva para identificação de alunos suscetíveis à evasão escolar Análise preditiva para identificação de alunos suscetíveis à evasão escolar](https://scholar.archive.org/work/eq6wfaftlnhyfovndjyp5htx7a/access/wayback/https://brazilianjournals.com/index.php/BRJD/article/download/33014/pdf)

[AERONAVES CONFIGURADAS POR CONTROLE DO TIPO PREDITIVO NEURAL](http://mtc-m16.sid.inpe.br/col/sid.inpe.br/jeferson/2003/05.14.14.55/doc/publicacao.pdf)

[Arquitetura neural para controle da locomoção de um robô quadrúpede baseada em referências biológicas](https://www.lume.ufrgs.br/handle/10183/8277)

[O que são sistemas embarcados](https://www.oficinadanet.com.br/post/13538-o-que-sao-sistemas-embarcados#:~:text=O%20sistema%20embarcado%2C%20tamb%C3%A9m%20chamado,ao%20sistema%20que%20ele%20controla.&text=Atualmente%2C%20v%C3%A1rios%20produtos%20possuem%20uma,nosso%20dia%2Da%2Ddia).

<https://materialpublic.imd.ufrn.br/curso/disciplina/1/63/5/5>

<https://sci-hub.se/10.1109/TIE.2011.2169636>

GONÇALVES, RODRIGO MIKOSZ;. **MODELAGEM PREDITIVA DE LINHA DE COSTA UTILIZANDO REDES NEURAIS ARTIFICIAIS.** Universidade Federal de Pernambuco (UFPE) Centro de Tecnologia e Geociências – CTG, Departamento de Engenharia. Disponível em <https://www.scielo.br/j/bcg/a/DJGyBq7LJm39Ggmj5KypQ4P/?format=pdf&lang=pt>

#include <Ultrasonic.h>

class EstadoAtual{

public:

long int tempo;//Marca o tempo do estado

int EM[12];// Marca o Estado dos 12 motores

float EIX;//Marca a inclinacao no eixo X

float EIY;//

float EIZ;//

float EAX;//Mostra a aceleracao atual do robo no eixo X

float EAY;

float EAZ;

float ED;// Marca a distancia lida pelo sensor ultrassonico

int CM[12];//Marca o comando atual dado ao motor

};

class vetorEstados{

private:

EstadoAtual historico[1000];

public:

};

class modoColetaDados{

void setup(){

Serial.println("Modo coleta de dados iniciado");

Serial.println("Exibe vetor de estados coletado")

}

void loop(){

Serial.println("1: Atualizando Sensores");

Serial.println("2: Gerando comandos aleatorios");

Serial.println("3: Gravando Estados Atuais na ultima linha do arquivo");

Serial.println("4: Exibindo Estado atual");

}

};

**FIGURAS**

| Figura 1 - | Palavra | 09 |
| --- | --- | --- |
| Figura 2 - | Palavra palavra palavra | 09 |
| Figura 3 - | Palavra palavra palavra palavra palavra palavra palavra palavra palavra | 10 |
| Figura 4 - | Palavra palavra palavra palavra palavra | 10 |
| Figura 5 - | Palavra palavra palacra palavra palavra | 11 |
| Figura 6 - | Palavra palavra palavra palavra palavra | 11 |
|  |  |  |
|  |  |  |
|  |  |  |

**LISTAS DE TABELAS**

| Tabela 1 - | Modelos palavra palavra palavra | 08 |
| --- | --- | --- |
| Tabela 2 - | Modelos palavra palavra palavra | 10 |
| Tabela 3 - | Modelos palavra palavra palavra | 12 |
| Tabela 4- | Palavra palavra palavra palavra palavra | 12 |
| Tabela 5 - | Palavra palavra palavra palavra palavra | 12 |

**SUMÁRIO**

[**1. INTRODUÇÃO**](#_heading=h.30j0zll) **7**

[1.1. Justificativa](#_heading=h.929cqt8lanqa) 13

[1.2. Colocação do Problema](#_heading=h.9yna9y22jjw9) 13

[1.3. Objetivos](#_heading=h.bf58co3a1yur) 13

[1.3.1 Objetivos impostos ao robô quadrúpede](#_heading=h.1v3knfpmw6mi) 14

[1.4 Organização do Trabalho](#_heading=h.c7ia4btid0c8) 15

[**2 FUNDAMENTAÇÃO TEÓRICA**](#_heading=h.8eh42d4xsvzp) **16**

[2.1 Métodos de controle](#_heading=h.u157zwgptka0) 17

[2.1.1 Controle Preditivo](#_heading=h.et4xb9jo7x2e) 17

[2.1.1 Método do Modelo Interno](#_heading=h.wrqgri3ani0g) 18

[2.3 Redes Neurais Artificiais](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.gsem6jwpz6l) [18](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.gsem6jwpz6l)

[2.1.1 Redes Neurais Preditivas em sistemas desconhecidos](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.9gs8ob4h74e) [18](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.9gs8ob4h74e)

[**3 METODOLOGIA**](#_heading=h.vcpem2ehjj08) **19**

[3.1 CRONOGRAMA](#_heading=h.w3lxgd28ao5o) 19

[3.2 RECURSOS](#_heading=h.2p4c0b2li65x) 21

[3.2.1 Materiais](#_heading=h.hjgp3197ry5c) 21

[3.2.2 Ferramentas computacionais](#_heading=h.5houcbcosbds) 21

[3.2.3 Equipamentos Utilizados](#_heading=h.r54y6zeds2ol) 21

[3.3 MONTAGEM DO SISTEMA](#_heading=h.widr0fk1n63x) 21

[3.3.1 Modelo em 3D](#_heading=h.mov5j0u6o6df) 21

[3.3.2 Passo a Passo de Montagem](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.yufzyvcmisfl) [21](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.yufzyvcmisfl)

[3.3.3 Testes e validação da montagem](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.lu4z3jcjl6el) [21](https://docs.google.com/document/d/1BvH2Hhew99n9rx3GxqIbVaitTK1J2ZCI/edit#heading=h.lu4z3jcjl6el)

[3.4 Estudo e Modelagem do Sistema](#_heading=h.3fb78m4wddri) 21

[3.4.1 Coleta de Dados do Comportamento do Sistema frente a diferentes entradas](#_heading=h.4olrxm5qwoaq) 21

[3.4.2 Definição de um modelo matemático do Sistema](#_heading=h.23c9lu23cy7x) 21

[3.4.3 Estudo do modelo matemático utilizando ferramentas computacionais](#_heading=h.ie7iiqz4337m) 21

[3.5 MODELAGEM DAS REDES NEURAIS](#_heading=h.2dj00oypwm8p) 21

[3.5.1 Descrição do sistema de controle e o algoritmo a ser utilizado](#_heading=h.3strcerbjq8g) 21

[3.5.2 Definição dos Critérios de Desempenho](#_heading=h.l8g56awxsezp) 21

[3.5.3 Definição dos modelos de Rede Neural](#_heading=h.3wgxyvse57v8) 21

[3.5.4 Algoritmo de propagação e retropropagação](#_heading=h.pbab29z7cz7z) 21

[3.5.5 Análise de Desempenho e Validação do Sistema](#_heading=h.cqlyzt26n27i) 22

[3.6 APLICAÇÃO DAS REDES NEURAIS NO ROBÔ QUADRUPEDE](#_heading=h.9op8jq3g7bcy) 22

[3.6.1 Aplicação das Redes Neurais pré treinadas no robô quadrúpede](#_heading=h.nqwxtcaphxc) 22

[3.6.2 Análise dos Critérios de Desemepenho Na Rede Pré Treinada](#_heading=h.dgwljg9w8504) 22

[3.6.3 Continuidade do Treinamento no Ambiente Real](#_heading=h.56pv3xsge7v5) 22

[3.6.4 Continuidade do Treinamento no Ambiente Real](#_heading=h.ec6hsidcyksv) 22

[3.7 OTIMIZAÇÃO DAS REDES NEURAIS](#_heading=h.3zmpbn9oi03v) 22

[3.7.1 Técnicas para aperfeiçoamento das Redes Neurais](#_heading=h.e6cyy5ro1pf) 22

[3.7.2 Interpretação e conclusão dos resultados da otimização](#_heading=h.i1bmu5e4r2ld) 22

[3.7.3 Registro de Desempenho](#_heading=h.n47jv8ortjg7) 22

[**4.0 RESULTADOS E DISCUSSÃO**](#_heading=h.nnlsqdp3ej3z) **22**

[**5.0 CONCLUSÃO**](#_heading=h.xmzngkpkf87i) **22**

[**6.0 REFERÊNCIAS**](#_heading=h.1t3h5sf) **23**

# 1. INTRODUÇÃO

A utilização de robôs com controlador embutido, ou como são conhecidos, os sistemas embarcados, é crescente, e atualmente existem diversas tentativas de fazer com que alcancem a inteligência artificial, para que sejam capazes de analisar e aprender com o seu ambiente para atingir um determinado objetivo. Neste trabalho, pretende-se investigar e aplicar técnicas de modelagem de redes neurais a fim de produzir um robô quadrúpede que seja capaz de realizar as seguintes ações:

* Prever as consequências de suas ações, ou seja predizer os estados futuros de seus sensores com base nas decisões tomadas.
* Comandar os motores a fim de atingir um objetivo(Permanecer em equilíbrio e evitar obstáculos)

Dessa forma, este trabalho terá como foco principal desenvolver um modelo de inteligência artificial em um sistema embarcado, mostrado na imagem abaixo, investigar os resultados obtidos e documentar o desempenho do controlador preditivo para tal aplicação.

## 1.1. Justificativa

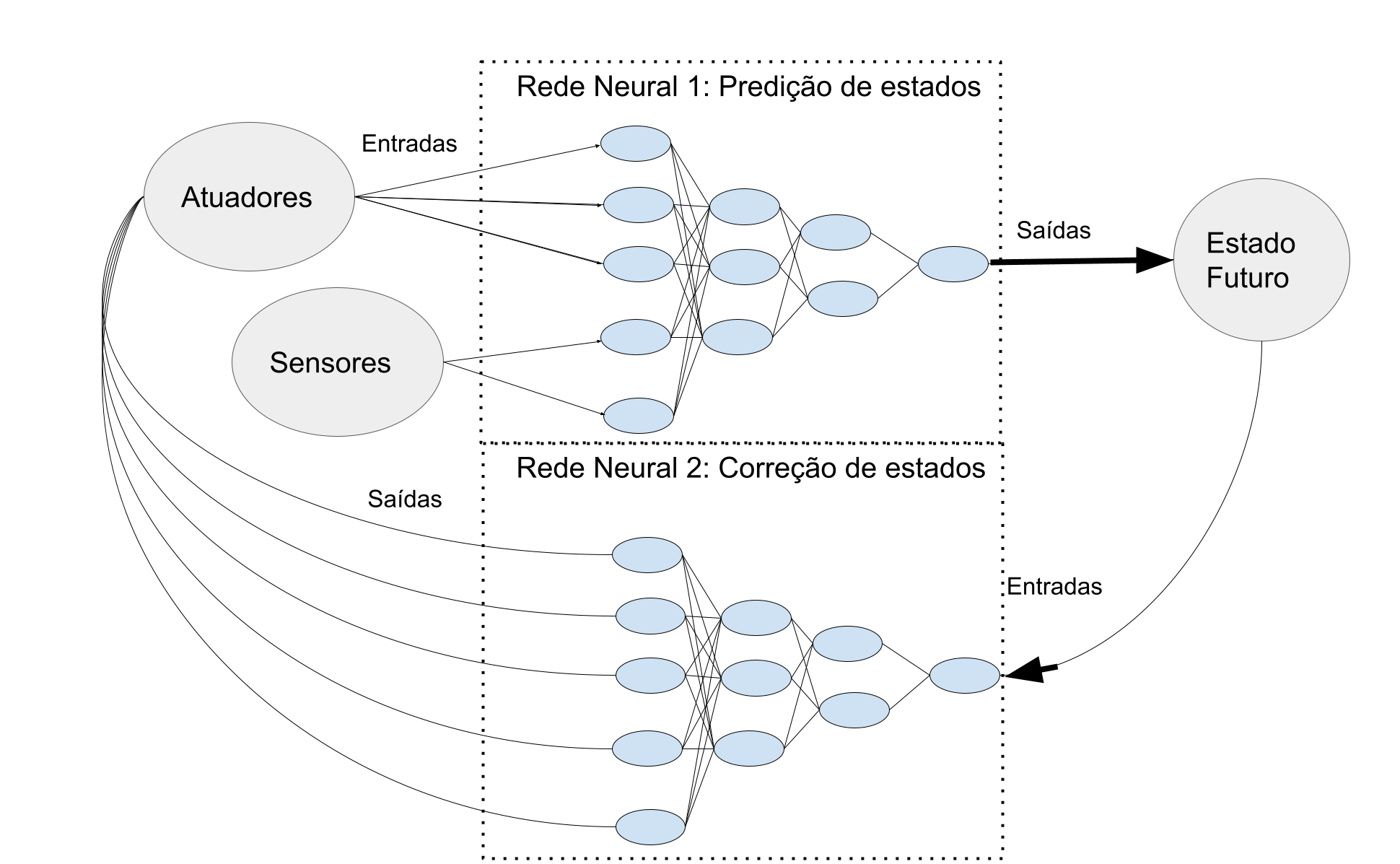
Reproduzir técnicas de controle, associadas a redes neurais artificiais, para projetar e aplicar um controlador não linear, tal que seja capaz de prever o comportamento do robô quadrúpede com base no estado atual e nas escolhas tomadas pelo robô. A utilização de redes neurais para este problema é devido ao fato de serem naturalmente adaptativas, o que as tornam essencialmente vantajosas em relação ao controlador convencional.

## 1.2. Colocação do Problema

Pretende-se projetar um controlador baseado em redes neurais para solucionar o problema descrito e avaliar o desempenho do robô ao longo de seu treinamento utilizando o microcontrolador Arduino Mega.

## 1.3. Objetivos

O trabalho está voltado para a construção de dois modelos de redes neurais, responsáveis pelo controle do robô. Um modelo capaz de prever os próximos N estados, com base no estado inicial dos sensores e um modelo capaz de prever os próximos comandos a serem dados para atingir os objetivos impostos no ítem 1.3.1.



Com isso, pode-se resumir o objetivo do trabalho em testar a seguinte característica do sistema que define comportamento de uma inteligência artificial:

“A aptidão ou capacidade de um dispositivo para desempenhar funções que são normalmente associadas à inteligência humana, tais como raciocínio, aprendizagem e auto-aperfeiçoamento”.

http://www.ppgee.eng.ufba.br/teses/3fa84f4664183cab6c0c9022d582c762.pdf

### **1.3.1 Objetivos impostos ao robô quadrúpede**

O robô tem os seguintes objetivos:

I: Manter-se em equilíbrio, mesmo que hajam perturbações ;

II: Minimizar as vibrações e impactos, a fim de preservar as conexões, ou seja, manter a aceleração nos três eixos o mais próximo de zero possível ;

III:Manter-se virado para o lado que não haja obstáculos, ou seja, sempre que detectar um objeto em uma distância menor do que a distância máxima DM, ele deverá virar o sensor para outra direção.

IV: Prever as consequências de suas ações, ou seja, estimar qual serão os estados resultantes dos comandos dados aos motores para os próximos N passos.

## 1.4 Organização do Trabalho

O Trabalho está dividido nos seguintes temas:

**I- Fundamentação Teórica:** Revisão Bibliográfica acerca de trabalhos anteriores envolvendo Controle Preditivo, Redes Neurais e sistemas embarcados baseados que se aproximam do robô utilizado nos testes. Introdução aos assuntos contidos no trabalho.

**II- Cronograma:** Apresenta as datas limites para realização de cada tarefa, na forma de tabela a partir do início do desenvolvimento até a defesa do TCC.

**III- Descrição e modelagem do Sistema a ser controlado:** Descreve de forma matemática e conceitual, o sistema embarcado a ser controlado e o ambiente ao qual é submetido, além disso, justifica as escolhas de cada componente com base nas variáveis que devem ser medidas e controladas pelo microcontrolador utilizado. Também define as entradas e saídas do modelo.

**V- Descrição do sistema de controle e o algoritmo a ser utilizado:** Descreve a lógica a ser utilizada pelo algoritmo, além do microcontrolador a ser utilizado, baseado no sistema descrito no item III.

**VI- Definição dos Critérios de Desempenho:** Descreve os critérios a serem analisados ao longo do treinamento de forma a possibilitar a comparação dos resultados obtidos com os de controladores convencionais utilizados em sistemas lineares.

**VII - Definição dos modelos de Rede Neural:** Define de forma lógica, possível modelo a ser utilizado para rede neural, tanto para predição de estados, quanto para correção de estados.

**VIII- Algoritmo de propagação e retropropagação:** Descrição matemática da relação entre a entrada e saída de cada modelo, definição do erro a ser utilizado no treinamento e atualização dos pesos de cada camada na retropropagação do erro.

**IX- Treinamento da rede neural:** Descrição do Treinamento do modelo e da obtenção dos dados.

**X- Teste da rede neural:** Valida o modelo com valores colhidos do sistema.

**XI- Montagem do sistema Embarcado e Aplicação dos modelos de treinamento e teste:** Descreve a forma de construção, conexão e utilização de todo o sistema.

**XII - Análise dos critérios de desempenho:** Analisa os valores de forma gráfica, a respeito do desempenho da rede neural utilizada a fim de validar o sistema.

**XIII - Otimização da rede neural:** Aperfeiçoamento da rede neural após a sua validação.

**XIV- Interpretação e conclusão dos resultados:** Comparação, descrição e detalhamento dos resultados obtidos.

# 2 FUNDAMENTAÇÃO TEÓRICA

A teoria utilizada para compreender as técnicas utilizadas para compreensão deste trabalho está nos tópicos a seguir:

* Redes Neurais Artificiais preditivas
* Método do Modelo Interno

## 2.1 Métodos de controle

### **2.1.1 Controle Preditivo**

O controle preditivo utiliza técnicas para que seja possível prever o comportamento de um determinado sistema a partir de um modelo matemático, este modelo pode ser tanto uma Rede Neural quanto um modelo matemático da saída em relação às variáveis de estado. As vantagens deste tipo de Controle é poder prever o erro de determinada a ação e tomar a melhor ação de controle a partir do erro previsto, portanto não é preciso que o erro aconteça.

No artigo ***Controle Preditivo Baseado em Modelo para Conversores Formadores de Rede com Operação,***é feito o controle por chaveamento de forma preditiva para minimizar o erro em Conversores seguindo os seguintes passos:

*(1) Medir a variável de controle;*

*(2) Aplicar o estado de chaveamento calculado no período*

*de amostragem anterior;*

*(3) Estimar a variável de controle para o instante k + 1;*

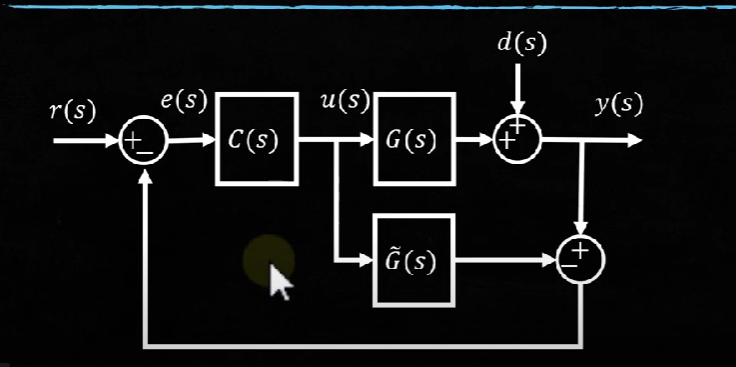
*(4) Prever a variável de controle para o instante k + 2 para todos os estados de chaveamento do conversor;*

*(5) Avaliar a função custo;*

*(6) Escolher o estado de chaveamento que minimiza a função custo o qual será aplicado no próximo período de amostragem.*

### **2.1.1 Método do Modelo Interno**

Modelo Interno

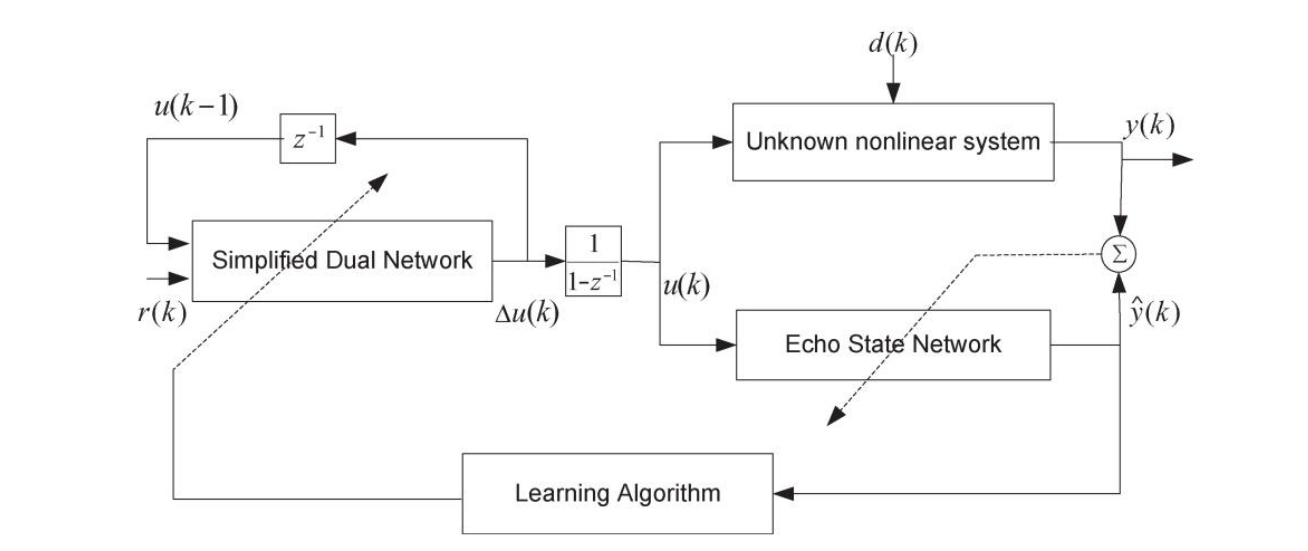


[controlautomaticoeducacion.com](https://controlautomaticoeducacion.com/control-realimentado/imc-internal-model-control/)

No diagrama, o controle do processo é feito através do erro entre o sinal real do sistema e o seu modelo matemático, a fim de obter um controlador tolerante a erros de modelagem. O modelo matemático é obtido a partir da resposta ao degrau do sistema em malha aberta.

### **2.1.1 Redes Neurais Preditivas em sistemas desconhecidos**

No Artigo “***Model Predictive Control of Unknown Nonlinear Dynamical Systems Based on Recurrent Neural Networks***”, utilizado como uma das bases para o controle, foi utilizado duas redes neurais, a rede de estado de eco (ESN) e a rede dupla simplificada (SDN). A rede ESN é utilizada para identificação do sistema e a rede SDN utilizada para otimização. A abordagem baseada em RNN proposta tem muitas propriedades desejáveis como convergência global e baixa complexidade.



## 2.3 Controle inteligente do caminhar de robôs móveis utilizando algoritmos genéticos e redes neurais artificiais

## 2.4 Controle adaptativo de sistemas dinâmicos: uma proposta para o relaxamento da condição de excitação persistente

https://downloads.editoracientifica.org/articles/210705383.pdf

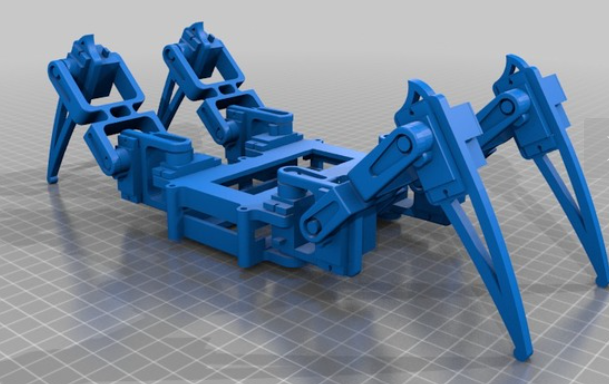
**2.5 Controle de Robô quadrúpede utilizando Redes Neurais**

**2.6 Controle Adaptativo Por Modelo de Referência**

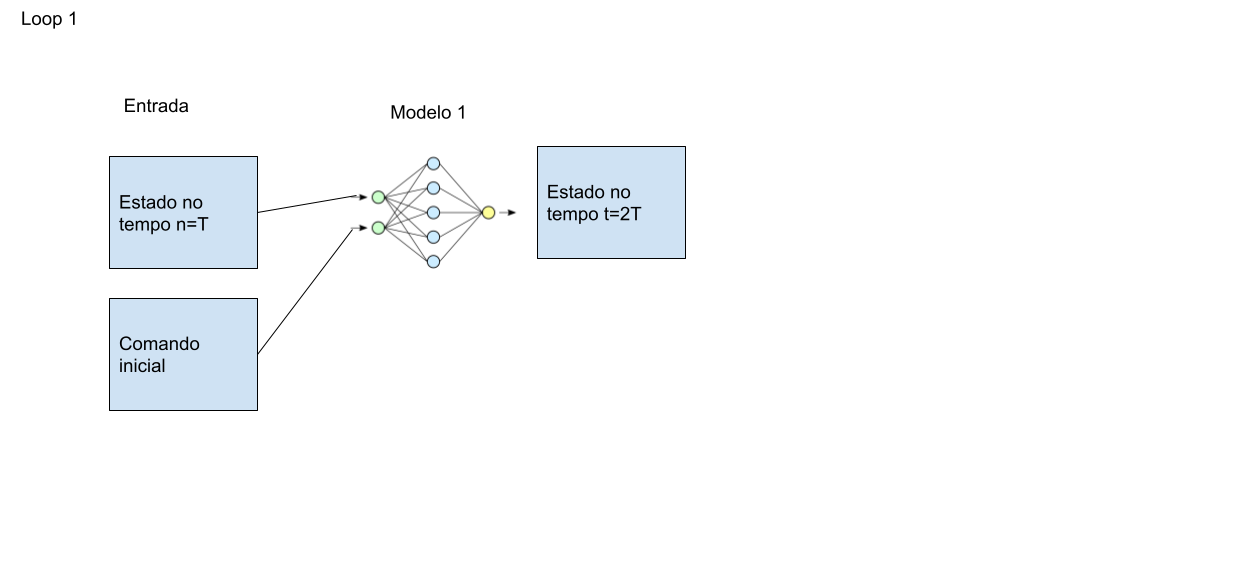
# 3 METODOLOGIA

Para testar e colher os resultados da rede neural preditiva em um modelo não linear, pretende-se utilizar um microcontrolador em um sistema embarcado baseado em um robô quadrúpede:

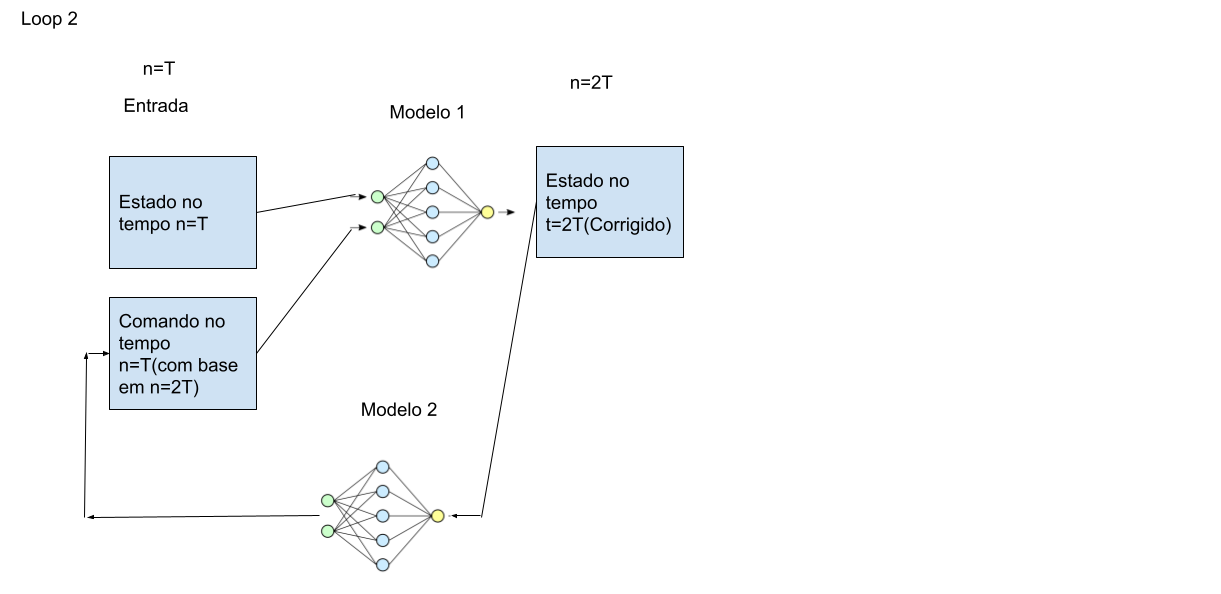
Figura 1: Spider robot, quad robot, quadruped



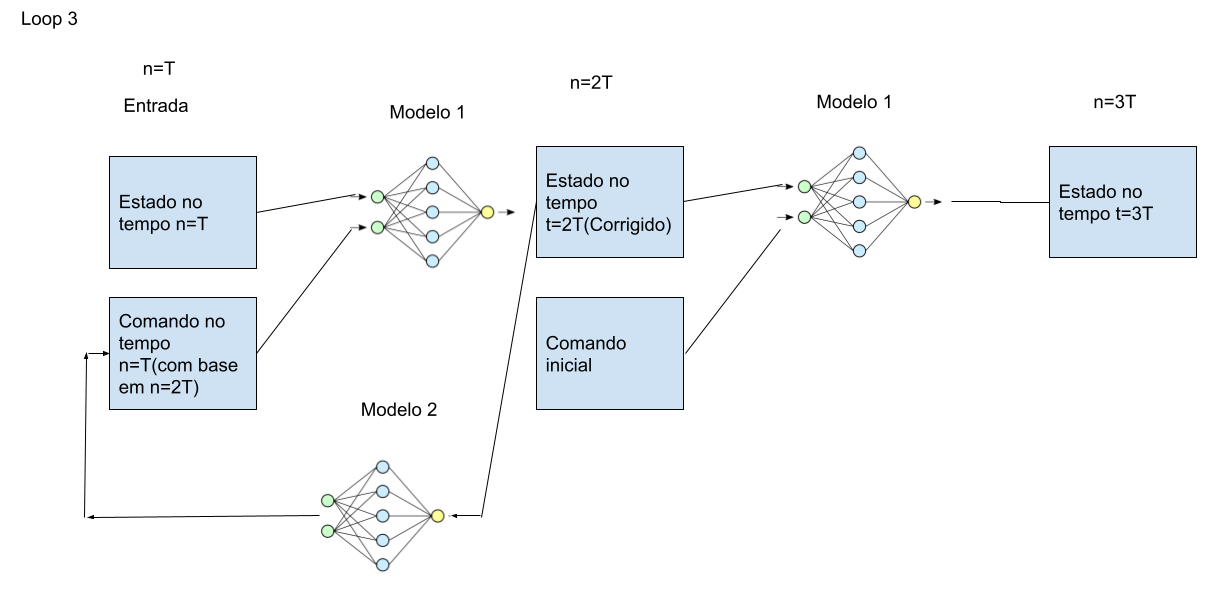
Fonte: <https://www.thingiverse.com/thing:2796820/files>

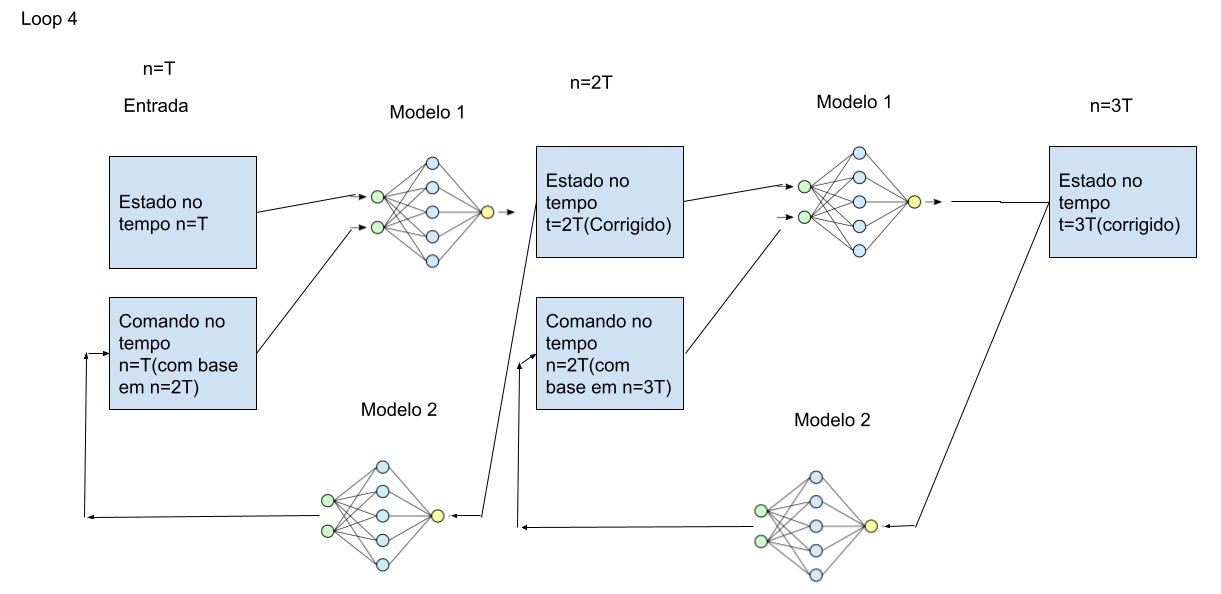


No segundo loop, o modelo 2 irá corrigir os estados, com base na trajetória prevista, por fim o modelo 1 irá prever novamente os estados com base nas alterações dadas nos comandos previstos.



Com isso, repetir o processo para as demais saídas.





A aprendizagem do modelo 1 será feita com base no quão certo ele previu o estado, utilizando um erro médio quadrático entre o vetor E[T], estado no tempo T, e o vetor Eh[T], estado previsto para o tempo T. E a aprendizagem do modelo 2 será feita com base no quanto o estado resultante do comando aproximou-se do estado desejado, em que o erro será baseado em uma pontuação dada de acordo com os objetivos que devem ser atingidos. O resultado deste processo será um vetor contendo a trajetória dos estados do sistema, este vetor é dinâmico, e seus valores são constantemente alterados e corrigidos pelos modelos trabalhando em conjunto.

Erro do modelo 1:

| Vetor resultante | | | | | |
| --- | --- | --- | --- | --- | --- |
| tempo(n) | n=T | n=2.T | n=3T | ... | n=N.T |
| Estados | E1[T] | E1[2T] | E1[3T] | ... | E1[NT] |
| E2[T] | E2[2T] | E2[3T] | ... | E2[NT] |
| E3[T] | E3[2T] | E3(T) | ... | E3(T) |
| E3[T] | E3[2T] | E3[3T] | ... | E3[NT] |
| Comandos | C1[T] | C1[2T] | C1[3T] | ... | C1[NT] |
| C2[T] | C2[2T] | C2[3T] | ... | C2[NT] |
| C3[T] | C3[T] | C3[3T] | ... | C3[NT] |

Modelo 1:

O modelo 1 terá como entrada os estados atuais do robô quadrúpede e os comandos dados no tempo atual e, terá como saída o próximo estado do robô. Considerando como entrada o estado e comandos medidos no tempo n=T, a saída do sistema será o estado no tempo n=(2T), onde T é o intervalo de medições feito pelo microcontrolador.

Modelo 2:

O modelo 2 irá determinar os comandos a serem dados para robô no tempo n=T, de acordo com os estados previsto para o tempo n=(2T) e o estado atual, considerando que o primeiro comando dado será um vetor nulo.

Portanto, o modelo de controle preditivo pode ser simplificado da seguinte maneira:

Os estados e comandos do sistema são:

| Variáveis De Estado | | | | |
| --- | --- | --- | --- | --- |
| Sensores | | | | |
| Nome | Descrição | Range | Tipo | unidade |
| EM1 | Ângulo do Motor 1 | 0-180 | int | graus |
| EM2 | Ângulo do Motor 2 | 0-180 | int | graus |
| EM3 | Ângulo do Motor 3 | 0-180 | int | graus |
| EM4 | Ângulo do Motor 4 | 0-180 | int | graus |
| EM5 | Ângulo do Motor 5 | 0-180 | int | graus |
| EM6 | Ângulo do Motor 6 | 0-180 | int | graus |
| EM7 | Ângulo do Motor 7 | 0-180 | int | graus |
| EM8 | Ângulo do Motor 8 | 0-180 | int | graus |
| EM9 | Ângulo do Motor 9 | 0-180 | int | graus |
| EM10 | Ângulo do Motor 10 | 0-180 | int | graus |
| EM11 | Ângulo do Motor 11 | 0-180 | int | graus |
| EM12 | Ângulo do Motor 12 | 0-180 | int | graus |
| EIX | Inclinação do robô no eixo x | 0-180 | int | graus |
| EIY | Inclinação do robô no eixo y | 0-180 | int | graus |
| EIZ | Inclinação do robô no eixo z | 0-180 | int | graus |
| EAX | Aceleração do robô no eixo x | 0-1 | int | - |
| EAY | Aceleração do robô no eixo y | 0-1 | int | - |
| EAZ | Aceleração do robô no eixo z | 0-1 | int | - |
| EV | Vibração do robô, medido indiretamente através da variação de EIX e EIY | 0-180 | float | graus/período |
| ED | Distância medida pelo sensor ultrassônico | 0-10 | float | metros |
| Comandos | | | | |
| Nome | Descrição | Range | Tipo | unidade |
| CM1 | comanda motor M1 | 0-180 | int | graus |
| CM2 | comanda motor M2 | 0-180 | int | graus |
| CM3 | comanda motor M3 | 0-180 | int | graus |
| CM4 | comanda motor M4 | 0-180 | int | graus |
| CM5 | comanda motor M5 | 0-180 | int | graus |
| CM6 | comanda motor M6 | 0-180 | int | graus |
| CM7 | comanda motor M7 | 0-180 | int | graus |
| CM8 | comanda motor M8 | 0-180 | int | graus |
| CM9 | comanda motor M9 | 0-180 | int | graus |
| CM10 | comanda motor M10 | 0-180 | int | graus |
| CM11 | comanda motor M11 | 0-180 | int | graus |
| CM12 | comanda motor M12 | 0-180 | int | graus |
| Critérios de desempenho do controle utilizados como treinamento do modelo 2(Modelo Preditivo) | | | | |
| Nome | Descrição | | Tipo | unidade |
| D\_TS | Tempo de subida medido em número de períodos | | float | T |
| D\_TA | Tempo de acomodação medido em número de períodos | | float | s |
| D\_E | Erro em regime permanente | | float | % |
| D\_EC | Esforço de controle | | float | % |

## 3.1 CRONOGRAMA

| **TAREFA** | **INÍCIO** | **TÉRMINO** |
| --- | --- | --- |
| **TAREFAS PRELIMINARES** | Maio | Junho |
| Aquisição dos Materiais | 1/5/22 | 15/5/22 |
| Montagem do robô | 15/5/22 | 29/5/22 |
| Montagem do circuito | 30/5/22 | 13/6/22 |
| Teste e validação do sistema | 14/6/22 | 28/6/22 |
| Estudo e Modelagem do Sistema | 14/6/22 | 28/6/22 |
| **ESTUDO E MODELAGEM DO SISTEMA** | Junho | Julho |
| Coleta de Dados do Comportamento do Sistema frente a diferentes entradas | 29/6/22 | 9/7/22 |
| Definição de um modelo matemático do Sistema | 10/7/22 | 20/7/22 |
| Estudo do modelo matemático utilizando ferramentas computacionais | 20/7/22 | 30/7/22 |
| **MODELAGEM DAS REDES NEURAIS** | Julho | Setembro |
| Descrição do sistema de controle e o algoritmo a ser utilizado | 30/7/22 | 6/8/22 |
| Definição dos Critérios de Desempenho | 7/8/22 | 14/8/22 |
| Definição dos modelos de Rede Neural | 15/8/22 | 22/8/22 |
| Algoritmo de propagação e retropropagação | 23/8/22 | 30/8/22 |
| Treinamento e Testes das Redes Neurais em Ambiente Simulado | 31/8/22 | 7/9/22 |
| **APLICAÇÃO DAS REDES NEURAIS NO ROBÔ QUADRUPEDE** | Setembro | Outubro |
| Aplicação das Redes Neurais pré treinadas no robô quadrupede | 8/9/22 | 15/9/22 |
| Análise dos Critérios de Desemepenho Na Rede Pré Treinada | 14/9/22 | 21/9/22 |
| Continuidade do Treinamento no Ambiente Real | 21/9/22 | 28/9/22 |
| Análise de Desempenho e Validação do Sistema | 28/9/22 | 5/10/22 |
| **Otimização Das Redes Neurais** | Outubro | Outubro |
| Técnicas para aperfeiçoamento das Redes Neurais | 6/10/22 | 13/10/22 |
| Interpretação e conclusão dos resultados | 13/10/22 | 20/10/22 |
| Registro de Desempenho | 21/10/22 | 28/10/22 |

## 

## 3.2 RECURSOS

### **3.2.1 Materiais**

Os componentes utilizados para este projeto são divididos em atuadores, sensores e outros, como descritos na tabela a seguir:

| Lista de Materiais | | | | |
| --- | --- | --- | --- | --- |
| Atuadores | | | | |
| Número | Componente | Modelo | Quantidade | Função |
| 1 | Servo Motor |  | 12 | Controle do ângulo das patas |
| Sensores | | | | |
| Número | Componente | Modelo | Quantidade | Função |
| 2 | Sensor Ultrassônico |  | 1 | Medição da distância linear de objetos no caminho |
| 3 | Módulo Sensor de Inclinação | MPU6050 | 1 | Medir a Inclinação e aceleração em x, y e z do sistema |
| Auxiliares | | | | |
| Número | Componente | Modelo | Quantidade | Função |
| 4 | Arduino Mega | - | 1 | Utilizado para controle do Sistema |
| 5 | Jumpers | - | 28 | Utilizado para conexão dos materiais |
| 6 | Bateria | - | 1 | Para alimentar o sistema e diminuir as variações de tensão da fonte |
| 7 | Fonte 5v-3A | - | 1 | Alimentar o sistema |
| 8 | Filamento de impressão 3D | - | 200g | Para impressão das peças |
| 9 | Espaguete termo retrátil | - | 1M | Para isolar o circuito |
| 10 | Estanho para Solda | - | 50g | Para soldar os componentes |
|  |  |  |  |  |

Para fazer o controle, foi definido a utilização do Arduino Mega, pela praticidade, custo e quantidade de saídas digitais pwm. A fonte a ser utilizada foi dimensionada pelo consumo de corrente do circuito. Utilizando uma fonte de bancada e um multímetro, foi aplicada uma tensão de 5V no circuito e os motores foram comandados para ângulos aleatórios a fim de definir qual o pico de consumo do sistema. O resultado foi que o circuito teve um consumo de 3A.

### **3.2.2 Ferramentas computacionais**

Para auxiliar nos testes e modelagem, tanto do sistema a ser controlado quanto dos modelos de redes neurais, foram utilizados as seguintes ferramentas:

* Jupyter Notebook: Utilizado para desenvolvimento das redes neurais a serem utilizadas no projeto
* Visual Studio Community: Programação do Arduino e simulação das Redes Neurais
* Mathworks MATLAB R2020a: Utilizado para simular o modelo matemático e obter os critérios de desempenho idealizados para o projeto
* Tinkercad: Utilizado para testar o código em c++ a ser utilizado no arduino mega

### **3.2.3 Equipamentos Utilizados**

Para montagem e testes, foram utilizados os seguintes equipamentos:

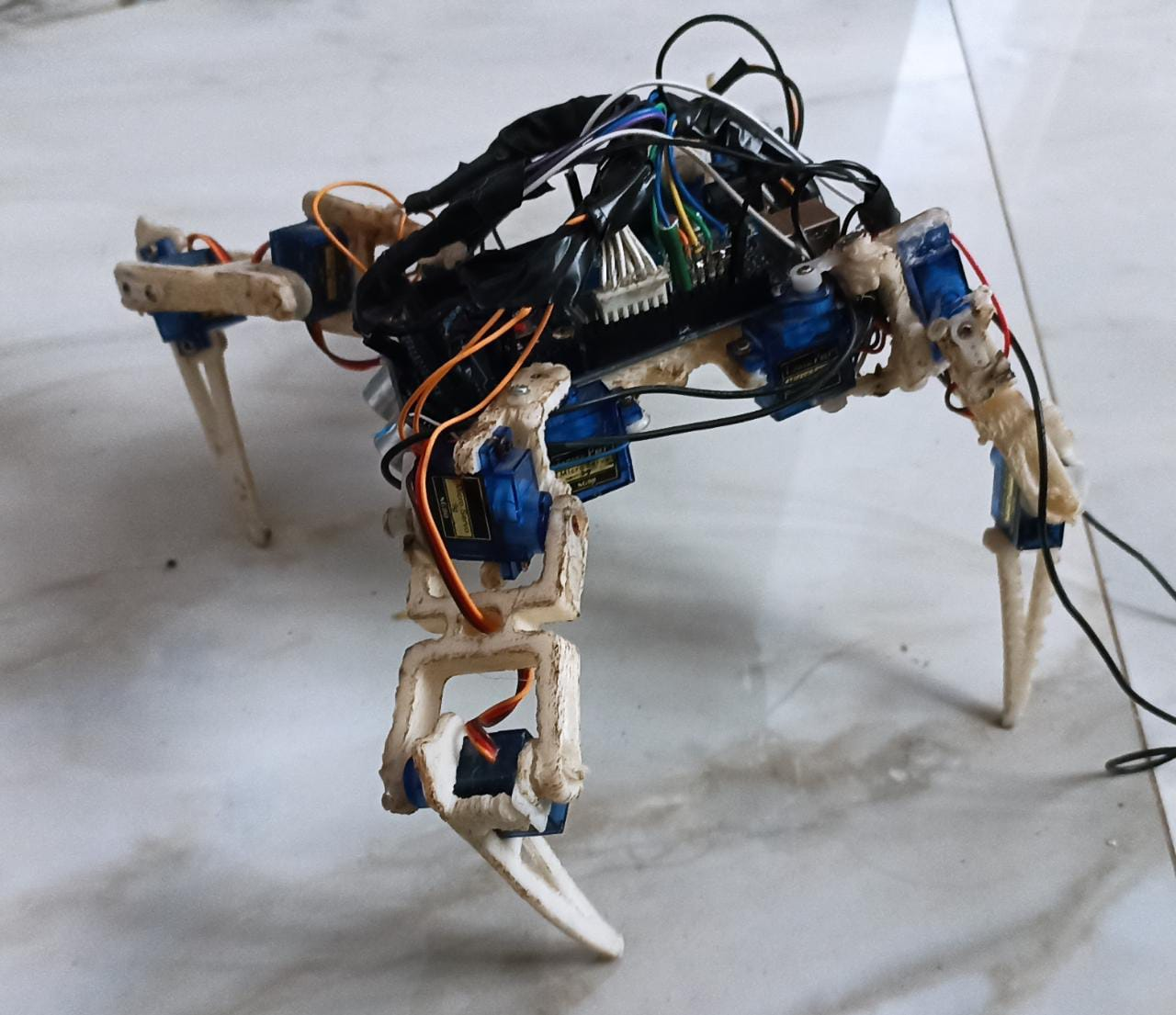
* Multímetros;
* Osciloscópio;
* Ferro de solda;
* Computador;
* Fonte de bancada;

## 3.3 MONTAGEM DO SISTEMA

### **3.3.1 Modelo em 3D**

O modelo em 3D foi obtido no site thingiverse, criado pelo usuário Monhambe em 16 de Fevereiro de 2018. Foi utilizada uma impressora 3D para obter as peças do sistema a ser controlado, na figura 2, que foi montado utilizando parafusos para fixar os servos motores. Por fim, os componentes foram conectados ao controlador com o auxílio de jumpers e de solda com estanho

FIGURA 2 - Robô Quadrúpede Montado



Após a Montagem, foram feitos testes de continuidade e a alimentação do sistema, para testar os componentes e validar a montagem do sistema.

## 3.4 Estudo e Modelagem do Sistema

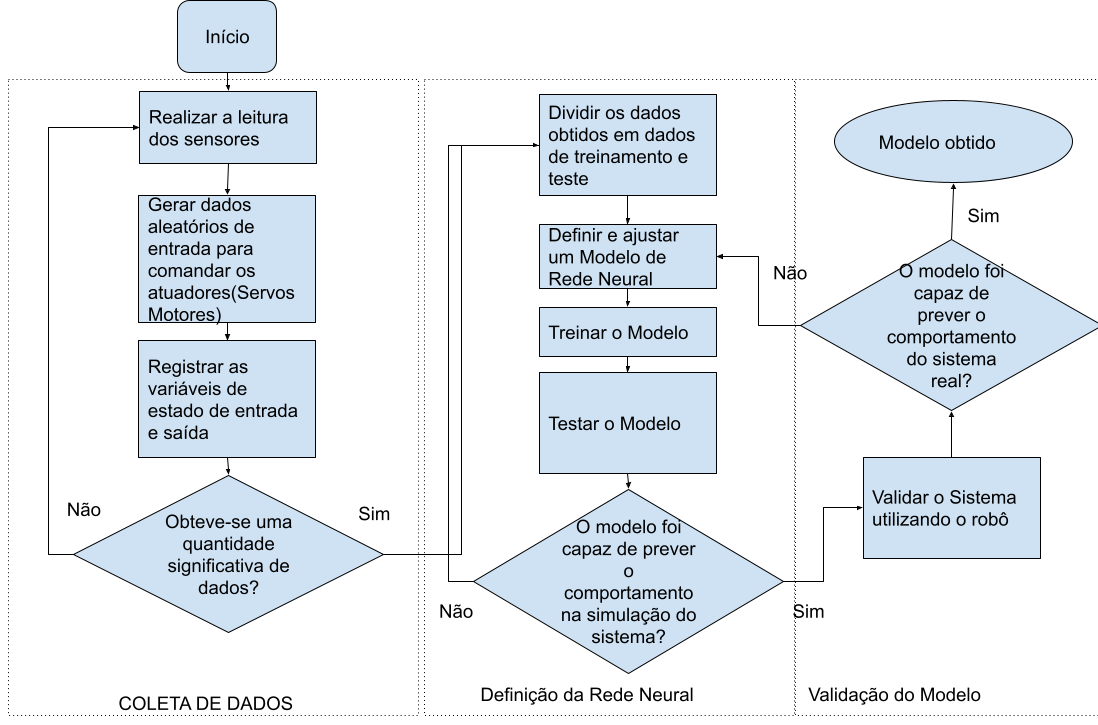
Para modelar um sistema é preciso descrever matematicamente o seu comportamento de forma a prever os estados de suas variáveis em função de perturbações e da própria dinâmica existente no sistema, isto é cria um bloco capaz de relacionar de forma aproximada uma entrada e uma saída.

Para o robô quadrúpede, pode-se considerar como entrada, os estados dos sensores e comandos a serem dados aos atuadores. Os estados são os ângulos atuais dos motores, a inclinação nos três eixos(x,y e z), a aceleração nos três eixos, e a distância medida pelo sensor ultrassônico. Os comandos dados são os ângulos inseridos na função Servo.write(), da biblioteca “Servo.h”, que determina para qual ângulo o motor deverá se mover.

Como saída, pode-se considerar os estados resultantes, ou seja, a inclinação, a aceleração e a distância da próxima leitura do sistema.

A técnica de modelagem utilizada é descrita no fluxograma a seguir:.

FIGURA 3: Modelagem do Sistema



Com esta técnica, pretende-se obter um modelo capaz de representar matematicamente o comportamento do sistema, prevendo os estados futuros com base nas variáveis de entrada do sistema com o menor erro possível. O modelo preditivo resultante será a base para a construção da rede neural que controlará o sistema, com base no erro previsto pelo modelo obtido neste método.

### **3.4.1 Coleta de Dados do Comportamento do Sistema frente a diferentes entradas**

Falar sobre como foi feita a coleta de dados, quais dados foram coletados, descrever o algoritmo e qual a importância de cada dado para a modelagem.

Falar sobre Persistência de excitação.

Descrever o sinal de entrada no sistema.

Descrever o sinal de saída do sistema.

Registrar graficamente os resultados.

### **3.4.2 Definição de um modelo matemático do Sistema Utilizando Redes Neurais**

Introduzir o método a ser utilizado para obtenção do modelo matemático

Utilização do matlab para obter a forma de onda e resposta ao impulso

### **3.4.3 Estudo do modelo matemático utilizando ferramentas computacionais**

Obter as respostas do modelo a diferentes entradas

Testar o modelo no simulador

Registrar as características do sistema de forma gráfica e conceitual;

## 3.5 Síntese do Controlador Neural

### **3.5.1 Descrição do sistema de controle e o algoritmo a ser utilizado**

Utilizando o modelo criado no item anterior, descrever o controlador ideal e características do controlador utilizando teorias de controle por redes neurais.

Descrever o algoritmo a ser utilizado para pré treino;

Descrever o algoritmo a ser utilizado para controle e tempo em ambiente real;

### **3.5.2 Definição dos Critérios de Desempenho**

Definir os critérios de desempenho a serem atingidos;

### **3.5.3 Definição dos modelos de Rede Neural**

Definir os modelos de redes neurais, tipos de camadas e suas características com base nos resultados

### **3.5.4 Algoritmo de propagação e retropropagação**

Descrever de forma matemática o erro a ser utilizado da saída das redes neurais

Descrever de forma matemática a propagação da rede neural e a retropropagação do erro.

Construção dos algoritmos de propagação e retropropagação em python

### **3.5.5 Análise de Desempenho e Validação do Sistema**

Analisar as redes neurais e o aprendizado no modelo computacional;

Validar o aprendizado;

### **3.5.5 Algoritmo genético para seleção da Rede Neural**

Descrever como o algoritmo será utilizado para seleção da melhor rede neural, com melhor estrutura para alcançar os objetivos.

## 3.6 APLICAÇÃO DAS REDES NEURAIS NO ROBÔ QUADRÚPEDE

### **3.6.1 Aplicação das Redes Neurais pré treinadas no robô quadrúpede**

Utilizar a rede neural que teve o melhor desempenho no simulador para construir a rede neural no robô quadrúpede, utilizando os pesos do pré treinamento

### **3.6.2 Análise dos Critérios de Desempenho Na Rede Pré Treinada**

Descrever os resultados obtidos ao longo do treinamento e medições no robô em simulação e ajustar os critérios de desempenho para o ambiente real

### **3.6.3 Continuidade do Treinamento no Ambiente Real**

Iniciar o treinamento no ambiente

### **3.6.4 Teste no Ambiente Real**

Colocar o robô quadrúpede em diferentes ambientes e analisar a capacidade de adaptação/

## 3.7 OTIMIZAÇÃO DAS REDES NEURAIS

### **3.7.1 Técnicas para aperfeiçoamento das Redes Neurais**

Descrever como serão feitos os ajustes finais para melhorar os resultados obtidos

### **3.7.2 Interpretação e conclusão dos resultados da otimização**

### **3.7.3 Registro de Desempenho**

Registrar os gráficos de desempenho ao longo do treinamento, tanto no simulador quanto no ambiente real

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 4.0 RESULTADOS E DISCUSSÃO

# 5.0 CONCLUSÃO

# 6.0 REFERÊNCIAS

[MODELAGEM PREDITIVA DE LINHA DE COSTA UTILIZANDO REDES NEURAIS ARTIFICIAIS](https://www.scielo.br/j/bcg/a/DJGyBq7LJm39Ggmj5KypQ4P/?format=pdf&lang=pt)

[Análise preditiva para identificação de alunos suscetíveis à evasão escolar Análise preditiva para identificação de alunos suscetíveis à evasão escolar](https://scholar.archive.org/work/eq6wfaftlnhyfovndjyp5htx7a/access/wayback/https://brazilianjournals.com/index.php/BRJD/article/download/33014/pdf)

[AERONAVES CONFIGURADAS POR CONTROLE DO TIPO PREDITIVO NEURAL](http://mtc-m16.sid.inpe.br/col/sid.inpe.br/jeferson/2003/05.14.14.55/doc/publicacao.pdf)

[Arquitetura neural para controle da locomoção de um robô quadrúpede baseada em referências biológicas](https://www.lume.ufrgs.br/handle/10183/8277)

[O que são sistemas embarcados](https://www.oficinadanet.com.br/post/13538-o-que-sao-sistemas-embarcados#:~:text=O%20sistema%20embarcado%2C%20tamb%C3%A9m%20chamado,ao%20sistema%20que%20ele%20controla.&text=Atualmente%2C%20v%C3%A1rios%20produtos%20possuem%20uma,nosso%20dia%2Da%2Ddia).

<https://materialpublic.imd.ufrn.br/curso/disciplina/1/63/5/5>

<https://sci-hub.se/10.1109/TIE.2011.2169636>

GONÇALVES, RODRIGO MIKOSZ;. **MODELAGEM PREDITIVA DE LINHA DE COSTA UTILIZANDO REDES NEURAIS ARTIFICIAIS.** Universidade Federal de Pernambuco (UFPE) Centro de Tecnologia e Geociências – CTG, Departamento de Engenharia. Disponível em <https://www.scielo.br/j/bcg/a/DJGyBq7LJm39Ggmj5KypQ4P/?format=pdf&lang=pt>

#include <Ultrasonic.h>

class EstadoAtual{

public:

long int tempo;//Marca o tempo do estado

int EM[12];// Marca o Estado dos 12 motores

float EIX;//Marca a inclinacao no eixo X

float EIY;//

float EIZ;//

float EAX;//Mostra a aceleracao atual do robo no eixo X

float EAY;

float EAZ;

float ED;// Marca a distancia lida pelo sensor ultrassonico

int CM[12];//Marca o comando atual dado ao motor

};

class vetorEstados{

private:

EstadoAtual historico[1000];

public:

};

class modoColetaDados{

void setup(){

Serial.println("Modo coleta de dados iniciado");

Serial.println("Exibe vetor de estados coletado")

}

void loop(){

Serial.println("1: Atualizando Sensores");

Serial.println("2: Gerando comandos aleatorios");

Serial.println("3: Gravando Estados Atuais na ultima linha do arquivo");

Serial.println("4: Exibindo Estado atual");

}

};

(Página onde será inserida a ficha catalográfica- Solicitar junto à biblioteca)

Arthur Carlos de Faria

**Modelagem e Controle de Robô Quadrúpede Utilizando Redes Neurais preditivas e Adaptativas**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Aprovado em: \_\_\_/\_\_\_/\_\_\_\_\_\_ pela banca examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof.(a) Dr. XXXXXXXXX (Orientador) – IFMG Campus Betim

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof.(a) Me. XXXXXXXXX (Coorientador) – IFMG Campus Betim

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. XXXXXXXXX– Instituição

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eng.(a) XXXXXXXXX– Instituição

Dedicatória (opcional).

**AGRADECIMENTOS**

Agradecimentos (opcional).

Epígrafe (opcional).

**Resumo**

Breve texto que apresenta o objetivo do trabalho, os métodos utilizados, os resultados e considerações finais. Deve ser escrito de maneira clara e objetiva, evitando o uso da primeira pessoa, conter aproximadamente de 150 a 500 palavras.

**Palavras-chave:** palavra 1; palavra 2; palavra 3; palavra 4; palavra 5.

**ABSTRACT**

Elemento obrigatório. Versão em língua estrangeira do resumo, com as mesmas informações do resumo em língua portuguesa.

No IFMG, optou-se pelo resumo no idioma inglês (Abstract) ou Espanhol (Resumen).

**Keywords:** keyword 1; keyword 2; keyword 3; keyword 4; keyword 5.

**LISTA DE ILUSTRAÇÕES**

[Figura 1 – Parte da estrutura interna do IFMG Campus Betim. 17](#_heading=h.2et92p0)

**lista de gráficos**

**No table of figures entries found.**

**LISTA DE QUADROS**

[Quadro 1.2 – Referência para estruturação de um trabalho de conclusão de curso (TCC) de um curso de graduação. Essa estrutura não é obrigatória. 15](#_heading=h.1fob9te)

**LISTA DE TABELAS**

**No table of figures entries found.**

**LISTA DE ABREVIATURAS E SIGLAS**

ABNT - Associação Brasileira de Normas Técnicas

IFMG - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais

NBR - Normas Técnicas Brasileiras

TCC - Trabalho de Conclusão de Curso

**lista de símbolos**

| @ | Arroba |
| --- | --- |
|  | Constante de tempo do sistema de primeira ordem |
| © | Copyright |
|  | Função de transferência |

**Sumário**

[**1**](#_heading=h.30j0zll) **INTRODUÇÃO 15**

[**1.1**](#_heading=h.3dy6vkm) **Justificativa 17**

[**1.2**](#_heading=h.1t3h5sf) **Colocação do Problema 17**

[**1.3**](#_heading=h.4d34og8) **Objetivos 17**

[**1.4**](#_heading=h.2s8eyo1) **Organização do Trabalho 17**

[**2**](#_heading=h.17dp8vu) **FUNDAMENTAÇÃO TEÓRICA 18**

[**2.1**](#_heading=h.3rdcrjn) **Cronograma 18**

[**2.2**](#_heading=h.26in1rg) **Recursos 18**

[**3**](#_heading=h.lnxbz9) **METODOLOGIA 19**

[**4**](#_heading=h.35nkun2) **RESULTADOS E DISCUSSÃO 20**

[**5**](#_heading=h.1ksv4uv) **CONCLUSÃO 21**

[**Referências 22**](#_heading=h.44sinio)

[**glossário (opcional) 23**](#_heading=h.2jxsxqh)

[**Apêndice A – <TÍTULO DESTE APÊNDICE> 24**](#_heading=h.z337ya)

[**Anexo A – <Título deste anexo> 25**](#_heading=h.3j2qqm3)

[**ÍNDICE (OPCIONAL) 26**](#_heading=h.1y810tw)

# INTRODUÇÃO

A organização do corpo do texto fica a critério do autor e do orientador. O Quadro 1.2 apresentada uma referência de estruturação de um TCC. É apenas uma referência, para servir de guia, não é obrigatório seguir essa estrutura.

**CONVERSE COM SEU ORIENTADOR PARA ESTRUTURAR O SEU TCC!**

Quadro 1.2 – Referência para estruturação de um trabalho de conclusão de curso (TCC) de um curso de graduação. Essa estrutura não é obrigatória.

| **Capítulo / Seção** | **Breve Descrição do Conteúdo** |
| --- | --- |
| **1 INTRODUÇÃO** | Introduzir e delimitar o tema  Contextualizar o problema a ser investigado |
| **1.1 Justificativa** | Mostrar a relevância do assunto  Justificar necessidade de realização do trabalho |
| **1.2 Colocação do Problema** | Expor o problema de forma clara, concisa e direta  Apresentar pergunta que fundamenta o trabalho  Apresentar hipótese (se aplicável) |
| **1.3 Objetivos** | Apresentar objetivo geral em 1 parágrafo  Apresentar objetivos específicos em tópicos, iniciando com verbos fortes, mais direcionados |
| **1.4 Organização do Trabalho** | Apresentar brevemente a estrutura do trabalho  Apresentar o que tem em cada capítulo |
| **2 FUNDAMENTAÇÃO TEÓRICA** (ou Revisão Bibliográfica) | Apresentar os principais métodos e tecnologias disponíveis para resolução do problema (“estado da arte”)  Referenciar trabalhos que tentaram resolver problemas semelhantes (muitas citações)  *Obs.: não é necessário descrever com detalhes TUDO o que foi utilizado no trabalho, mas o mínimo necessário para compreensão dos principais pontos do trabalho. Lembre-se: seu trabalho será lido, principalmente, por engenheiros e estudantes de engenharia.* |
| **3 METODOLOGIA** (ou Materiais e Métodos) | Descrever os materiais utilizados e o papel de cada um deles no desenvolvimento  Descrever os procedimentos realizados e as razões para a escolha da metodologia usada |
| **3.1 Cronograma**  (TCC1 apenas) | Tabela com cronograma (e.g., mensal ou quinzenal) para todo o TCC, começando no TCC 1 até a defesa |
| **3.2 Recursos**  (TCC1 apenas) | Tabela com recursos necessários para realização do trabalho, normalmente classificados em material permanente, material de consumo e pessoal |
| **4 RESULTADOS E DISCUSSÃO** | Apresentar os resultados alcançados usando, se possível, recursos visuais como gráficos e tabelas  Analisar criticamente os resultados, fornecendo possíveis razões para os resultados alcançados |
| **5 CONCLUSÃO** (ou Considerações Finais) | Sintetizar, brevemente, o que foi feito e as principais conclusões do trabalho  Apontar possíveis desenvolvimentos futuros para avanço do tema |
| **REFERÊNCIAS** | Lista de referências bibliográficas conforme padrão ABNT  *Obs.: bibliografias consultadas, mas não citadas no texto, não devem ser listadas!* |
| **APÊNDICES** | Se aplicável, colocar material (elaborados pelo próprio autor) importante para o detalhamento do trabalho |
| **ANEXOS** | Se aplicável, colocar documentos anexos (elaborados por terceiros) importantes para o detalhamento do trabalho |

Fonte: Elaborado pelos professores do DAUTI do IFMG Campus Betim.

Para não ter dor de cabeça com estruturação e formatação do trabalho, use as ferramentas de Estilo do Word, para escrever capítulos e seções. Assim, elas aparecem automaticamente no Sumário do trabalho!

Use também as numerações automáticas de legendas em Figuras, Tabelas, Quadros, Equações, etc.

Exemplo de citação direta com mais de 3 linhas.

As citações diretas, isto é, que ocupam quatro linhas ou mais, devem ter esta formatação aqui. Eventualmente, pode ser colocado **um grifo em negrito para destacar a informação mais importante**, a ser extraída da citação, para ajudar o leitor a selecionar a informação mais relevante de todo esse texto que ele está vendo. Neste caso, deve-se colocar a expressão grifo do autor entre parênteses. Se não tiver grifo, é só tirar. (AUTOR, DATA, p. XXX, grifo do autor).

A Figura 1 mostra um exemplo de figura.

Figura 1 – Parte da estrutura interna do IFMG Campus Betim.



Fonte: IFMG Campus Betim.

A Tabela 1 mostra um exemplo de tabela. Para atualizar a numeração das tabelas, equações, gráficos e figuras, basta selecionar a numeração e apertar F9.

Tabela 1 – Este é o título da tabela.

| Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| --- | --- | --- | --- | --- |
| 1 | 12 | 52 | 64 | 99 |
| 2 | 12 | 43 | 345 | 88 |
| 3 | 13 | 523 | 45 | 54 |
| 4 | 14 | 73 | 34 | 34 |

Fonte: IFMG Campus Betim.

## Justificativa

xxxxxxxxxxxxxxxxxxxxxxxxx.

## Colocação do Problema

xxxxxxxxxxxxxxxxxxxxxxxxx.

## Objetivos

xxxxxxxxxxxxxxxxxxxxxxxxx.

## Organização do Trabalho

xxxxxxxxxxxxxxxxxxxxxxxxx.

# FUNDAMENTAÇÃO TEÓRICA

xxxxxxxxxxxxxxxxxxxxxxxxx.

## Cronograma

xxxxxxxxxxxxxxxxxxxxxxxxx.

## Recursos

xxxxxxxxxxxxxxxxxxxxxxxxx.

# METODOLOGIA

xxxxxxxxxxxxxxxxxxxxxxxxx.

# RESULTADOS E DISCUSSÃO

xxxxxxxxxxxxxxxxxxxxxxxxx.

# CONCLUSÃO

xxxxxxxxxxxxxxxxxxxxxxxxx.

###### Referências

Lista de referências (coloque apenas as que foram citadas!). Existe uma ferramenta do Word para fazer citação automática e criando link da citação para a referência.

###### GLOSSÁRIO (opcional)

**Trabalho de Conclusão de Curso**: é um trabalho realizado ao final de um curso de graduação, que é regulamentado pelo Colegiado do Curso...

**Outro termo**: definição desse outro termo...

###### Apêndice A – <TÍTULO DESTE APÊNDICE>

O apêndice contém material produzido pelo próprio autor. Para mais de um apêndice, numere-os com “APÊNDICE A”, “APÊNDICE B”, etc. As figuras, tabelas, etc. contidas no apêndice não devem aparecer nas listas de figuras, tabelas, etc.

###### Anexo A – <Título deste anexo>

O anexo contém material elaborado por terceiros. São numerados com “ANEXO A”, “ANEXO B”, etc. As figuras e tabelas contidas no anexo não devem aparecer nas listas de figuras, tabelas, etc.

###### ÍNDICE (opcional)

Resultados, 16, 21

Outro Termo, <páginas em que o termo aparece>