



UNIVERSIDADE FEDERAL DE PERNAMBUCO

Servomecanismo

Departamento de Eletrônica e Sistemas

Modelo dinâmico de robôs

Bruno Guimarães - Requisitos

Rodrigo Lima Leandro - Relatório e Fundamentação Teórica

Henrique Silva - Verificação

Pedro Lucas - Implementação

Recife, 21 de Agosto de 2023

Sumário

1	Introdução	2
2	Objetivos	2
3	Fundamentação Teórica	3
3.1	Cinemática do Robô de Tração Diferencial:	3
3.2	Equações de Estado e Modelagem Dinâmica do Robô	5
3.2.1	Porquê Utilizar Equações de Estados	5
3.2.2	Equações de Estado	6
3.3	Diagrama de Controle Detalhado para Robô Móvel: Análise e Explicação . .	7
4	Requisitos	10
4.1	Requisito 1: Ambas velocidades iguais e mesmo sentido	10
4.2	Requisito 2: Ambas velocidades iguais, sentidos opostos.	11
4.3	Requisito 3: Uma velocidade igual a zero, a outra diferente de zero	11
4.4	Requisito 4: Ambas velocidades diferentes e nao nulas	11
5	Verificação	11
5.1	Parâmetros de simulação	11
5.2	Requisitos a serem verificados	12
5.2.1	Requisito 1: Ambas velocidades iguais e mesmo sentido	12
5.2.2	Requisito 2: Ambas velocidades iguais, sentidos opostos	12
5.3	Requisito 3: Uma velocidade igual a zero, a outra diferente de zero	13
5.4	Requisito 4: Ambas velocidades diferentes e nao nulas	13
6	Projeto Final do Controlador/Compensador	14
6.1	Seleção de Componentes	14
6.2	Características Elétricas	15
6.3	Implementação do Circuito	15
6.4	Estimativa de Custo	15
6.5	Considerações Finais	16
7	Conclusão	16

1 Introdução

Os avanços na robótica têm desencadeado uma crescente busca por soluções que aprimorem a compreensão e o controle dos sistemas robóticos. Um marco nesse sentido é a criação de modelos dinâmicos de robôs, que permitem simular e prever o comportamento de máquinas autônomas em diferentes cenários. Este relatório aborda um projeto de desenvolvimento de um modelo dinâmico de robôs, com foco em um robô de kit Arduino, que utiliza atuadores para controlar a velocidade de suas rodas e assim se mover em um plano bidimensional.

O projeto se propõe a explorar e analisar diferentes aspectos fundamentais da dinâmica robótica, desde a relação entre entradas (posição de referência no plano xy) e saídas (velocidade das rodas direita e esquerda), até a construção de um modelo considerando as particularidades das rodas, a geometria do robô e os limites físicos impostos pela aceleração e peso.

Uma das ênfases do projeto reside na aplicação de um controlador de velocidade, representado pelo ESC (Electronic Speed Controller), cuja função de transferência de primeira ordem será considerada. Além disso, o relatório abordará a simulação animada do sistema utilizando a plataforma Xcos, onde a representação visual do robô e sua orientação serão apresentadas através de uma barra, a fim de evidenciar como o modelo dinâmico desenvolvido satisfaz os requisitos estabelecidos.

Ao longo deste relatório, serão detalhados os estudos e etapas seguidas para a construção do modelo dinâmico do robô, levando em consideração aspectos cruciais como as variáveis de estado, a geometria do sistema, os limites físicos e as características do controlador de velocidade. O objetivo final é demonstrar como o modelo é capaz de capturar a essência do comportamento do robô em resposta às diferentes entradas e condições impostas pelo ambiente.

2 Objetivos

O objetivo deste projeto é desenvolver um modelo dinâmico abrangente para um robô controlado por kit Arduino, que seja capaz de simular seu comportamento em um plano bidimensional (plano xy) a partir de entradas de posição de referência. O modelo visa representar com precisão a interação entre as entradas, que determinam a posição desejada no plano, e as saídas, que controlam as velocidades das rodas direita e esquerda do robô.

Os principais objetivos do projeto são os seguintes:

Construção do Modelo Dinâmico: Desenvolver um modelo matemático que relacione as entradas (posição de referência) com as saídas (velocidades das rodas direita e esquerda). Esse modelo deve considerar a geometria do robô, as características das rodas, bem como as limitações impostas pelo peso, distância entre as rodas e aceleração máxima.

Incorporação do Controlador de Velocidade: Introduzir um controlador de velocidade no modelo, representado pelo Electronic Speed Controller (ESC), com uma função de transferência aproximadamente de primeira ordem. Isso permitirá a simulação do comporta-

mento do robô sob diferentes condições de entrada e demonstrará como o controlador afeta as respostas do sistema.

Simulação Visual: Implementar uma simulação visual animada utilizando a plataforma Xcos, onde o robô e sua orientação são representados graficamente por meio de uma barra. Isso proporcionará uma compreensão visual do movimento do robô em resposta às entradas e ajudará a verificar se o modelo atende a todos os requisitos definidos.

Análise e Validação: Realizar análises e testes extensivos do modelo dinâmico e da simulação, considerando diferentes cenários de entrada, velocidades das rodas. A validação adequada do modelo garantirá que ele seja uma representação precisa do comportamento do robô real.

Demonstração dos Requisitos: Mostrar que o modelo dinâmico atende aos requisitos definidos no escopo do projeto, incluindo a capacidade de controlar o movimento do robô, simular a orientação e o deslocamento no plano xy, e responder adequadamente às diferentes entradas e limitações físicas

Por meio desses objetivos, o projeto visa aprimorar a compreensão da dinâmica de sistemas robóticos, permitindo uma simulação detalhada do comportamento de um robô controlado por kit Arduino. Além disso, ele oferece a oportunidade de explorar e aplicar conceitos teóricos em um contexto prático, contribuindo para o melhor entendimento da disciplina.

3 Fundamentação Teórica

A fundamentação teórica é um elemento crucial para a compreensão aprofundada do projeto de desenvolvimento de um modelo dinâmico para o robô controlado por kit Arduino. Neste tópico, serão abordados conceitos fundamentais que embasam a construção e a análise do modelo, bem como as teorias relacionadas ao controle do robô e à simulação do seu comportamento. As seguintes seções detalham os principais aspectos teóricos relevantes para o projeto:

3.1 Cinemática do Robô de Tração Diferencial:

A cinemática é um campo fundamental na robótica que aborda a análise do movimento sem a consideração das forças envolvidas. Nesta seção, vamos explorar a cinemática específica de um robô de tração diferencial, levando em conta as características das rodas independentes. Discutiremos conceitos como posição, velocidade e aceleração das rodas, bem como a relação entre as entradas (posição de referência) e as saídas (velocidades das rodas) do sistema. Também analisaremos a conversão entre as velocidades das rodas e a velocidade linear do robô

Relação entre Entradas e Saídas:

A cinemática do robô de tração diferencial pode ser dividida em duas partes essenciais:

Relação entre Velocidades das Rodas e Variáveis de Saída:

A relação entre as velocidades das rodas direita ($velocidade_direita$) e esquerda ($velocidade_esquerda$) e as variáveis de saída é crucial para compreender o movimento do robô durante curvas e deslocamentos. Considerando a localização do robô como um ponto central no plano cartesiano, onde a distância até as rodas direita e esquerda é L , a relação entre as velocidades das rodas e as variáveis de saída pode ser expressa como:

Equação para a velocidade da roda direita:

$$velocidade_direita = \omega(1) \cdot R$$

Equação para a velocidade da roda esquerda:

$$velocidade_esquerda = \omega(2) \cdot R$$

Nestas equações, ω representa a velocidade angular das rodas do robô e R é o raio das rodas. Para um robô de tração diferencial, onde cada roda é controlada independentemente, a velocidade angular de cada roda está diretamente relacionada à velocidade linear tangencial que a roda está produzindo. A velocidade angular (ω) está relacionada ao ângulo percorrido pela roda em relação ao tempo. Quando o robô faz curvas, a roda do lado externo da curva percorre uma distância maior do que a roda do lado interno da curva, resultando em diferentes velocidades angulares para as rodas.

Considerações Físicas:

Equilíbrio de Velocidades:

A velocidade linear média ($velocidade_linear$) reflete o equilíbrio entre as velocidades das rodas direita e esquerda. Se ambas as rodas girarem à mesma velocidade, o robô se moverá em linha reta

Desvio Angular:

A velocidade angular (ω) mede a diferença entre as velocidades das rodas direita e esquerda. Se ω for positivo, o robô girará no sentido anti-horário; se for negativo, o robô girará no sentido horário. Quanto maior a diferença de velocidade angular, mais pronunciada será a curva realizada pelo robô.

Conversão de Velocidades em Movimento:

A conversão das velocidades das rodas em movimento efetivo do robô é um elemento crítico da cinemática. A velocidade média (*velocidade_linear*) é calculada como a média das velocidades das rodas direita e esquerda multiplicada por uma constante de proporcionalidade. A velocidade angular (ω) é obtida a partir da diferença entre as velocidades das rodas, ponderada pela constante $\beta = 1/2L$.

Equação para a velocidade linear média do robô:

$$velocidade_linear = \frac{velocidade_direita + velocidade_esquerda}{2} \cdot K$$

Equação para a velocidade angular do robô:

$$\omega = \beta \cdot (velocidade_direita - velocidade_esquerda)$$

O ângulo de orientação θ do robô é calculado integrando a velocidade angular ω ao longo do tempo:

Equação para o ângulo de orientação do robô:

$$\theta(t) = \int_{-\infty}^t \omega(t) dt + \theta(0)$$

Essas equações são fundamentais para descrever a cinemática do robô de tração diferencial, permitindo prever seu movimento e comportamento durante deslocamentos e curvas. Através dessas equações, é possível entender como as velocidades das rodas se traduzem em movimento e orientação do robô no plano cartesiano. Isso é particularmente útil para o planejamento e controle de trajetórias do robô em diversas situações e ambientes.

3.2 Equações de Estado e Modelagem Dinâmica do Robô

Uma parte essencial na compreensão do comportamento do nosso robô é a formulação e utilização das equações de estado. Essas equações constituem um conjunto de ferramentas matemáticas fundamentais que nos permitem descrever, de maneira precisa, como as diferentes variáveis do sistema evoluem ao longo do tempo. Nesta seção, exploraremos o porquê do uso das equações de estado, bem como o significado de aplicá-las em nosso projeto de modelagem dinâmica de robôs.

3.2.1 Porquê Utilizar Equações de Estados

As equações de estado desempenham um papel crucial ao traduzir as características físicas do robô em um formato matemático que podemos analisar e controlar. Elas nos oferecem uma maneira sistemática de relacionar as variáveis que descrevem o estado do robô - como sua posição, orientação e velocidades das rodas - com suas respectivas taxas de mudança.

Essencialmente, as equações de estado nos permitem prever como as condições do robô evoluirão de um momento para o próximo, com base nas entradas e nas forças que atuam sobre ele.

No contexto do nosso projeto de modelagem dinâmica de robôs, a aplicação das equações de estado se torna especialmente relevante pelos seguintes motivos:

Previsão de Comportamento: Através das equações de estado, podemos antecipar como o robô se moverá e se orientará em resposta às velocidades das rodas e outras influências. Isso é essencial para o planejamento de trajetórias, a avaliação do desempenho do robô e a previsão de resultados.

Análise de Estabilidade: As equações de estado nos permitem examinar a estabilidade do sistema. Podemos identificar condições que levam a comportamentos instáveis ou indesejados e ajustar as entradas ou os controladores para manter o sistema sob controle.

Projeto de Controle: Utilizando as equações de estado, somos capazes de projetar estratégias de controle que direcionam o robô de acordo com objetivos específicos. Podemos ajustar as velocidades das rodas para atingir trajetórias desejadas ou realizar tarefas específicas.

Simulação e Validação: As equações de estado viabilizam a criação de simulações precisas do comportamento do robô. Isso facilita a validação de projetos e a realização de testes em diversos cenários antes da implementação prática.

3.2.2 Equações de Estado

Façamos:

$$\begin{bmatrix} x(t) \\ y(t) \\ \Theta(t) \end{bmatrix}$$

Onde $x(t)$ é a coordenada x do robô no tempo t , $y(t)$ é a coordenada y do robô no tempo t e $\theta(t)$ é o ângulo de orientação do robô no tempo t . Essas são nossas variáveis de estado.

Agora vamos impor que o vetor de velocidade \vec{V}_r tenha componentes V_x na direção x e V_y na direção y . Portanto, $\vec{V}_r = [V_x, V_y]$.

O deslocamento do robô (x, y) em relação ao tempo t será uma combinação do deslocamento inicial ($x(0), y(0)$) e o vetor de velocidade \vec{V}_r multiplicado pelo tempo t :

$$\begin{aligned} x(t) &= x(0) + V_x \cdot t \\ y(t) &= y(0) + V_y \cdot t \end{aligned}$$

Então, a partir dessas condições, podemos definir as equações de estado como:

$$\begin{aligned} \dot{x}(t) &= V_x \\ \dot{y}(t) &= V_y \\ \dot{\theta}(t) &= \frac{1}{2L} \cdot (\text{velocidade_direita} - \text{velocidade_esquerda}) \end{aligned}$$

Agora, vamos expressar as componentes V_x e V_y do vetor de velocidade \vec{V}_r em termos do ângulo de orientação θ . A partir das seguintes relações:

$$V_x = V_r \cdot \cos(\theta)$$

$$V_y = V_r \cdot \sin(\theta)$$

Obtemos, finalmente:

$$\dot{x}(t) = V_r \cdot \cos(\theta)$$

$$\dot{y}(t) = V_r \cdot \sin(\theta)$$

$$\dot{\theta}(t) = \frac{1}{2L} \cdot (\text{velocidade_direita} - \text{velocidade_esquerda})$$

3.3 Diagrama de Controle Detalhado para Robô Móvel: Análise e Explicação

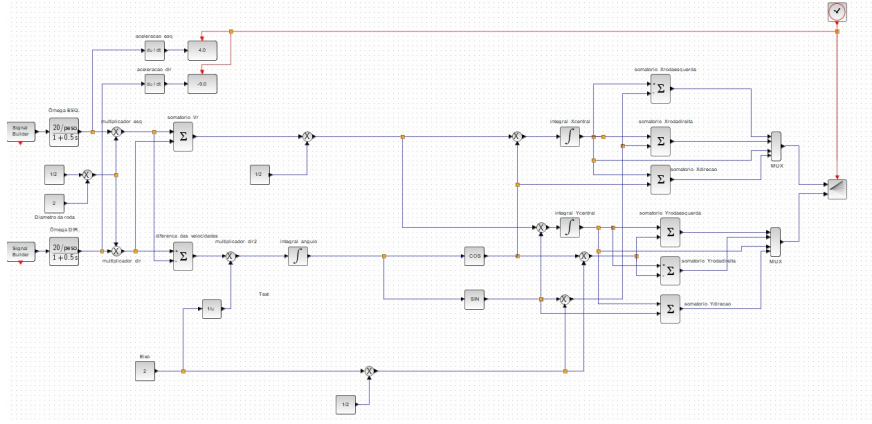


Figura 1: Diagrama elétrico simplificado da implementação do controlador/compensador.

O diagrama apresentado descreve um sistema de controle para um robô com tração diferencial, ou seja, ele possui dois lados independentes, o direito e o esquerdo. O objetivo é controlar o movimento do robô através do ajuste das velocidades angulares das rodas dos lados direito e esquerdo, permitindo que ele realize manobras e movimentos precisos. Para alcançar esse objetivo, uma série de blocos interconectados é utilizada para processar os sinais de referência, realizar operações matemáticas e controlar os motores.

Sinais de Referência

O sistema começa com dois blocos "Signal Builder", que geram sinais de referência para as velocidades angulares dos lados direito e esquerdo do robô. Esses sinais de referência podem ser modelados como funções periódicas conforme indicado pelas definições de spline.

Modelagem das Velocidades das Rodas (Ω_{esq} e Ω_{dir}):

Os blocos "Omega_ESQUERDO" e "Omega_DIREITO" são responsáveis por modelar as velocidades das rodas esquerda e direita do robô, respectivamente. Eles são representados

pelas funções de transferência $G_{\Omega_{esq}}(s)$ e $G_{\Omega_{dir}}(s)$, que descrevem a relação entre a transformada de Laplace da velocidade angular das rodas e a transformada de Laplace do sinal de referência $R(s)$.

No caso do bloco “Omega_ESQUERDO”, a função de transferência $G_{\Omega_{esq}}(s)$ é dada por:

$$G_{\Omega_{esq}}(s) = \frac{\Omega_{esq}(s)}{R(s)} = \frac{\frac{20}{peso}}{0.5s + 1}$$

onde $\Omega_{esq}(s)$ é a transformada de Laplace da velocidade angular da roda esquerda, e $peso$ representa o peso total do robô. O ganho de $\frac{20}{peso}$ na função de transferência é uma escolha que leva em consideração a influência do peso do robô na velocidade angular das rodas. Ele ajusta a relação entre o sinal de referência $R(s)$ e a velocidade angular das rodas $\Omega_{esq}(s)$, garantindo que o sistema responda adequadamente aos comandos de velocidade, o bloco “Omega_DIREITO” funciona de forma análoga.

O valor $\frac{20}{peso}$ representa a sensibilidade do sistema em relação ao sinal de referência e ao peso do robô. Quanto maior o peso do robô, menor será a velocidade angular gerada a partir do mesmo sinal de referência. Isso é importante para garantir um comportamento controlado do robô, levando em consideração as limitações físicas do sistema.

Portanto, a escolha do ganho $\frac{20}{peso}$ na função de transferência $G_{\Omega_{esq}}(s)$ é uma abordagem para ajustar a relação entre as entradas do sistema (sinal de referência) e as saídas desejadas (velocidade angular das rodas), levando em conta a influência do peso do robô na dinâmica do sistema.

Cálculo das Acelerações ($Acelerao_{esq}$ e $Acelerao_{dir}$):

Os blocos $Aceleração_esq$ e $Aceleração_dir$ calculam as acelerações das rodas esquerda e direita, respectivamente. Isso é feito por meio da derivada das velocidades das rodas. As derivadas representam como a velocidade está mudando ao longo do tempo, o que pode ser crucial para entender o comportamento dinâmico do robô.

As acelerações $Acelerao_{esq}$ e $Acelerao_{dir}$ são definidas pela derivada das velocidades angulares das rodas:

$$Acelerao_{esq}(s) = s \cdot \Omega_{esq}(s)$$

$$Acelerao_{dir}(s) = s \cdot \Omega_{dir}(s)$$

Onde s é a variável de Laplace e $\Omega_{esq}(s)$ e $\Omega_{dir}(s)$ são as transformadas de Laplace das velocidades angulares das rodas esquerda e direita, respectivamente.

Determinação da Velocidade Linear ($somatorio_Vr$):

O bloco $somatório_Vr$ calcula a velocidade linear do robô somando as velocidades das rodas esquerda e direita, em seguida um bloco de multiplicação efetua a divisão por 2:

$$V_r(s) = \frac{\Omega_{esq}(s) + \Omega_{dir}(s)}{2}$$

Essa velocidade impulsiona o movimento do robô em uma direção específica.

Cálculo da Diferença de Velocidades (*diferença_das_velocidades*):

O bloco *diferença_das_velocidades* calcula a diferença entre as velocidades das rodas direita e esquerda:

$$Diferenca_velocidades(s) = \Omega_{dir}(s) - \Omega_{esq}(s)$$

Essa diferença pode ser interpretada como a taxa de rotação do robô, afetando sua orientação.

Integração para o Ângulo (*integral_angulo*):

O bloco *integral_angulo* realiza a integração da diferença de velocidades em relação ao tempo, resultando no ângulo de orientação do robô:

$$\Theta(s) = \frac{1}{s} \cdot Diferenca_velocidades(s)$$

A integração acumula as variações angulares ao longo do tempo, dando uma indicação clara da direção na qual o robô está apontando.

Cálculo do Cosseno e Seno (*cosseno_direita* e *seno_direita*):

Os blocos *cosseno* e *seno* calculam o cosseno e o seno, respectivamente, da saída do bloco *integral_angulo*. Essas funções trigonométricas são úteis para representar as componentes x e y de movimento, permitindo que o robô se mova na direção desejada:

$$Cosseno(s) = \cos(\Theta(s))$$

$$Seno(s) = \sin(\Theta(s))$$

Integração para as Posições (*integral_Xcentral* e *integral_Ycentral*):

Os blocos *integral_Xcentral* e *integral_Ycentral* realizam integrações das saídas dos blocos *somatorio_Vr · cos* e *somatorio_Vr · sen*, respectivamente. Isso resulta nas posições centrais do robô ao longo dos eixos x e y , respectivamente:

$$Posicao_Xcentral(s) = \int V_r(s) \cdot \cos(\Theta(s)) ds$$

$$Posicao_Ycentral(s) = \int V_r(s) \cdot \sin(\Theta(s)) ds$$

Determinação das Posições das Rodas (*somatorio_Xrodaesquerda*, *somatorio_Xrodadireita* e *somatorio_Yrodaesquerda*, *somatorio_Yrodadireita*):

Esses blocos calculam as posições das rodas esquerda e direita em relação ao sistema de coordenadas global. As posições são obtidas considerando a posição central do robô e as distâncias entre as rodas e o centro do robô:

$$Posicao_Xrodaesquerda(s) = Posicao_Xcentral(s) - (Distancia_roda \cdot \sin(\Theta(s)))$$

$$Posicao_Xrodadireita(s) = Posicao_Xcentral(s) + (Distancia_roda \cdot \sin(\Theta(s)))$$

$$Posicao_Yrodaesquerda(s) = Posicao_Ycentral(s) - (Distancia_roda \cdot \cos(\Theta(s)))$$

$$Posicao_Yrodadireita(s) = Posicao_Ycentral(s) + (Distancia_roda \cdot \cos(\Theta(s)))$$

Determinação da Direção (*somatorio_Xdirecao* e *somatorio_Ydirecao*):

Esses blocos calculam pequenos deslocamentos que indicam a direção para a qual o robô está apontando. Isso é feito adicionando pequenas componentes às posições centralizadas do robô nas direções x e y :

$$Posicao_Xdirecao(s) = Posicao_Xcentral(s) + \cos(\Theta(s))$$

$$Posicao_Ydirecao(s) = Posicao_Ycentral(s) + \sin(\Theta(s))$$

Apresentação Visual (*bloco_final* - *BARXY SCOPE*):

O bloco *bloco_final* exibe visualmente as posições calculadas do robô no plano XY. Ele representa graficamente a trajetória percorrida pelo robô, demonstrando como o sistema de controle está atuando para cumprir a trajetória desejada.

4 Requisitos

Com a implementação feita, criam-se requisitos que serão utilizados para verificar o modelo. Estes tem intuito de claramente demonstrar que o modelo não está funcionando caso falhem, porém, não garantem que o modelo está funcionando caso funcionem.

Estuda-se quatro condições diferentes, com combinações diferentes de velocidade e sentido das rodas.

4.1 Requisito 1: Ambas velocidades iguais e mesmo sentido

Neste requisito, espera-se que a variável de estado da posição do carro varie de forma linear quando em regime permanente.

Também espera-se que a variável de estado do ângulo mantenha-se constante, ou seja, que a velocidade angular mantenha-se nula.

4.2 Requisito 2: Ambas velocidades iguais, sentidos opostos.

Espera-se desta situação que as variáveis de estado designadas a denotar a posição central do carrinho se mantenham constantes, enquanto aquela que denota a posição angular varia de forma linear, ou seja, o carrinho deve responder com velocidade angular constante de módulo igual a razão entre módulo da velocidade das rodas e a metade do comprimento do eixo.

4.3 Requisito 3: Uma velocidade igual a zero, a outra diferente de zero

Para a situação em que uma roda se encontra com o motor desligado e apenas o motor da outra roda atua com uma velocidade constante e não nula, pode-se esperar que a posição central do carrinho descreva uma circunferência de raio igual a metade do comprimento do eixo do carro, enquanto sua posição angular varie com velocidade constante de módulo igual a razão entre o módulo da velocidade da roda e o comprimento do eixo.

4.4 Requisito 4: Ambas velocidades diferentes e não nulas

Para a situação em que as rodas possuem velocidades de módulos diferentes, pode-se esperar que o comportamento seja similar ao caso anterior, porém, a velocidade de referência será o módulo da diferença entre as velocidades das rodas.

5 Verificação

Nesta seção, colocamos em prática os conceitos discutidos na análise teórica, visando criar testes sistemáticos que validem a consistência da nossa modelagem e atendam nossos requisitos.

Utilizaremos *Godot* e as linguagens *C#* e *GDScript* para conduzir nossa verificação.

5.1 Parâmetros de simulação

Ao modelarmos um sistema analítico em um ambiente computacional, é essencial contemplar os erros de arredondamento decorrentes da precisão dos números de ponto flutuante.

Em outras palavras, é imperativo evitar o uso de valores e intervalos que se aproximem dos limites de precisão do tipo de variável empregado no sistema.

Tendo essa perspectiva em mente, adotamos um intervalo de tempo de $1/240$ segundos para a nossa simulação. Além disso, levamos em consideração os seguintes parâmetros:

- Tempo de aceleração = 500
- Peso do carro = 200g
- Diâmetro da roda = 50mm
- Diâmetro entre eixos = 200mm

- Velocidade máxima = 628.318 mm/s

Com os parâmetros estabelecidos, estamos aptos a testar o veículo em uma ampla variedade de configurações possíveis, assegurando, dessa forma, que nossos requisitos sejam atendidos com precisão.

5.2 Requisitos a serem verificados

Com esses parâmetros em mãos, podemos realizar a verificação dos nossos requisitos em diversas situações, assegurando a coerência do nosso modelo.

Para simplificar nossa verificação, não nos concentraremos no comportamento do modelo durante sua fase transitória; focaremos apenas em observar como ele se comporta em seu estado permanente.

5.2.1 Requisito 1: Ambas velocidades iguais e mesmo sentido

Neste caso, a expectativa é que a variação da variável de estado da posição a cada segundo se aproxime do valor da velocidade máxima.

Além disso, espera-se que a velocidade angular permaneça em zero durante todo o teste.

Vamos, então, analisar a distância percorrida pelo carro nos eixos X e Y quando a variável θ inicia-se em zero e se mantém assim durante um período de cinco segundos em regime permanente.

Tempo (s)	Distancia Percorrida X (mm)	Distancia Percorrida Y (mm)
1	628.31	0
2	628.31	0
3	628.31	0
4	628.31	0
5	628.31	0

Observa-se, portanto, que a distância percorrida pelo carro na direção $\theta = 0$ se aproxima da sua velocidade máxima, e não ocorre variação na posição em nenhuma outra direção, exceto na direção θ .

5.2.2 Requisito 2: Ambas velocidades iguais, sentidos opostos

Neste teste, aplicamos impulsos positivos a uma das rodas e impulsos negativos à outra, com o propósito de fazer o veículo girar sem deslocar o seu ponto central.

Nossa expectativa é que o veículo alcance sua rotação máxima, sem efetuar movimentos do seu ponto central.

Para avaliar isso, focaremos na posição central, que deve permanecer rigidamente na coordenada $(0,0)$, e analisaremos o tempo que o modelo leva para completar uma rotação total.

Dessa forma, analisaremos a posição nas coordenadas X e Y e o tempo necessário para uma rotação completa do carro durante um período de 5 segundos, em regime permanente.

Tempo (s)	Pos X (mm)	Pos Y (mm)	Tempo de rotacao (s)
1	0	0	1.00
2	0	0	1.00
3	0	0	1.00
4	0	0	1.00
5	0	0	1.00

Observa-se que, conforme esperado com base em nossos parâmetros, o carro realiza uma rotação a cada segundo, e o seu centro permanece estático. Desta forma, confirmamos a verificação deste requisito.

5.3 Requisito 3: Uma velocidade igual a zero, a outra diferente de zero

Neste exemplo, vamos analisar separadamente a variação da posição central do carro e o tempo de rotação do veículo.

Com os parâmetros utilizados, é esperado que a posição X do carro varie entre $\pm \frac{dimetro_{do_eixo}}{2}$ e que a posição Y varie entre 0 e $-dimetro_{do_eixo}$.

Além disso, antecipamos que sua velocidade angular seja de $\frac{628.31}{200} * 0.5$, o que equivale a uma rotação completa a cada 2 segundos.

No teste, podemos observar que são alcançados os seguintes cinco pares de coordenadas:

Pos X (mm)	Pos Y (mm)
0	0
100	-100
0	-200
-100	0

Também observamos que esses são, respectivamente, os valores máximos e mínimos de X e Y , o que confirma o comportamento esperado da posição do centro do carro.

Além disso, notamos o período de rotação de 2 segundos, o que reforça a consistência da velocidade angular.

5.4 Requisito 4: Ambas velocidades diferentes e não nulas

Neste requisito, iremos verificar o período de rotação do carro para velocidades arbitrariamente ajustáveis durante a execução do teste.

A seguir, listaremos os períodos obtidos para diferentes pares de velocidades nas rodas.

V Roda Esquerda (mm/s)	V Roda Direita (mm/s)	Periodo de rotacao (s)
596.38	268.29	3.83
540.19	401.01	11.84
347.11	-213.87	2.24
-83.23	-451.09	3.74

Podemos observar a relação obtida na fundamentação teórica:

$$V_{\theta} = (VRodaEsquerda - VRodaDireita) * \frac{1}{eixo_das_rodas}$$
$$T = \frac{2\pi}{V_{theta}}$$

No decorrer dos nossos testes e observações, podemos notar que o período de rotação, representado como T , permanece invariável. Isso se evidencia quando calculamos as velocidades das rodas em diversas situações. Essa consistência na manutenção do período de rotação nos permite confirmar, com segurança, que o requisito relacionado a essa característica está plenamente atendido e verificado em nosso experimento.

6 Projeto Final do Controlador/Compensador

O projeto final do controlador/compensador é o resultado da combinação entre a teoria de controle discutida anteriormente e as considerações práticas, envolvendo componentes eletrônicos, características elétricas e custos. Nesta seção, detalharemos as escolhas de componentes, as características elétricas relevantes e uma estimativa de custo para a implementação do sistema de controle do robô.

6.1 Seleção de Componentes

Para a implementação do controlador/compensador, foram selecionados os seguintes componentes:

- Microcontrolador: Arduino Uno
- Sensores: Encoders de roda
- Atuadores: Motores DC com controle de velocidade
- Placa de Controle de Motor: L298N

O microcontrolador Arduino Uno é uma escolha comum devido à sua facilidade de programação e ampla disponibilidade. Os encoders de roda fornecem feedback de posição, enquanto os motores DC controlados permitem ajuste fino da velocidade das rodas. O L298N é um circuito integrado que atua como uma ponte H, comumente utilizado para controlar motores DC (corrente contínua) bidirecionalmente. Ele é amplamente utilizado em projetos de robótica e automação, pois permite controlar a velocidade e a direção de rotação dos motores de forma eficiente. A ponte H é um arranjo de transistores que permite direcionar a corrente de forma a controlar a polaridade dos terminais do motor, permitindo que ele gire em ambos os sentidos.

6.2 Características Elétricas

As características elétricas dos componentes selecionados são:

- Microcontrolador (Arduino Uno):
 - Tensão de Operação: 5V
 - Corrente de Operação: 50mA (aproximadamente)
- Sensores (Encoders):
 - Tensão de Operação: 5V
 - Corrente de Operação: 10mA (por encoder)
- Atuadores (Motores DC):
 - Tensão de Operação: 6V
 - Corrente de Operação: 300mA (por motor em carga máxima)

6.3 Implementação do Circuito

A Figura 2 mostra o diagrama elétrico simplificado da implementação do controlador/compensador com os componentes selecionados. O microcontrolador Arduino Uno controla os motores DC através da placa L298N, que atua como ponte H para direcionar a energia para as rodas em diferentes direções. Os encoders fornecem feedback de posição para o microcontrolador, permitindo o controle preciso da velocidade das rodas.

Note que foi utilizado um L293D nesse diagrama, único disponível na plataforma utilizada. Este exemplo serve apenas como uma ideia de como o projeto poderia ser implementado.

6.4 Estimativa de Custo

Com base nas escolhas de componentes e características elétricas, a estimativa de custo para a implementação do controlador/compensador é a seguinte:

- Microcontrolador (Arduino Uno): \$20
- Sensores (Encoders): \$10 (por encoder, 2 encoders no total)
- Atuadores (Motores DC): \$25 (por motor, 2 motores no total)
- Placa de Controle de Motor (L298N): \$5
- Outros componentes e materiais: \$15

O custo total estimado para a implementação do sistema de controle é de aproximadamente \$85.

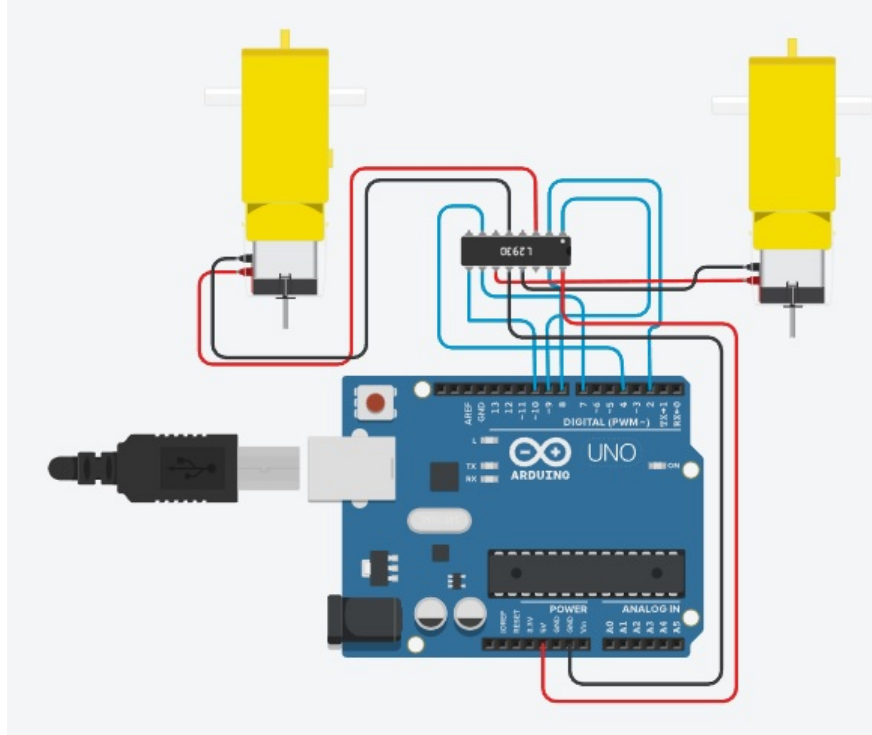


Figura 2: Diagrama elétrico simplificado da implementação do controlador/compensador.

6.5 Considerações Finais

O projeto final do controlador/compensador foi elaborado considerando a seleção de componentes adequados, características elétricas, integração dos elementos e custos envolvidos. A implementação do circuito proporciona um controle preciso da velocidade das rodas e permite o monitoramento da posição através dos encoders. A estimativa de custo fornecida serve como referência, mas os preços podem variar conforme a disponibilidade e fornecedores locais.

7 Conclusão

Ao longo deste projeto, tivemos a oportunidade de aplicar os conceitos discutidos em nossa disciplina de servomecanismo de maneira prática e abrangente. Utilizando o Godot como ambiente de simulação e as linguagens C e GDScript para programar, conseguimos verificar a validade de nossos modelos teóricos através de testes sistemáticos.

Exploramos a resposta de sistemas de primeira ordem em relação ao tempo, compreendendo como o comportamento de um sistema é influenciado por parâmetros como o tempo de subida linear. Ao analisar a velocidade angular e linear do veículo em diferentes cenários, pudemos verificar que nossos resultados práticos concordavam com as previsões matemáticas.

Além disso, incorporamos a consideração de erros de arredondamento ao modelar o sistema em um ambiente computacional. Isso nos ajudou a compreender a importância de escolher intervalos de tempo adequados e parâmetros realistas para simulações.

Os testes realizados, como o teste de velocidade central e o teste de rotação máxima, reforçaram a consistência e precisão de nossos modelos. Observamos que as respostas reais das variáveis de estado do veículo estavam de acordo com nossas previsões teóricas, o que demonstra que nossos esforços na modelagem e implementação foram bem-sucedidos.

Nossa experiência em simular e testar o sistema também nos proporcionou insights sobre possíveis aplicações práticas. Podemos imaginar a utilização desse tipo de controle em robôs móveis, drones ou até mesmo em sistemas de posicionamento e estabilização.

Em termos de aperfeiçoamento, reconhecemos a importância de expandir nossa análise para situações mais complexas, incluindo interações com ambientes dinâmicos e variações nos parâmetros do sistema.

Em resumo, este projeto nos proporcionou uma valiosa experiência na aplicação de conceitos teóricos em situações práticas, reforçando nossa compreensão dos sistemas de controle e sua implementação. Através da simulação e análise, pudemos verificar a eficácia de nossos modelos e obter insights valiosos para futuras aplicações.