

# Relatório - Comparação de Modelagem Dinâmica em Ambiente de Simulação Robótica para Robôs Móveis de Tração Diferencial

Mateus Pincho de Oliveira  
Prof. Antônio Marcus Nogueira Lima, Automação Inteligente 25.2

November 20, 2025

## Abstract

Este relatório tem por objetivo descrever e analisar a implementação de um modelo dinâmico de um robô de tração diferencial. Em seguida, comparar a solução numérica do modelo dinâmico com os resultados de uma simulação na plataforma CoppeliaSim.

## I. INTRODUÇÃO

**O** Controle de robôs móveis com rodas depende intrinsecamente da modelagem matemática escolhida para analisar a dinâmica do sistema. O modelo dinâmico de um sistema analisa a movimentação do corpo a partir das forças que atuam sobre o mesmo. Para a classe de robôs móveis de tração diferencial, sua dinâmica é altamente não-linear e envolve restrições não-holônomicas, que devem ser incluídas na análise do seu movimento.

Este relatório tem por objetivo descrever a implementação computacional e a análise do modelo dinâmico de um robô de tração diferencial, modelo Pioneer P3DX, fazendo uma comparação com o ambiente de simulação robótica CoppeliaSim. O trabalho baseia-se nas formulações de Lagrange para derivar as equações de movimento que relacionam as tensões de armadura dos motores CC com as acelerações e velocidades do robô.

## II. METODOLOGIA

Um robô é um agente capaz de se movimentar pelo ambiente. Para localizá-lo no espaço, é necessário definir um sistema coordenado de referência. O sistema de coordenadas inercial ( $I$ ) representa a referência global do ambiente, fixada no plano em que o robô se movimenta. Comumente, também utiliza-se o sistema de coordenadas local, anexado sobre o centro de massa do robô e denotado por  $R$ . A configuração - ou pose - do robô é dada pela combinação da posição  $(x, y)$  e a orientação  $\theta$  para determinado referencial.

Para a classe de robôs móveis de tração diferencial, existe uma série de restrições relacionadas ao movimento passível de ser realizado pelo robô, que devem ser consideradas durante a análise da cinemática e dinâmica desta classe de robôs móveis. A primeira restrição diz que não há movimento de deslizamento lateral, em que o robô é capaz apenas de atuar direção longitudinal do seu eixo de tração. A segunda restrição diz que as rodas do robô funcionam no mecanismo de rolamento puro, onde cada roda toca unicamente um ponto quando em contato com o solo. Não há deslizamento das rodas no eixo longitudinal, bem como no eixo ortogonal.

O estudo do movimento do robô, sem considerar as forças que atuam sobre o sistema, leva a obtenção de um modelo cinemático. Esta análise foca apenas em entender como a velocidade das rodas do robô impactam na sua pose final no sistema de coordenadas inercial. Para a realização de análises de simulações detalhadas, a modelagem dinâmica se torna essencial, pois considera-se as forças que atuam sobre o sistema. Utilizando a abordagem proposta por Lagrange, que considera a energia cinética e potencial armazenada no sistema, é possível derivar um modelo de equações diferenciais para representar o comportamento da dinâmica do robô. As forças atuantes no sistema são geradas por um conjunto de motores de corrente contínua controlados pelo circuito de armadura.

Dessa forma, para o sistema robótico em questão observa-se o seguinte sistema de equações diferenciais:

$$\begin{bmatrix} \dot{i}_{a_D} \\ \dot{i}_{a_E} \end{bmatrix} = \begin{bmatrix} \frac{1}{L_a} \left( v_{a_D} - K_b N \dot{\phi}_D - R_a i_{a_D} \right) \\ \frac{1}{L_a} \left( v_{a_E} - K_b N \dot{\phi}_E - R_a i_{a_E} \right) \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} = \begin{bmatrix} \frac{R \cos(\theta)}{2} \left( \dot{\phi}_D + \dot{\phi}_E \right) \\ \frac{R \sin(\theta)}{2} \left( \dot{\phi}_D + \dot{\phi}_E \right) \\ \frac{R}{2L} \left( \dot{\phi}_D - \dot{\phi}_E \right) \end{bmatrix}$$

$$\begin{bmatrix} \ddot{\phi}_D \\ \ddot{\phi}_E \end{bmatrix} = \bar{M}^{-1}(q) \left( -\bar{V}(q, \dot{q}) \begin{bmatrix} \dot{\phi}_D \\ \dot{\phi}_E \end{bmatrix} + \bar{B}(q)\tau \right)$$

onde

$$\begin{aligned} \tau &= K_t N [i_{aD} \quad i_{aE}]^T \\ \bar{M}(q) &= \begin{bmatrix} I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 - I) & I_w + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix} \\ \bar{V}(q, \dot{q}) &= \begin{bmatrix} 0 & \frac{R^2}{2L} m_c d \dot{\theta} \\ -\frac{R^2}{2L} m_c d \dot{\theta} & 0 \end{bmatrix} \\ \bar{B}(q) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Para o caso implementado e simulado com o robô P3DX, considerou-se os seguintes parâmetros:

- $R_a = 0,71 \Omega$ ;  $L_a = 0,66 \text{ mH}$ ;  $K_b = 0,023 \text{ V/rad} \cdot \text{s}$ ;  $K_t = 0,029 \text{ N} \cdot \text{m/A}$ ; e  $N = 38,3$ .
- $R = 0,1 \text{ m}$ ;  $L = 0,19 \text{ m}$  e  $d = 0,05 \text{ m}$ .
- $m_w = 1 \text{ Kg}$ ;  $m_c = 7 \text{ Kg}$ ;  $I_m = 0,0025 \text{ Kg} \cdot \text{m}^2$ ;  $I_r = 0,005 \text{ Kg} \cdot \text{m}^2$  e  $I_c = 1 \text{ Kg} \cdot \text{m}^2$ .

A implementação computacional do modelo dinâmico foi desenvolvida em linguagem Python, estruturando-se o sistema como um conjunto de equações diferenciais ordinárias (EDOs) acopladas que descrevem simultaneamente a dinâmica elétrica dos atuadores e a dinâmica mecânica. Para a resolução numérica deste sistema, utilizou-se a biblioteca científica SciPy, especificamente a função solve\_ivp, configurada com o método de Runge-Kutta (RK45). A rotina de simulação recebe como entrada os parâmetros físicos do robô Pioneer P3DX e o vetor de tensões de armadura, atualizando iterativamente o vetor de estados — composto pelas velocidades angulares das rodas, correntes elétricas e a pose global  $(x, y, \theta)$  — para computar a trajetória resultante ao longo do tempo de simulação.

Para fins de comparação, utilizou-se o software de simulação robótica CoppeliaSim. O cenário experimental foi configurado com o modelo virtual do robô Pioneer P3DX, conforme mostra a Figura 1, posicionado sobre uma superfície plana sujeita a atrito e gravidade, regida pelo motor de física Bullet 2.78. A comunicação entre o algoritmo desenvolvido em Python e o simulador foi estabelecida através da *ZMQ Remote Api*, configurada em modo de operação síncrono. Esta configuração garante que cada passo de integração do modelo matemático corresponda exatamente a um passo de tempo fixo ( $\text{dt}=50\text{ms}$ ) na simulação física. A cada iteração, as velocidades angulares calculadas pelo modelo dinâmico são enviadas como sinais de controle de velocidade para as juntas dos motores virtuais, e a pose resultante  $(x, y, \theta)$  é recuperada via API para análise comparativa das trajetórias.

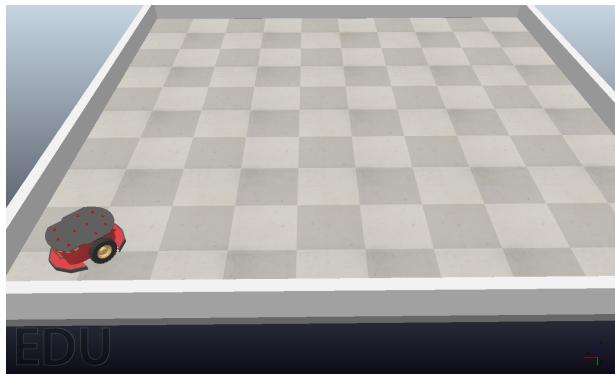


Fig. 1: Ambiente de testes em simulação física no software CoppeliaSim

### III. RESULTADOS E DISCUSSÕES

Para validar o modelo dinâmico implementado e analisar a fidelidade da resposta do sistema frente a diferentes comportamentos, foram realizados cinco experimentos distintos. A metodologia consistiu em aplicar degraus de tensão como entrada no modelo matemático (Python), calcular as velocidades angulares resultantes e utilizá-las como setpoint de controle para as juntas do robô no ambiente CoppeliaSim.

Os cenários experimentais foram definidos da seguinte forma:

- 1) Movimento Retilíneo: Três testes com tensões simétricas aplicadas a ambas as rodas ( $v_D = v_E$ ) com amplitudes de 3V, 6V e 9V, respectivamente.
- 2) Movimento Curvilíneo: Aplicação de tensão diferencial, sendo 3.0V na roda esquerda e 3.5V na roda direita.
- 3) Movimento Reverso: Aplicação de tensão simétrica negativa de -3V em ambas as rodas.

### A. Análise do Movimento e Perturbações

Nos experimentos de movimento em linha reta, a simulação do modelo dinâmico no ambiente Python resultou na geração de trajetórias retilíneas. Pelas equações de movimento, quando as tensões são idênticas, as velocidades angulares convergem para o mesmo valor, resultando em velocidade angular do robô  $\omega$  nula. Esse comportamento pode ser visto nas Figura 2

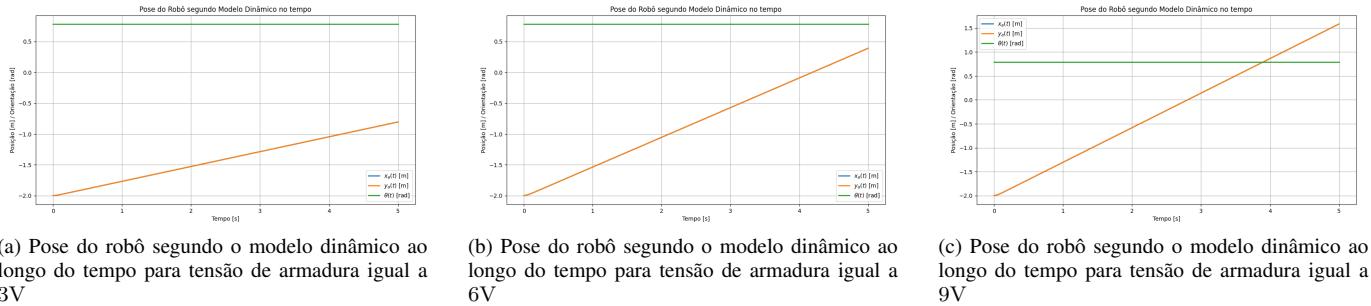


Fig. 2: Comparação dos três cenários para o modelo dinâmico no ambiente Python: (a) 3V, (b) 6V e (c) 9V.

As velocidades angulares das rodas esquerda e direita utilizadas para a simulação no ambiente do CoppeliaSim são calculadas a partir do modelo dinâmico implementado. As velocidades utilizadas nos cenários experimentais estão mostradas nas Figura 3

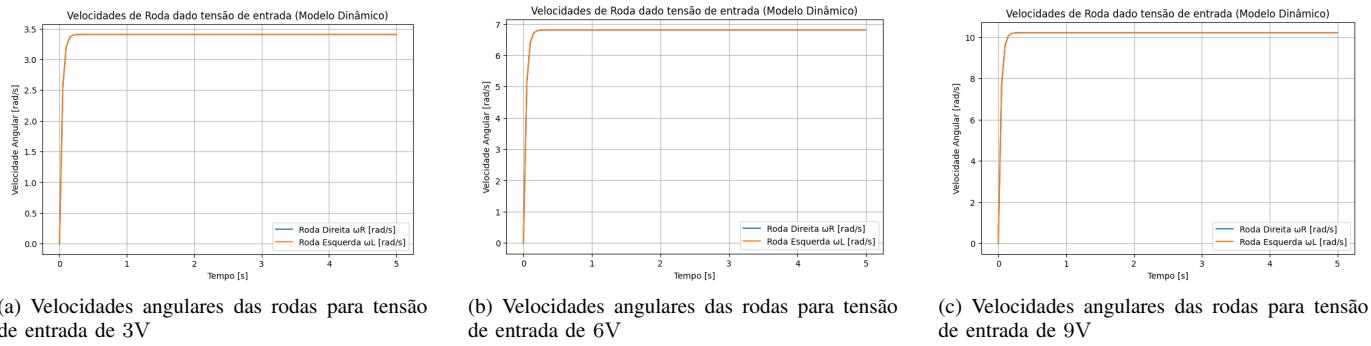


Fig. 3: Velocidades angulares de rodas para o movimento retilíneo

Ao comparar a pose do robô com a simulação no CoppeliaSim, observou-se uma divergência nas trajetórias XY, caracterizada por um deslocamento lateral (*drift*), como pode ser visto nas Figura 4. Essa discrepância é atribuída às simplificações adotadas na modelagem dinâmica, que desconsidera perturbações físicas presentes no simulador. É possível visualizar a evolução do erro da pose calculada pelo modelo dinâmico na Figura 5. O erro de posição linear foi determinado pela distância Euclidiana entre as coordenadas ( $x, y$ ) calculadas pelo modelo e as obtidas via API do CoppeliaSim. Simultaneamente, o erro de orientação foi calculado pela diferença escalar entre os ângulos  $\theta$  de ambos os modelos. Para verificar a precisão de cada experimento, computou-se a Raiz do Erro Quadrático Médio (RMSE) para ambas as métricas ao longo de toda a trajetória e estão evidenciados na Tabela I.

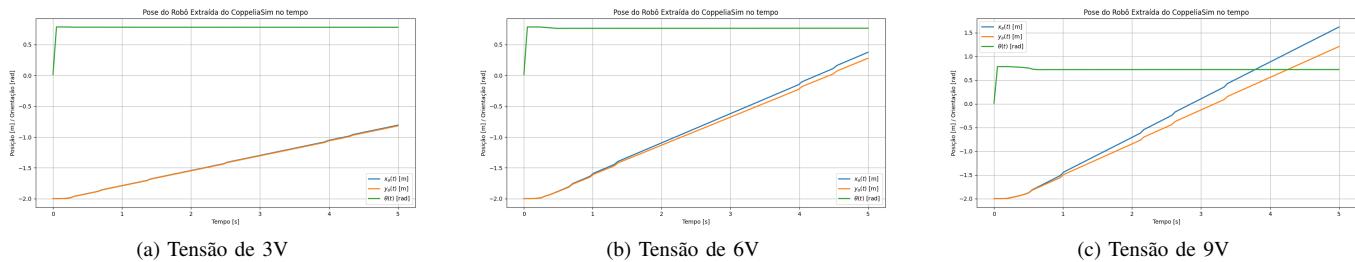


Fig. 4: Comparação da pose do robô obtida no CoppeliaSim ao longo do tempo para diferentes tensões de armadura: (a) 3V, (b) 6V e (c) 9V.

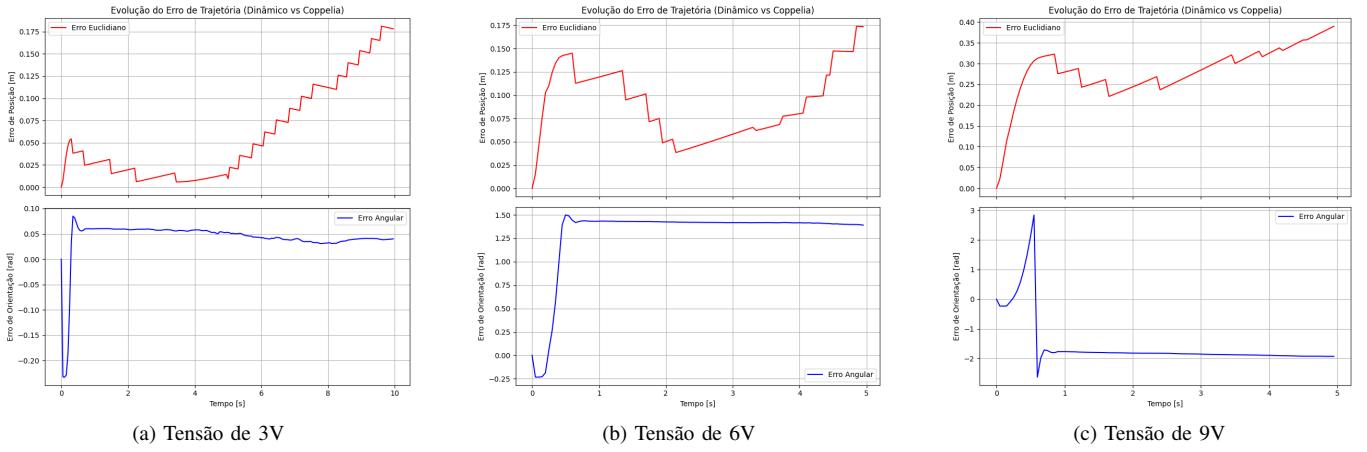


Fig. 5: Comparação do erro pose do robô calculada pelo modelo dinâmico e a obtida no CoppeliaSim ao longo do tempo para diferentes tensões de armadura: (a) 3V, (b) 6V e (c) 9V.

Experimento	3V	6V	9V
RMSE Posição	0.0798	0.0976	0.288
RMSE Orientação	0.0585	1.3610	1.17919

TABLE I: Erro RMSE de Posição e Orientação para os três experimentos de movimento retilíneo

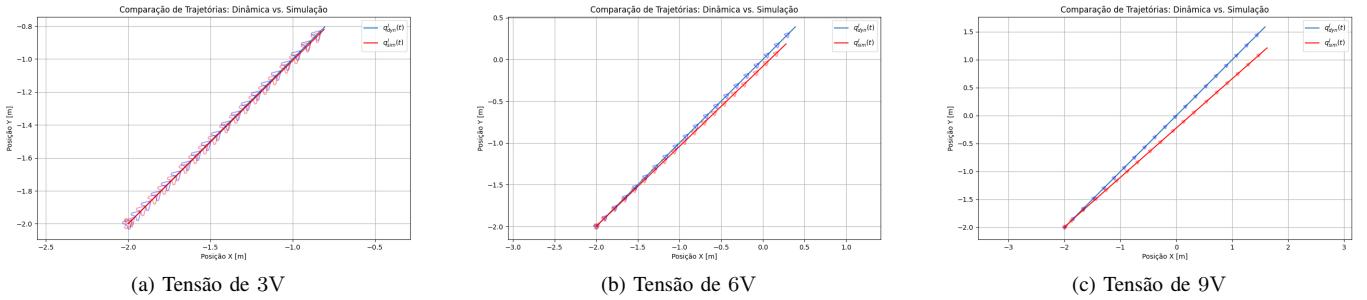


Fig. 6: Comparação das trajetórias realizadas pelo robô no plano de fase (modelo dinâmico versus simulação) para diferentes tensões de armadura: (a) 3V, (b) 6V e (c) 9V.

O fator preponderante identificado para esse desvio foi a dinâmica da roda de apoio (*caster wheel*). No início do movimento, a roda caster no simulador precisa rotacionar sobre seu próprio eixo para se alinhar à direção do movimento. Durante esse breve intervalo, ela exerce uma força lateral e um torque resistivo que não existem no modelo matemático. Isso provoca uma pequena variação no momento de inércia do chassis, causando uma variação inicial na orientação  $\theta$  do robô no simulador, que, ao ser integrada ao longo do tempo, resulta no erro de posição lateral acumulado observado nos gráficos comparativos do plano de fase, mostrado nas Figuras 6.

Estas diferenças tornam-se ainda mais significativas quando analisa-se o caso em que as rodas possuem tensões de armadura distintas, bem como para o movimento retilíneo contrário com tensão de armadura negativa. Para estes dois casos, a roda passiva necessita se realinear com o eixo do robô, o que provoca o deslocamento lateral acentuado, mostrado nas Figuras 7.

Para determinar o valor limite de velocidade angular das rodas, o suficiente para gerar discrepâncias significativas entre os modelos comparados, foi realizado o seguinte experimento: aplicou-se uma rampa de tensão até o limite de saturação dos motores (12V) e verificou-se qual o valor de velocidade angular necessário para atingir um erro de posição entre o modelo dinâmico e a simulação física maior que 5cm.

Constatou-se que, para atingir um erro de posição de 5 cm, é necessário que as rodas estejam com velocidade angular de 3.55 rad/s, como pode ser visto na Figura 8. Dessa forma, para que seja possível utilizar o modelo dinâmico desprezando os efeitos de outras forças resistivas, bem como da roda de apoio não modelada, deve-se limitar a velocidade angular das rodas ao valor limiar encontrado.

A comparação das trajetórias nestes cenários corroborou com a hipótese de que as forças de não modeladas atuantes na roda passiva e o atrito são as principais causas da divergência entre o modelo analítico e a simulação física. Apesar dessas

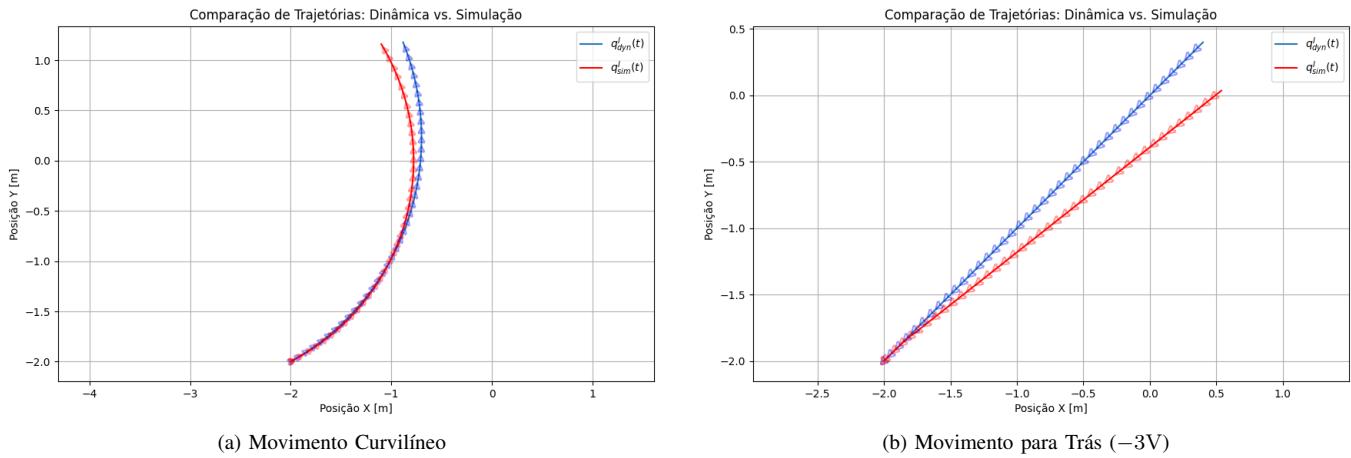


Fig. 7: Análise das trajetórias no plano de fase para cenários com maior influência de perturbações: (a) movimento curvilíneo (tensão diferencial) e (b) movimento reverso (realinhamento da roda caster).

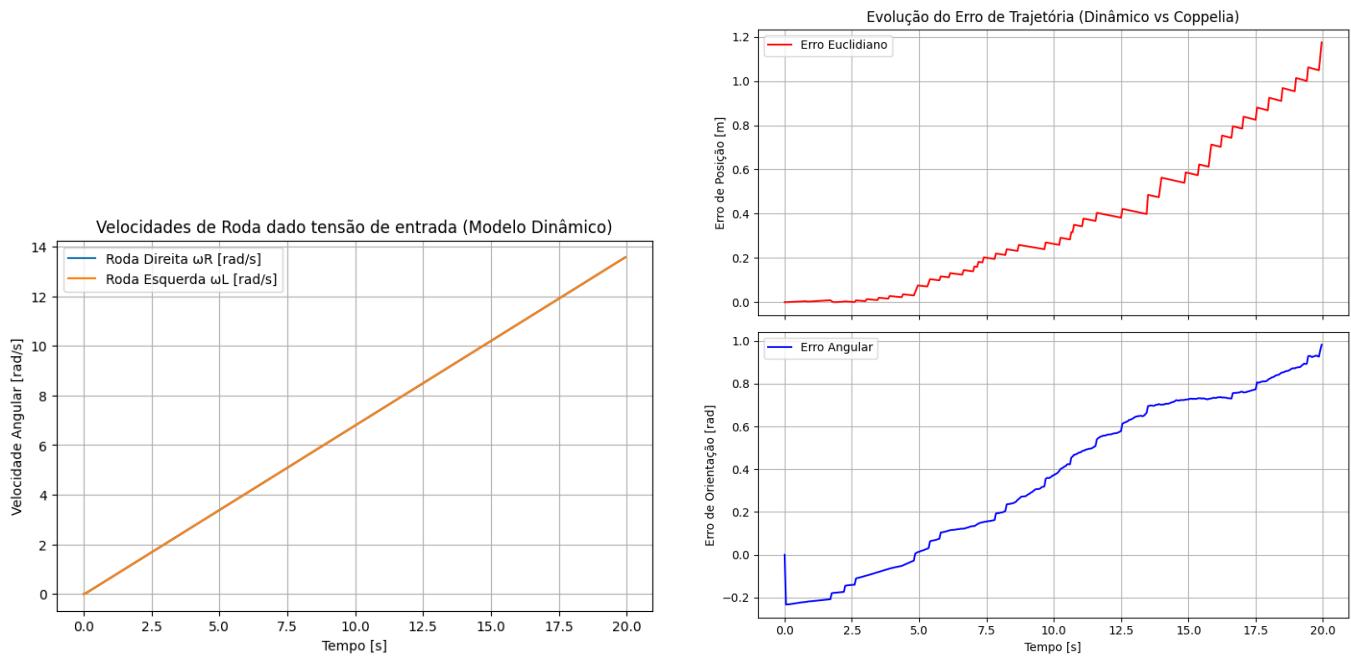


Fig. 8: Análise das trajetórias no plano de fase para cenários com maior influência de perturbações: (a) movimento curvilíneo

perturbações não modeladas, o modelo dinâmico foi capaz de prever com coerência a magnitude das velocidades e a tendência geral do movimento em todos os cinco cenários testados. É possível assumir, que para baixas velocidades, o modelo dinâmico deve desconsiderar a rede de apoio e o atrito é suficiente para descrever com consistência o movimento da cabra no ambiente.

#### IV. Conclusões

#### IV. CONCLUSÃO

Este relatório apresentou a implementação computacional e a análise do modelo dinâmico de um robô móvel de tração diferencial, utilizando a formulação de Lagrange para descrever o sistema eletromecânico. A validação do modelo foi realizada através de uma comparação direta com o ambiente de simulação física CoppeliaSim, submetendo ambos os sistemas a diferentes níveis de tensão de armadura. Os resultados obtidos demonstraram que o modelo matemático implementado em Python é consistente e capaz de prever as velocidades de regime e o comportamento geral do robô em condições ideais. No entanto, a comparação com a simulação física no CoppeliaSim evidenciou divergências nas trajetórias, como deslocamentos laterais (*drift*).

A análise destes erros confirmou que as simplificações adotadas na modelagem analítica — especificamente a desconsideração das forças dissipativas de atrito e da dinâmica da roda de apoio (*caster wheel*) — são significativas.

#### REFERENCES

- [1] R. Dhaouadi and A. A. Hatab, "Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework," *Adv. Robot. Autom.*, vol. 2, no. 2, p. 107, Sep. 2013, doi: 10.4172/2168-9695.1000107.