

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

JOÃO LUCAS FOLTRAN CONSONNI

Reguladores Quadráticos e Odometria Visual para Controle de
Veículos Autônomos

São Carlos
2024

Reguladores Quadráticos e Odometria Visual para Controle de Veículos Autônomos

Monografia apresentada ao Curso de Engenharia Elétrica - Ênfase em Sistemas de Energia e Automação, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do Título de Engenheiro Eletricista

Orientador: Prof. Dr. Marco Henrique Terra

São Carlos
2024

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

C755r Consonni, João Lucas Foltran
 Reguladores Quadráticos e Odometria Visual para
 Controle de Veículos Autônomos / João Lucas Foltran
 Consonni; orientador Marco Henrique Terra. São Carlos,
 2024.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Sistemas de Energia e Automação) -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2024.

1. Veículos Autônomos. 2. Reguladores Lineares
Quadráticos. 3. Controle Robusto. 4. Odometria Visual.
5. Marcadores Fiduciais. 6. Operação de Transbordo. I.
Título.

FOLHA DE APROVAÇÃO

Nome: João Lucas Foltran Consonni

Título: “Reguladores Quadráticos e Odometria Visual para Controle de Veículos Autônomos”

**Trabalho de Conclusão de Curso defendido e aprovado em
26/11/2024,**

com NOTA 9 (Nove), pela Comissão Julgadora:

Prof. Titular Marco Henrique Terra - Orientador - SEL/EESC/USP

Prof. Adjunto Lucas Barbosa Marcos - UFSCar

Mestre Eduardo Godinho Ribeiro - Doutorando EESC/USP

**Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Professor Associado José Carlos de Melo Vieira Júnior**

AGRADECIMENTOS

Em primeiro lugar, a Deus, pela minha vida, por todas as oportunidades concedidas e por ter me dado forças para superar todos os obstáculos.

Aos meus familiares, por todo o apoio e sacrifícios que fizeram para eu estar nesta universidade e que contribuíram para o desenvolvimento deste trabalho.

Aos amigos, que me deram alegria nos momentos de frustração e ajuda nos momentos de dificuldade no decorrer deste trabalho.

Ao professor Marco Henrique Terra, por ter sido o meu orientador e me guiado no decorrer do desenvolvimento desta monografia.

Aos colegas do Laboratório de Sistemas Inteligentes (LASI), que me ajudaram a solucionar problemas difíceis no decorrer do trabalho.

A todos aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

RESUMO

CONSONNI, J. L. F. Reguladores Quadráticos e Odometria Visual para Controle de Veículos Autônomos. 2024. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

O avanço no campo de veículos autônomos leva a um cenário de possível diminuição em acidentes automobilísticos e expansão das aplicações para diversas áreas, como na agricultura. Entretanto, sua segurança ainda é questionada devido a alguns acidentes causados por esses veículos. Diante disso, com o objetivo de tornar o tráfego desse tipo de veículo mais seguro, neste projeto foi proposto o desenvolvimento de um controlador, utilizando o regulador linear quadrático e sua versão robusta. Visando a utilização de apenas um tipo de sensor no veículo, a estimativa de estados foi feita por meio de técnicas de odometria visual e utilização de marcadores fiduciais, especificamente, marcadores ArUco. A aplicação proposta foi a operação de transbordo em uma plantação de cana-de-açúcar, na qual um caminhão deve seguir lateralmente a colheitadeira. Os testes foram realizados em simulações, no *software* Carla, e em laboratório, utilizando dois robôs que representam a colheitadeira e o caminhão. Os resultados obtidos não atingiram erro de regime nulo devido a erros associados à detecção do marcador ArUco e estimativa de estados. Concluiu-se que os controladores desenvolvidos funcionam, de forma geral, e foram avaliadas soluções para os problemas que limitaram seu desempenho e eficácia.

Palavras Chaves: **Veículos Autônomos, Reguladores Lineares Quadráticos, Controle Robusto, Odometria Visual, Marcadores Fiduciais, Operação de Transbordo.**

ABSTRACT

CONSONNI, J. L. F. Quadratic Regulators and Visual Odometry for Autonomous Vehicles Control. 2024. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

Advances in the field of autonomous vehicles are leading to a scenario of a possible reduction in car accidents and the expansion of applications to various areas, such as agriculture. However, their safety is still questioned due to some accidents caused by these vehicles. Aiming to make this type of vehicle's traffic safer, this project proposes the development of a controller using the quadratic linear regulator and its robust version. In order to use only one type of sensor in the vehicle, state estimation was carried out using visual odometry techniques and fiducial markers, specifically ArUco markers. The proposed application was a transshipment operation in a sugarcane plantation, in which a truck must follow the harvester laterally. The tests were carried out in simulations, in the software Carla, and in the laboratory, using two robots representing the harvester and the truck. The results obtained did not reach a null steady-state error due to errors associated with ArUco marker detection and state estimation. It was concluded that the controllers developed work, in general, and solutions were evaluated for the problems that limited their performance and effectiveness.

Keywords: Autonomous Vehicles. Quadratic Linear Regulators. Robust Control. Visual Odometry. Fiducial Markers. Transshipment Operation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Caminhão autônomo Scania em plantação de cana-de-açúcar	18
Figura 2 – Modelo de bicicleta dos veículos	24
Figura 3 – Modelo cinematográfico	25
Figura 4 – Imagem para calibração de câmera	28
Figura 5 – Mudança de coordenadas	29
Figura 6 – Estimativa da pose de ArUcos	30
Figura 7 – Percursos de simulação	34
Figura 8 – Cenário de movimento circular	35
Figura 9 – Robôs utilizados em laboratório	36
Figura 10 – Eixos do sistema de coordenadas utilizado	38
Figura 11 – Cenário 1 - Linha reta - LQR	43
Figura 12 – Cenário 1 - Linha reta - RLQR	44
Figura 13 – Cenário 1 - Caminho percorrido	45
Figura 14 – Cenário 2 - Rotatória - LQR	46
Figura 15 – Cenário 2 - Rotatória - RLQR	47
Figura 16 – Cenário 2 - Caminho percorrido	48
Figura 17 – Cenário 3 - Trajeto curto com retas e curvas - LQR	49
Figura 18 – Cenário 3 - Trajeto curto com retas e curvas - RLQR	50
Figura 19 – Cenário 3 - Caminho percorrido	51
Figura 20 – Cenário 4 - Trajeto longo com retas e curvas - LQR	52
Figura 21 – Cenário 4 - Trajeto longo com retas e curvas - RLQR	53
Figura 22 – Cenário 4 - Caminho percorrido	54
Figura 23 – Visão da câmera no teste em laboratório	55
Figura 24 – Estados LQR - Laboratório	55
Figura 25 – Perda de referência do ArUco	56
Figura 26 – LQR - Laboratório	57
Figura 27 – LQR - Linha Reta - Laboratório	58
Figura 28 – RLQR - Linha Reta - Laboratório	59

LISTA DE TABELAS

Tabela 1 – Parâmetros de simulação	41
--	----

SUMÁRIO

1	INTRODUÇÃO	17
2	REVISÃO TEÓRICA	19
2.1	Reguladores lineares quadráticos	19
2.1.1	Regulador Linear Quadrático	19
2.1.2	Regulador Linear Quadrático Robusto	20
2.1.3	Estabilidade	22
2.2	Modelagem do veículo	24
2.3	Odometria visual	27
2.3.1	Calibração da câmera	27
2.3.2	Estimação da Pose	28
2.3.3	Marcadores ArUco	30
3	METODOLOGIA	33
3.1	Simulação do cenário proposto	33
3.1.1	Ambiente de simulação	33
3.1.2	Desenvolvimento do algoritmo	33
3.2	Validação do código em laboratório	36
3.2.1	Ambiente de testes	36
3.2.2	Dados para o controle	37
3.2.3	Algoritmo de controle	38
3.2.4	Execução dos algoritmos	39
4	RESULTADOS	41
4.1	Simulações	41
4.2	Testes em laboratório	55
5	CONCLUSÃO	61
	REFERÊNCIAS	63

1 INTRODUÇÃO

O Brasil tem o agronegócio como um grande impulsionador da economia. Em 2023, o Produto Interno Bruto (PIB) do agronegócio representou cerca de 24% do PIB do país¹. Segundo as projeções da OECD (*Organisation for Economic Co-operation Development*) e da FAO (*Food and Agriculture Organization*), cerca de 87% do crescimento da produção agrícola mundial será consequência do aumento de produtividade (OECD and Food and Agriculture Organization of the United Nations, 2021). Nesse contexto, o desenvolvimento de pesquisas com enfoque na automação do setor agrícola é fundamental, principalmente em um país no qual a atividade contribui de forma significativa na economia.

Visando aumentar produtividade, reduzir custos e melhorar a segurança, os veículos autônomos surgem como uma extensão natural da automação no campo. A evolução histórica das pesquisas sobre o assunto é mostrada por Rondelli, Franceschetti e Mengoli (2022). É possível perceber que os veículos são equipados com sensores, de modo a poderem navegar pelo terreno, e que possuem muitas aplicações, desde a coleta de frutas até pulverização de plantações e roçada do campo.

Diante desse cenário de constante evolução na agricultura, uma possibilidade de implementação de veículos autônomos é em plantações de cana-de-açúcar. Belardo, Cassia e Silva (2015) apresentam o processo de corte, transbordo e transporte da cana, avaliando aspectos econômicos.

O ponto de fundamental interesse nesse trabalho de conclusão de curso é a operação de transbordo, na qual a colheitadeira e o caminhão de transbordo devem se locomover paralelos, a uma distância estabelecida, de modo que a operação ocorra com sucesso. Nessa operação, enquanto a colheitadeira realiza o corte da cana-de-açúcar, ela despeja o que foi colhido no caminhão que anda lateralmente a ela. Desse modo, um caminhão autônomo que fosse capaz de realizar essa tarefa traria benefícios de segurança e redução de custos para o agricultor. Um exemplo dessa aplicação foi realizada pela Scania, como mostra a Figura 1.

Para o controle do veículo, optou-se pela aplicação de reguladores lineares quadráticos. Comparou-se o regulador nominal e o regulador robusto, que modela as incertezas do sistema. Com isso, busca-se reduzir as incertezas associadas à estimativa de estados provenientes de interferências do ambiente, visando um melhor desempenho e maior robustez.

A estimativa de estados é feita por meio de odometria visual, que é uma técnica robusta utilizada para o conhecimento da posição do veículo ao longo do tempo. Ela permite que um veículo se localize de forma robusta utilizando apenas uma sequência de imagens capturadas por uma câmera fixada no próprio veículo (AQEL et al., 2016).

¹ <https://www.cepea.esalq.usp.br/br/pib-do-agronegocio-brasileiro.aspx>

Figura 1 – Caminhão autônomo Scania em plantação de cana-de-açúcar



Fonte: (AUTOINDUSTRIA, 2022).

Desse modo, como objetivo geral do trabalho, tem-se o desenvolvimento de soluções para plantações de cana-de-açúcar, especificamente, na operação de transbordo.

Como objetivos específicos, tem-se o desenvolvimento de algoritmos de controle do veículo autônomo utilizando reguladores lineares quadráticos e odometria visual para estimativa de estados, mais especificamente, um controle servo-visual baseado em posição.

Este TCC está dividido em Revisão Bibliográfica, Metodologia, Resultados e Conclusão. Para a obtenção de resultados foi utilizado o simulador Carla (DOSOVITSKIY et al., 2017) e testes no LASI (Laboratório de Sistemas Inteligentes) utilizando um par de robôs que simulam a colheitadeira e o caminhão.

2 REVISÃO TEÓRICA

2.1 Reguladores lineares quadráticos

Para realizar o controle do veículo, utilizou-se o Regulador Linear Quadrático (LQR) e o Regulador Linear Quadrático Robusto (RLQR), cujas formulações são apresentadas abaixo.

2.1.1 Regulador Linear Quadrático

Considere o modelo em espaço de estados do seguinte sistema linear discreto variante no tempo

$$x_{k+1} = F_k x_k + G_k u_k, \quad (2.1)$$

para $k = 0, \dots, N$, no qual $x_k \in \mathbb{R}^n$ é o vetor de estado, $u_k \in \mathbb{R}^m$ é o vetor de entradas, $F_k \in \mathbb{R}^{n \times n}$ e $G_k \in \mathbb{R}^{n \times m}$ são matrizes conhecidas e x_0 é o vetor de estado inicial.

Seja \mathcal{U} o conjunto das possíveis entradas do sistema. O objetivo do regulador é encontrar a sequência de entradas $\mathcal{U}^* = \{u_0^*, \dots, u_N^*\} \in \mathcal{U}$ que minimize a função de custo

$$J(x, u) = x_N^T P_N x_N + \sum_0^{N-1} (x_k^T Q_k x_k + u_k^T R_k u_k), \quad (2.2)$$

na qual $P_N, Q_k \in \mathbb{R}^{n \times n}$ são matrizes semidefinidas positivas simétricas, $R_k \in \mathbb{R}^{m \times m}$ é uma matriz definida positiva simétrica, e N é chamado de *horizonte de tempo*.

Problema: Dado um estado inicial $x_0 \in \mathbb{R}^n$, determinar a sequência ótima de controle $\mathcal{U}^* = \{u_0^*, \dots, u_N^*\}$ e a sequência de estados $\{x_0^*, \dots, x_N^*\}$ correspondente, que seja solução do problema de otimização

$$\min_{u_k \in \mathcal{U}} J(x, u) \quad (2.3)$$

$$\text{sujeito a } x_{k+1} = F_k x_k + G_k u_k. \quad (2.4)$$

Por meio do método de programação dinâmica (DREYFUS, 2002), obtem-se a solução do problema. Defina as matrizes K_k, L_k e P_k por

$$K_k = -(R_k + G_k^T P_{k+1} G_k)^{-1} G_k^T P_{k+1} F_k, \quad (2.5)$$

$$L_k = F_k - G_k (R_k + G_k^T P_{k+1} G_k)^{-1} G_k^T P_{k+1} F_k, \quad (2.6)$$

$$P_k = F_k^T [P_{k+1} - P_{k+1} G_k (R_k + G_k^T P_{k+1} G_k)^{-1} G_k^T P_{k+1}] F_k + Q_k. \quad (2.7)$$

A solução é dada por

$$u_k^* = K_k x_k \quad (2.8)$$

$$x_{k+1}^* = L_k x_k. \quad (2.9)$$

O Algoritmo 1 mostra o cálculo dos estados e entradas ótimas.

Algoritmo 1 Controlador LQR

Inicialização de parâmetros: Inicializar N, x_0 e $P_N \succcurlyeq 0$.

Calcular para $k = N - 1, \dots, 0$:

$$\begin{aligned} K_k &= -(R_k + G_k^T P_{k+1} G_k)^{-1} G_k^T P_{k+1} F_k, \\ L_k &= F_k - G_k (R_k + G_k^T P_{k+1} G_k)^{-1} G_k^T P_{k+1} F_k, \\ P_k &= F_k^T [P_{k+1} - P_{k+1} G_k (R_k + G_k^T P_{k+1} G_k)^{-1} G_k^T P_{k+1}] F_k + Q_k. \end{aligned}$$

Obter para $k = 0, \dots, N - 1$:

$$\begin{aligned} u_k^* &= K_k x_k \\ x_{k+1}^* &= L_k x_k. \end{aligned}$$

2.1.2 Regulador Linear Quadrático Robusto

O Regulador Linear Quadrático Robusto é similar ao Regulador Linar Quadrático, mas apresenta a modelagem de incertezas do sistema em sua formulação.

Considere o modelo de espaço de estados variante no tempo

$$x_{k+1} = (F_k + \delta F_k)x_k + (G_k + \delta G_k)u_k, \quad (2.10)$$

no qual k, x_k, F_k, G_k e u_k são definidos da mesma forma que na Subseção 2.1.1, e δF_k e δG_k são incertezas no sistema, que são incorporadas nas matrizes F_k e G_k .

Considere o problema de mínimos quadrados regularizados para o sistema

$$x_{k+1} = (\mathcal{A} + \delta \mathcal{A})x_k + (b + \delta b), \quad (2.11)$$

onde a matriz \mathcal{A} e o vetor b estão sujeitas a incertezas limitadas em norma $\delta \mathcal{A}$ e δb , respectivamente. Assim, o problema de otimização é definido da seguinte forma

$$\min_x \max_{\delta \mathcal{A}, \delta b} J(x) \quad (2.12)$$

com a função de custo $J(x)$ dada por

$$\begin{aligned} J(x) &= \|x\|_{\mathcal{Q}}^2 + \|(\mathcal{A} + \delta\mathcal{A})x - (b + \delta b)\|_{\mathcal{W}}^2 = \\ &= x^T \mathcal{Q} x + [(\mathcal{A} + \delta\mathcal{A})x - (b + \delta b)]^T \mathcal{W}[(\mathcal{A} + \delta\mathcal{A})x - (b + \delta b)], \end{aligned} \quad (2.13)$$

sendo $x \in \mathbb{R}^n$ o vetor desconhecido, $\mathcal{Q} \in \mathbb{R}^{n \times n}$ e $\mathcal{W} \in \mathbb{R}^{m \times m}$ matrizes de ponderação simétricas conhecidas, A e b conhecidos e sujeitos a incertezas paramétricas $\delta A \in \mathbb{R}^{m \times n}$ e $\delta b \in \mathbb{R}^m$ modeladas por

$$\begin{bmatrix} \delta\mathcal{A} & \delta b \end{bmatrix} = \mathcal{H}\Delta \begin{bmatrix} E_{\mathcal{A}} & E_b \end{bmatrix}, \quad (2.14)$$

onde \mathcal{H} é uma matriz não-nula, $E_{\mathcal{A}}$ e E_b são matrizes conhecidas de dimensões apropriadas, Δ é uma matriz de contração arbitrária com $\|\Delta\|^2 \leq 1$. A solução ótima do problema (2.12) é mostrada em Sayed e Nascimento (1999).

Conforme apresentado por Terra, Cerri e Ishihara (2014), fazendo

$$\begin{aligned} x &\leftarrow \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix}, Q \leftarrow \begin{bmatrix} P_{k+1} & 0 \\ 0 & R_k \end{bmatrix}, W \leftarrow \begin{bmatrix} Q_k & 0 \\ 0 & \mu I_n \end{bmatrix}, \\ A &\leftarrow \begin{bmatrix} 0 & 0 \\ I_n & -G_k \end{bmatrix}, \delta A \leftarrow \begin{bmatrix} 0 & 0 \\ 0 & -\delta G_k \end{bmatrix}, b \leftarrow \begin{bmatrix} -I_n \\ F_k \end{bmatrix} x_k, \delta b \leftarrow \begin{bmatrix} 0 \\ \delta F_k \end{bmatrix} x_k, \\ H &\leftarrow \begin{bmatrix} 0 \\ H_k \end{bmatrix}, \Delta \leftarrow \Delta_k, E_{\mathcal{A}} \leftarrow \begin{bmatrix} 0 & -E_{G_k} \end{bmatrix}, E_b \leftarrow E_{F_k}, \end{aligned}$$

é obtida a solução para o sistema (2.10).

As matrizes K_k , L_k e P_k são calculadas:

$$\begin{bmatrix} L_k \\ K_k \\ P_k \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -I_n \\ 0 & 0 & \hat{F}_k \\ I_n & 0 & 0 \\ 0 & I_m & 0 \end{bmatrix}^T \begin{bmatrix} P_{k+1}^{-1} & 0 & 0 & 0 & I_n & 0 \\ 0 & R_k^{-1} & 0 & 0 & 0 & I_m \\ 0 & 0 & Q_k^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma_{\mu,k} & \hat{I}_n & -\hat{G}_k \\ I_n & 0 & 0 & \hat{I}_n^T & 0 & 0 \\ 0 & I_m & 0 & -\hat{G}_k^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ -I_n \\ \hat{F}_k \\ 0 \\ 0 \end{bmatrix}, \quad (2.15)$$

onde I_n e I_m são matrizes identidade de dimensões n e m , respectivamente, e

$$\hat{F}_k = \begin{bmatrix} F_k \\ E_{F_k} \end{bmatrix}, \hat{G}_k = \begin{bmatrix} G_k \\ E_{G_k} \end{bmatrix}, \hat{I}_n = \begin{bmatrix} I_n \\ 0 \end{bmatrix}, \Sigma_{\mu,k} = \begin{bmatrix} \Phi(\mu, \lambda) & 0 \\ 0 & \hat{\lambda}_{\mu}^{-1} I_l \end{bmatrix},$$

$$\Phi(\mu, \hat{\lambda}_{\mu}) = \mu^{-1} I - \hat{\lambda}_{\mu}^{-1} H_k H_k^T, \text{ e}$$

$$\hat{\lambda}_{\mu} = (1 + \alpha) \|\mu H_k^T H_k\|, \text{ para algum } \alpha > 0.$$

Assim,

$$u_k^* = K_k x_k \quad (2.16)$$

$$x_{k+1}^* = L_k x_k. \quad (2.17)$$

O Algoritmo 2 mostra o cálculo dos estados e entradas ótimas.

Algoritmo 2 Controlador RLQR

Inicialização de parâmetros: Inicializar N, x_0 e $P_N \succ 0$.

Calcular para $k = N - 1, \dots, 0$:

$$\begin{bmatrix} L_k \\ K_k \\ P_k \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -I_n \\ 0 & 0 & \hat{F}_k \\ I_n & 0 & 0 \\ 0 & I_m & 0 \end{bmatrix}^T \begin{bmatrix} P_{k+1}^{-1} & 0 & 0 & 0 & I_n & 0 \\ 0 & R_k^{-1} & 0 & 0 & 0 & I_m \\ 0 & 0 & Q_k^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma_{\mu,k} & \hat{I}_n & -\hat{G}_k \\ I_n & 0 & 0 & \hat{I}_n^T & 0 & 0 \\ 0 & I_m & 0 & -\hat{G}_k^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ -I_n \\ \hat{F}_k \\ 0 \\ 0 \end{bmatrix}, \quad (2.18)$$

Obter para $k = 0, \dots, N - 1$:

$$\begin{aligned} u_k^* &= K_k x_k \\ x_{k+1}^* &= L_k x_k. \end{aligned}$$

Fazendo $\mu \rightarrow \infty$, tem-se $\hat{\lambda}_\mu \rightarrow \infty \Rightarrow \hat{\lambda}_\mu^{-1} \rightarrow 0$ e $\Phi(\mu, \hat{\lambda}_\mu) \rightarrow 0$. Desse modo, $\Sigma_{\mu,k}$ se torna uma matriz nula quando $\mu \rightarrow \infty$.

2.1.3 Estabilidade

Será mostrada a estabilidade do RLQR, visto que a estabilidade do LQR segue procedimento similar. As expressões para o cálculo das matrizes são obtidas por meio da equação 2.15. A prova é baseada na dissertação de Cerri (2009), onde utiliza-se $\mu \rightarrow \inf$. Além disso, assuma que o par de matrizes F_k e G_k é controlável para todo k .

De acordo com a Seção 6.3 (CERRI, 2009), tem-se as seguintes relações:

$$P_{k+1} = \hat{F}_k^T (P_k - P_k \hat{G}_k (I + \hat{G}_k^T P_k \hat{G}_k)^{-1} \hat{G}_k^T P_k) \hat{F}_k + \hat{Q}, \quad (2.19)$$

sendo

$$\hat{F}_k := F_k - G R^{-1} E_G^T (E_G R^{-1} E_G^T)^{-1} E_F, \quad (2.20)$$

$$\hat{R} := R^{-1} + R^{-1} E_G^T (E_G R^{-1} E_G^T)^{-1} E_G R^{-1}, \quad (2.21)$$

$$\hat{G}_k := G_k \hat{R}^{\frac{1}{2}}, \quad (2.22)$$

$$\hat{Q} := Q + E_F^T (E_G R^{-1} E_G^T)^{-1} E_F, \quad (2.23)$$

e assume que a condição inicial P_0 satisfaz $0 \prec P_0 \prec Q$.

De acordo com o Lema 6.2.3 (CERRI, 2009), tem-se as seguintes relações:

$$\hat{K}_k = -(I + \hat{G}_k^T P_k \hat{G}_k)^{-1} \hat{G}_k^T P_k \hat{F}_k, \quad (2.24)$$

$$\hat{L}_k = \hat{F}_k + \hat{G}_k \hat{K}_k. \quad (2.25)$$

A demonstração da estabilidade é uma adaptação do Lema 6.3.5 (CERRI, 2009), considerando a variação temporal de F e G .

Lema 2.1.1 *Se $P_k \succ 0$, então todos os autovalores da matriz*

$$L_k = (F_k + G_k K_k) = F_k - G_k (I + G_k^T P_k G_k)^{-1} G_k^T P_k F_k, \quad (2.26)$$

pertencem ao círculo unitário aberto, ou seja, L_k é estável.

Demonstração: A demonstração é feita por redução ao absurdo. Assuma que \hat{L}_k não é estável. Desse modo, existe algum autovalor λ , tal que $|\lambda| \geq 1$, com $L_k x = \lambda x$ para $x \neq 0$.

Sabe-se que

$$P_{k+1} = \hat{F}_k^T (P_k - P_k \hat{G}_k (I + \hat{G}_k^T P_k \hat{G}_k)^{-1} \hat{G}_k^T P_k) \hat{F}_k + \hat{Q}, \quad (2.27)$$

que pode ser escrita como

$$P_{k+1} = \hat{L}_k^T P_k \hat{L}_k + \hat{K}_k^T I \hat{K}_k + \hat{Q}, \quad (2.28)$$

sendo

$$\hat{L}_k = (\hat{F}_k + \hat{G}_k \hat{K}_k) = \hat{F}_k - \hat{G}_k (I + \hat{G}_k^T P_k \hat{G}_k) \hat{G}_k^T P_k \hat{F}_k. \quad (2.29)$$

Pré-multiplicando e pós-multiplicando a Equação 2.28 por x^T e x , respectivamente, tem-se:

$$\begin{aligned} x^T P_k x &= x^T \hat{L}_k^T P_k \hat{L}_k x + x^T \hat{K}_k^T I \hat{K}_k x + x^T \hat{Q} x \Rightarrow \\ x^T P_k x &= |\lambda|^2 x^T P_k x + x^T \hat{K}_k^T I \hat{K}_k x + x^T \hat{Q} x \Rightarrow \\ (1 - |\lambda|^2) x^T P_k x &= x^T (\hat{K}_k^T I \hat{K}_k + \hat{Q}) x. \end{aligned}$$

Como $P_k \succ 0$ e $x \neq 0$, tem-se $x^T P_k x > 0$. Além disso, $|\lambda| \geq 1 \Rightarrow |\lambda|^2 \geq 1$. Logo, $(1 - |\lambda|^2) x^T P_k x \leq 0$. Entretanto, como $\hat{K}_k^T I \hat{K}_k + \hat{Q} \succ 0$ e $x \neq 0$, $x^T (\hat{K}_k^T I \hat{K}_k + \hat{Q}) x > 0$, chegando em um absurdo.

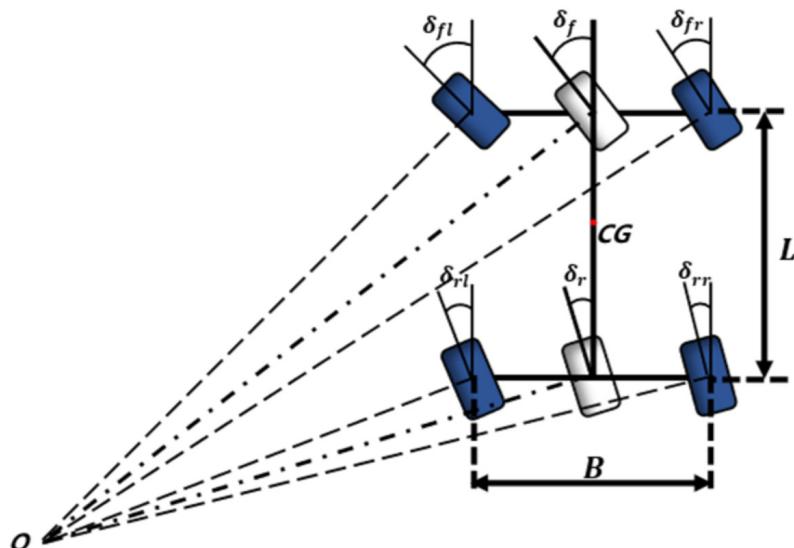
Portanto, \hat{L} não possui autovalores de módulo maior ou igual a 1, ou seja, é estável.

Para um desenvolvimento similar do Corolário 6.3.1 (CERRI, 2009), conclui-se que o regulador é estável.

2.2 Modelagem do veículo

Para simplificar a modelagem e testes, o veículo foi modelado utilizando o modelo de bicicleta, que é uma simplificação do modelo Ackermann (ZHAO et al., 2013), que une as rodas pertencentes ao mesmo eixo em uma única roda, no centro do eixo. Essa simplificação pode ser vista na Figura 2.

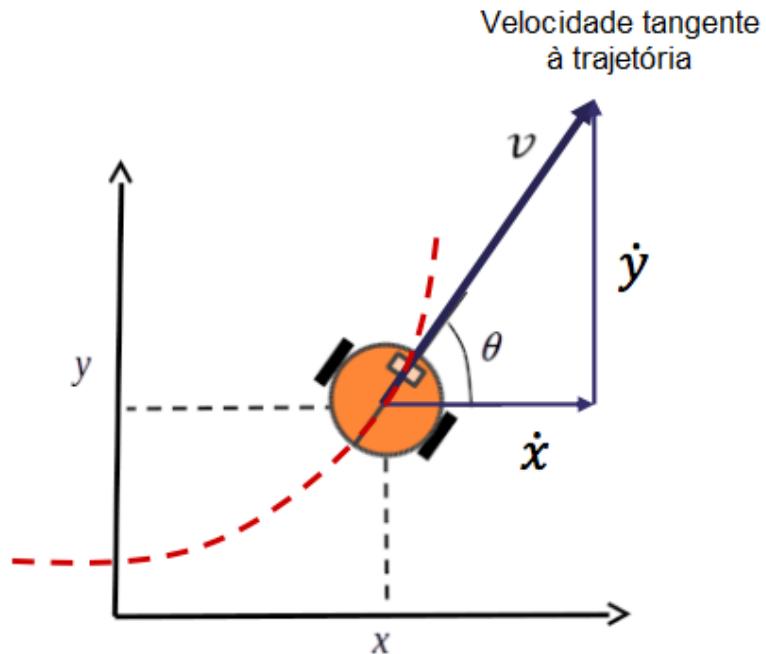
Figura 2 – Modelo de bicicleta dos veículos



Fonte: Lu et al. (2022).

Além disso, pode-se utilizar dois tipos de modelagem: cinemática ou dinâmica. A modelagem cinemática considera apenas grandezas como posição, orientação e velocidade. Já a modelagem dinâmica considera a massa do veículo e as forças atreladas a ele, como a força de atrito. A modelagem adotada foi a cinemática, cujo diagrama é mostrado na Figura 3.

Figura 3 – Modelo cinemático



Fonte: Adaptado de Ding (2021).

Pela geometria do problema, considerando o modelo de bicicleta e a modelagem cinemática, tem-se

$$\begin{cases} \dot{x} = v \cos(\theta), \\ \dot{y} = v \sin(\theta), \\ \dot{\theta} = \frac{v \tan(\delta)}{L}, \\ \dot{v} = a. \end{cases} \quad (2.30)$$

Define-se o vetor de estados com a posição em x , posição em y , orientação θ e velocidade v . Além disso, define-se o vetor de entradas com o ângulo de esterçamento δ e a aceleração a . De forma concisa,

$$z = \begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix} \text{ e } u = \begin{bmatrix} a \\ \delta \end{bmatrix}, \quad (2.31)$$

onde z é o vetor de estados e u é o vetor de entradas.

Desse modo, linearizando o sistema (2.30) e utilizando as definições em (2.31), tem-se o sistema linear

$$\dot{z} = Az + Bu \Rightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_{ref}\sin(\theta_{ref}) & \cos(\theta_{ref}) \\ 0 & 0 & v_{ref}\cos(\theta_{ref}) & \sin(\theta_{ref}) \\ 0 & 0 & 0 & \frac{\tan(\delta_{ref})}{L} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{v_{ref}}{L\cos^2(\delta_{ref})} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ \delta \end{bmatrix}. \quad (2.32)$$

Seja Δt o passo de discretização do sistema contínuo no tempo (2.32). Desse modo, as matrizes do sistema discreto são

$$F = A\Delta t + I = \begin{bmatrix} 1 & 0 & -v_{ref}\sin(\theta_{ref})\Delta t & \cos(\theta_{ref})\Delta t \\ 0 & 1 & v_{ref}\cos(\theta_{ref})\Delta t & \sin(\theta_{ref})\Delta t \\ 0 & 0 & 1 & \frac{\tan(\delta_{ref})}{L}\Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.33)$$

$$G = B\Delta t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{v_{ref}}{L\cos^2(\delta_{ref})}\Delta t \\ \Delta t & 0 \end{bmatrix}, \quad (2.34)$$

e o sistema discreto é

$$z_{k+1} = Fz_k + Gu_k \quad (2.35)$$

Sejam z_{ref} e u_{ref} o vetor de estado de referência e o vetor de entradas de referência, respectivamente. Além disso, defina $\tilde{z} = z - z_{ref}$ e $\tilde{u} = u - u_{ref}$.

Defina o sistema em espaço de estados conforme (2.36).

$$\tilde{z}_{k+1} = F\tilde{z}_k + G\tilde{u}_k. \quad (2.36)$$

Esse sistema tem como variáveis de estado os erros das variáveis cinemáticas. Desse modo, o ponto de operação desejado é a origem.

Por meio do Algoritmo 1, obtém-se a matriz de ganho K e, consequentemente, as saídas de controle para o veículo:

$$\tilde{u} = K\tilde{z} = K(z - z_{ref}) \quad (2.37)$$

$$u = \tilde{u} + u_{ref} = K(z - z_{ref}) + u_{ref}. \quad (2.38)$$

Para a aplicação do controlador, é necessário determinar as variáveis de estado do veículo a ser seguido. Essas variáveis podem ser estimadas com o uso de uma câmera, por meio de odometria visual.

2.3 Odometria visual

Veículos e robôs terrestres dependem da odometria para estimar a trajetória percorrida enquanto se locomovem, a qual é uma parte fundamental para navegação autônoma. Odometria visual é uma odometria feita por meio da análise de dados visuais, como imagens de câmeras. A odometria convencional estima a posição de um robô por meio da medição da rotação das rodas, utilizando sensores. Um dos problemas decorrentes disso é causado pelo escorregamento da roda, que torna as estimativas cada vez mais imprecisas pela perda de tração das rodas. A vantagem do método de odometria visual é que esse problema é evitado (LIM; BRAUNL, 2020).

2.3.1 Calibração da câmera

O primeiro passo para aplicações de odometria visual é a calibração da câmera, que é dividida em duas fases.

Primeiramente, a modelagem da câmera lida com a aproximação matemática do comportamento físico e ótico do sensor utilizando um conjunto de parâmetros. A segunda fase consiste com o uso de métodos diretos ou iterativos para estimar os valores desses parâmetros (SALVI; ARMANGUÉ; BATLLE, 2002).

Existem dois tipos de parâmetros a serem considerados no modelo. Um desses tipos é o conjunto de parâmetros intrínsecos, que modelam a geometria interna e as características óticas da câmera. Em linhas gerais, os parâmetros intrínsecos modelam o modo que a luz é projetada através da lente no plano da imagem do sensor. Já o outro tipo é o conjunto de parâmetros extrínsecos, que indicam a posição e a orientação da câmera em relação a um sistema de coordenadas do mundo, que fornece informações em relação à um sistema de coordenadas fixo, em oposição ao sistema de coordenadas da câmera (SALVI; ARMANGUÉ; BATLLE, 2002).

Os parâmetros (f_x, f_y, c_x, c_y) modelam a matriz da câmera C . Os parâmetros (k_1, k_2, k_3, p_1, p_2) modelam as distorções radial e tangencial causadas pela lente (HIERONYMUS, 2012).

- Distorção radial:

$$\begin{aligned}x_{distorcido} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\y_{distorcido} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6).\end{aligned}$$

- Distorção tangencial:

$$\begin{aligned}x_{distorcido} &= x + [2p_1xy + p_2(r^2 + 2x^2)] \\y_{distorcido} &= y + [p_1(r^2 + 2y^2) + 2p_2xy].\end{aligned}$$

- Coeficientes da matriz da câmera:

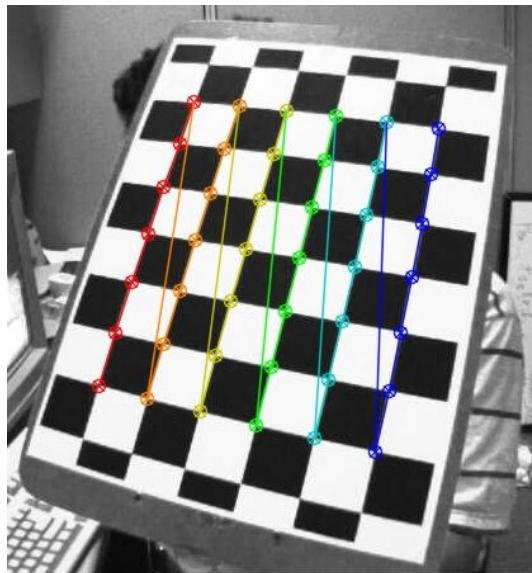
$$C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

f_x e f_y : Distâncias focais em x e y

c_x e c_y : Coordenadas x e y do centro óptico.

Para a calibração, é utilizado uma imagem quadriculada, alternando cores pretas e brancas, como visto na Figura 4. O algoritmo encontra os cantos correspondentes e calcula todos os parâmetros necessários ¹.

Figura 4 – Imagem para calibração de câmera



Fonte: Adaptado do tutorial do OpenCV¹.

2.3.2 Estimação da Pose

A estimação da pose consiste em determinar a posição e orientação de determinado objeto, ou seja, sua configuração espacial ². Uma das representações para a pose é um vetor de translação t e uma matriz de rotação R , dados por

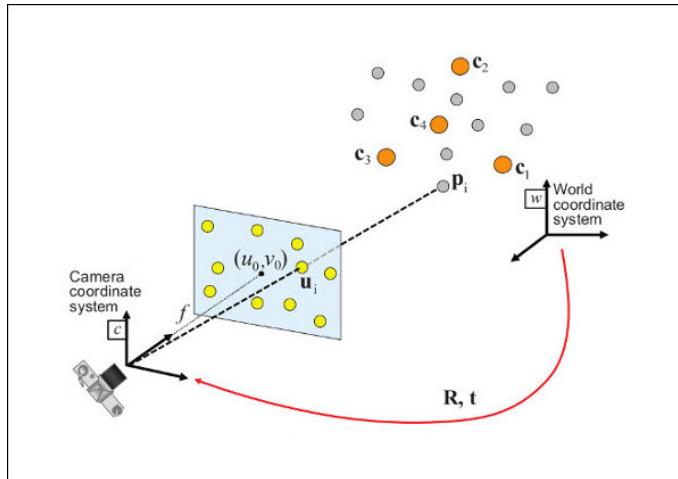
$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \text{ e } t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix},$$

que levam pontos das coordenadas globais para as coordenadas da câmera (u, v) , conforme a Figura 5.

¹ https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

² https://docs.opencv.org/3.4/d7/d53/tutorial_py_pose.html

Figura 5 – Mudança de coordenadas

Fonte: Adaptado do tutorial do OpenCV³.

Seja C a matriz da câmera, Π a matriz de projeção dos pontos em três dimensões para o plano óptico da câmera e T_w a matriz de transformação de coordenadas globais para as da câmera. Desse modo, tem-se:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = C\Pi T_w \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.39)$$

$$\begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix} = T_w \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.40)$$

Para obter essa representação, é utilizado o algoritmo PnP (*Perspective-n-Point*) (LI; XU; XIE, 2012), que encontra as matrizes R e t por meio da correspondência entre os pontos do sistema de coordenadas 3D do mundo com os pontos do sistema de coordenadas 2D da imagem. Para uma solução única, é necessário informar pelo menos 4 pontos para o algoritmo.³

³ https://docs.opencv.org/3.4/d5/d1f/calib3d_solvePnP.html

De modo a obter 4 pontos que possam ser utilizados no algoritmo, pode-se utilizar algum tipo de marcador, que ofereça uma referência ou medida na imagem.

2.3.3 Marcadores ArUco

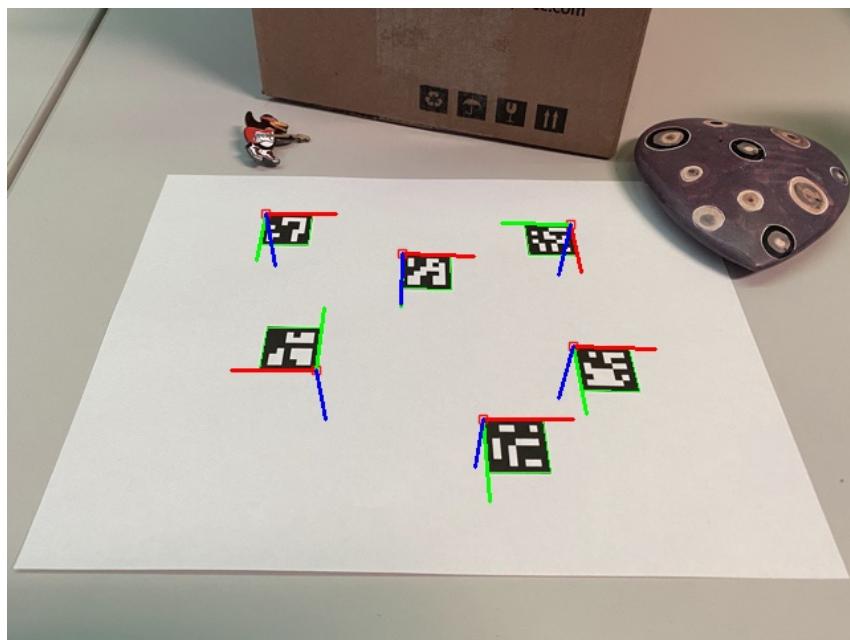
Marcadores ArUco são um tipo de marcador fiducial. Marcadores fiduciais, em sua forma geral, são objetos utilizados para fornecer um ponto de referência ou medida em uma imagem (KALAITZAKIS et al., 2021). Existem inúmeros tipos de marcadores fiduciais, cada qual com a sua especificidade.

Uma das aplicações primárias de marcadores fiduciais na robótica é localização e mapeamento. Mesmo que outros métodos como VIO e SLAM possam fornecer dados precisos, eles requerem condições ideais de luminosidade e características visuais distintas. Marcadores fiduciais são utilizados para aprimorar a qualidade da estimativa de posição em ambientes nos quais essas condições não são atendidas (KALAITZAKIS et al., 2021).

De acordo com os resultados de Kalaitzakis et al. (2021), pode-se perceber que o custo computacional dos marcadores ArUco e ARTag são os menores entre os marcadores analisados. Desse modo, essa é uma das motivações para a escolha deste marcador em específico.

Por meio da detecção dos 4 cantos do marcador ArUco e seu tamanho, é possível aplicar o algoritmo PnP (Subseção 2.3.2) e determinar a orientação, posição e profundidade (distância até a câmera) do marcador⁴, como exemplificado na Figura 6.

Figura 6 – Estimativa da pose de ArUcos



Fonte: Tutorial do OpenCV⁴.

⁴ https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

Portanto, unindo os conceitos dos reguladores e alimentando-os com os estados estimados por meio da odometria visual, com o auxílio do marcador ArUco, é possível realizar o controle do veículo.

3 METODOLOGIA

A metodologia consiste na análise de cenários de simulação e laboratório. Em ambos os casos são utilizados o LQR e RLQR para comparação. Na simulação, os estados são obtidos do simulador. Já nos testes em laboratório, os estados são estimados por odometria visual, utilizando o marcador ArUco como referência.

3.1 Simulação do cenário proposto

3.1.1 Ambiente de simulação

Para simular o cenário da operação de transbordo em plantações de cana-de-açúcar, foi utilizado o *software* Carla (DOSOVITSKIY et al., 2017). Para a simulação, foram utilizados dois veículos Tesla Model 3¹, onde um deles já tem um caminho pré-definido (chamado de colheitadeira), e o outro tem os dados de posição, velocidade, orientação e ângulo de esterçamento de ambos os veículos, que serão usados para o controle deste veículo (chamado de caminhão).

Para o desenvolvimento e avaliação do controlador foram utilizados 4 cenários com diferentes trajetórias: uma estrada em linha reta, uma rotatória, e duas pistas com partes retas e curvas, uma menor e outra maior. A Figura 7 mostra as trajetórias de cada cenário.

3.1.2 Desenvolvimento do algoritmo

Para o controlador, foram utilizados o LQR e o RLQR, com as matrizes do sistema variantes no tempo, visto que a cada iteração do controle o alvo muda de posição e, com isso, deve-se alterar as matrizes F e G . Desse modo, baseado na posição atual da colheitadeira, o algoritmo calcula a nova posição alvo, que é passada para o controlador LQR ou RLQR.

Baseado no modelo apresentado na Seção 2.2, os dados necessários para a atualização das matrizes F e G são:

- θ_{ref} : Orientação de referência.
- v_{ref} : Velocidade de referência.
- δ_{ref} : Ângulo de esterçamento de referência.

Seja ξ como uma grandeza genérica, como orientação ou velocidade. Defina ξ_{cam} e ξ_{col} como a grandeza ξ do caminhão e da colheitadeira, respectivamente.

Pela geometria do problema, as grandezas de referência podem ser calculadas. As orientações e os ângulos de esterçamento de ambos os veículos devem ser os mesmos.

¹ <https://www.tesla.com/model3>

Figura 7 – Percursos de simulação



(a) Trajetória - Primeiro cenário



(b) Trajetória - Segundo cenário



(c) Trajetória - Terceiro cenário



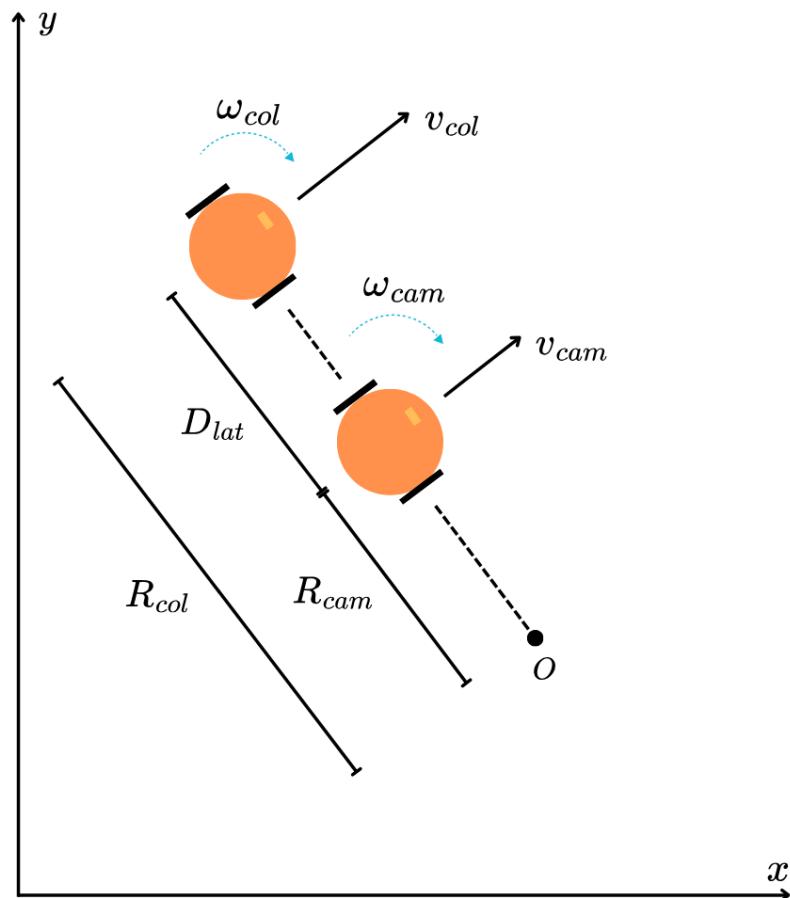
(d) Trajetória - Quarto cenário

Elaborado pelo autor.

Nesse caso, $\theta_{ref} = \theta_{col}$ e $\delta_{ref} = \delta_{col}$. Para a velocidade, tem-se a seguinte análise, baseada na Figura 8, na qual ω é a velocidade angular, v é a velocidade linear, R é o raio da curva, O é o centro da trajetória, e D_{lat} é a distância lateral.

- Linha reta: A velocidade do caminhão deve ser igual a da colheitadeira. Nesse caso, $v_{ref} = v_{col}$.
- Curva:

Figura 8 – Cenário de movimento circular



Fonte: Elaborado pelo autor

Para o desenvolvimento algébrico, adotou-se um ângulo de esterçamento positivo quando os veículos estão se movendo no sentido horário. Para consistência das equações, um raio de trajetória negativo indica uma trajetória curva com raio de valor absoluto igual ao valor apresentado e sentido anti-horário.

Como o é movimento curvilíneo, tem-se $\omega_{cam} = \omega_{col}$:

$$\omega_{col} = \omega_{cam} \Rightarrow \frac{v_{col}}{R_{col}} = \frac{v_{cam}}{R_{cam}} \Rightarrow v_{cam} = v_{col} \frac{R_{cam}}{R_{col}}. \quad (3.1)$$

Desse modo,

$$v_{ref} = v_{col} \frac{R_{cam}}{R_{col}}. \quad (3.2)$$

Para calcular R_{cam} e R_{col} , utiliza-se o ângulo de esterçamento δ_{col} :

$$R_{col} = \frac{L_{col}}{\tan(\delta_{col})} \text{ e } R_{cam} = R_{col} - D_{lat} \Rightarrow R_{cam} = \frac{L_{col}}{\tan(\delta_{col})} - D_{lat}. \quad (3.3)$$

Portanto, a velocidade de referência é dada por

$$v_{ref} = v_{col} \frac{R_{cam}}{R_{col}} = v_{col} \frac{R_{col} - D_{lat}}{R_{col}} = v_{col} \left(1 - D_{lat} \frac{\tan(\delta_{col})}{L_{col}} \right). \quad (3.4)$$

3.2 Validação do código em laboratório

3.2.1 Ambiente de testes

Para os testes em laboratório, foram utilizados dois robôs LIMO² da AgileX³ idênticos, de modo a imitar o cenário simulado no software Carla, descrito na Seção 3.1. A Figura 9 mostra uma foto dos robôs no laboratório.

Figura 9 – Robôs utilizados em laboratório



Fonte: Elaborado pelo autor.

O robô conta com vários sensores, mas o único sensor utilizado para os testes é uma câmera, instalada na lateral do robô que fará o papel do caminhão.

² <https://global.agilex.ai/solutions/4>

³ <https://global.agilex.ai/>

A câmera utilizada é OAK-D S2.⁴

O robô que representa a colheitadeira teve um marcador ArUco colocado na sua lateral, de modo que a câmera do outro robô pudesse localizá-lo.

3.2.2 Dados para o controle

Considere a mesma nomenclatura de caminhão e colheitadeira da Subseção 3.1.1.

Os dados imediatamente disponíveis são os dados de velocidade e ângulo de esterçamento do caminhão, obtidos por meio dos sensores. Para o funcionamento correto do controle, é necessário obter as medidas das variáveis contidas no sistema linearizado. Esses dados são velocidade de referência, orientação de referência, ângulo de esterçamento de referência, tamanho do veículo e tempo de discretização, que compõem as matrizes F e G, da Equação (2.35).

O tempo de discretização e o tamanho do veículo são fixos e conhecidos. As outras grandezas de referência mudam a cada instante e devem ser determinadas.

Os dados obtidos por meio da pose são a orientação e a posição relativas à câmera. Assim, os dados utilizados para o controle consideram o centro da câmera como origem do sistema. A Figura 10 mostra a referência dos eixos para o sistema de coordenadas utilizado.⁵

Perceba que o eixo z representa a profundidade e o eixo x representa a direção do movimento do veículo, já que no experimento a câmera encontra-se com o eixo óptico paralelo ao solo.

Como as medidas são relativas à câmera, tem-se $\theta_{ref} = \theta_{rel}$.

Utilizando duas medidas consecutivas de posição relativa (vetor t obtido do algoritmo PnP), é possível obter a velocidade relativa fazendo

$$\mathbf{v}_{rel_k} = \frac{\mathbf{t}_k - \mathbf{t}_{k-1}}{\Delta t}. \quad (3.5)$$

Desse modo, é possível determinar a velocidade da colheitadeira:

$$\mathbf{v}_{cam_k} = [v_{cam_k}, 0, 0] \Rightarrow \mathbf{v}_{col_k} = \mathbf{v}_{cam_k} + \mathbf{v}_{rel_k} \Rightarrow v_{col_k} = sign(\mathbf{v}_{col_k}^x) \|\mathbf{v}_{col_k}\|, \quad (3.6)$$

onde $\mathbf{v}_{col_k}^x$ é a componente x do vetor. O sinal dessa componente indica a direção do movimento.

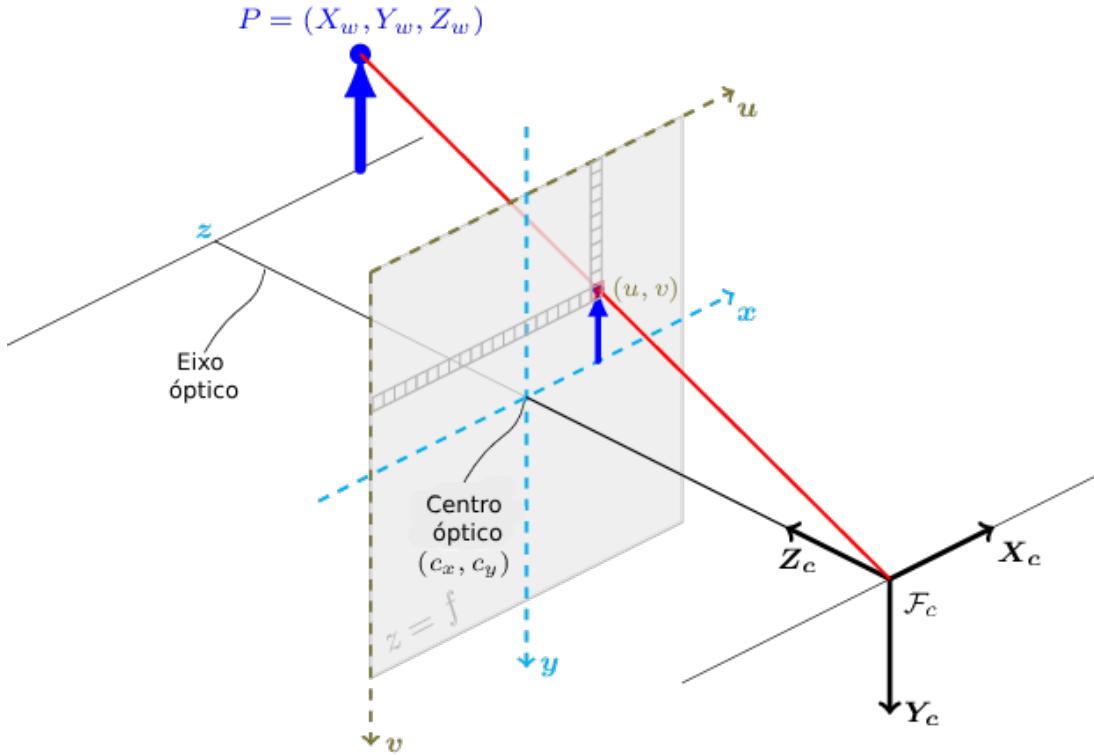
Além disso, é possível determinar o ângulo de esterçamento por meio de duas estimativas consecutivas. Discretizando a terceira equação do sistema (2.30):

$$\frac{\theta_k - \theta_{k-1}}{T_s} = \frac{v_k \tan(\delta_k)}{L} \Rightarrow \delta_k = \tan^{-1} \left(\frac{L}{v_k T_s} (\theta_k - \theta_{k-1}) \right). \quad (3.7)$$

⁴ <https://shop.luxonis.com/products/oak-d-s2>

⁵ https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html

Figura 10 – Eixos do sistema de coordenadas utilizado



Fonte: Adaptado do tutorial do OpenCV⁵.

Por fim, por meio da velocidade do caminhão e a velocidade da colheitadeira calculada, é possível calcular a velocidade de referência do mesmo modo que na Seção 3.1.

É importante destacar que a Equação (3.7) poderia ser ruidosa devido ao cálculo envolver muitas variáveis que também são estimativas. Entretanto, ao comparar o valor calculado com valores obtidos direto do simulador, percebeu-se apenas em alguns instantes a presença de ruído relevante no valor calculado, que não afetou o controle devido a sua rara ocorrência. Em laboratório não foi feita essa avaliação.

3.2.3 Algoritmo de controle

De posse de todos os valores citados na Seção 3.2.2, é possível calcular as matrizes F_k e G_k . A matriz de ganho K é obtida do LQR ou RLQR e, consequentemente, o sinal de controle é determinado.

$$z - z_{ref} = [-\mathbf{t}_k^x, D_{lat} - \mathbf{t}_k^z, -\theta_{ref}, v_{cam} - v_{ref}] \Rightarrow u = K(z - z_{ref}) + u_{ref}, \quad (3.8)$$

onde D_{lat} é a distância lateral entre o caminhão e a colheitadeira, que foi pré-estabelecida.

Por fim, do mesmo modo que na simulação, os sinais de controle são colocados em um intervalo de -1 a 1, representando a porcentagem de aceleração e de esterçamento. Matematicamente:

$$throttle = \max(-1, \min(1, a)); \quad (3.9)$$

$$steering = \max(-1, \min(1, \delta)). \quad (3.10)$$

Para o envio de comandos ao robô, é necessário fornecer diretamente o valor de velocidade e ângulo de esterçamento. Este último é obtido da saída de controle, fazendo:

$$\delta_{cmd} = steering \cdot \delta_{max}, \quad (3.11)$$

onde δ_{cmd} é o comando de ângulo de esterçamento e δ_{max} é o ângulo de esterçamento máximo.

Já a velocidade é obtida por meio de uma lei cinemática:

$$v_{cmd} = v_{ant} + throttle \cdot a_{max} \Delta t, \quad (3.12)$$

onde v_{cmd} é o comando de velocidade, v_{ant} é a velocidade anterior e a_{max} é a aceleração máxima.

Como escolha para o projeto, optou-se por não permitir o movimento de ré do robô. Além disso, o ângulo de esterçamento máximo permitido é de 15° para ambos os lados, aceleração e desaceleração máxima de 1 m/s², e velocidade máxima de 0,2 m/s. Ademais, para evitar mudanças abruptas no ângulo de esterçamento, limitou-se a variação máxima de uma iteração para a outra a 0,05 radianos. Esses restrições foram escolhidas por limitações do robô e dos ambientes de teste.

3.2.4 Execução dos algoritmos

Para obter um controle mais eficiente, o algoritmo de aquisição da pose por meio da câmera é rodado em um nó do *Robot Operating System* (ROS), enquanto o algoritmo de controle é rodado em outro nó. O nó de controle escuta os dados de pose, que são publicados pelo nó da câmera. Desse modo, é possível obter uma frequência maior do controlador, devido ao paralelismo e divisão de tarefas nos códigos.

4 RESULTADOS

4.1 Simulações

Para os cenários de simulação, foram traçados gráficos com os erros de distância longitudinal, distância lateral, velocidade e ângulo de orientação. As Figuras 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 e 22 mostram os dados para o cenários 1, 2, 3 e 4, respectivamente, sendo três Figuras por cenário, uma para cada regulador e uma para o caminho percorrido. Para cada cenário 3.1.1, foram utilizados os parâmetros da Tabela 1 e simulados ambos os reguladores, com e sem ruídos no controle. Os ruídos consistem na aplicação de algum sinal de controle de variação brusca, como ângulo de esterçamento máximo para algum dos lados ou frenagem repentina.

Cenário	v_{col} (km/h)	D_{lat} (m)	R	T_s (s)
1	10	3	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	0,01
2	10	3	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	0,01
3	10	5	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	0,01
4	10	7	$\begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$	0,01
Laboratório	Variável	Variável	$\begin{bmatrix} 1 & 0 \\ 0 & 0,2 \end{bmatrix}$	0,1

Tabela 1 – Parâmetros de simulação

As matrizes Q , E_F e E_G tiveram seus valores idênticos para todos os cenários de simulação e laboratório:

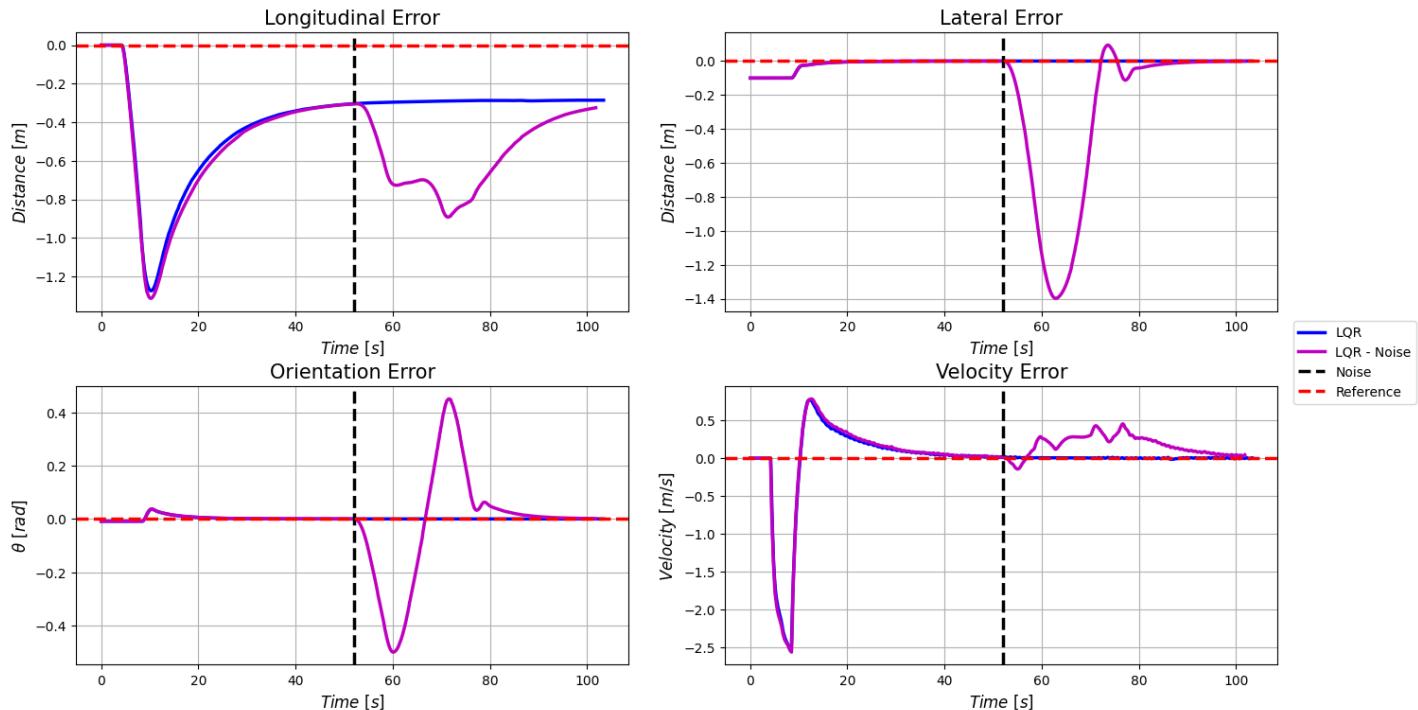
$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, E_F = \begin{bmatrix} 0,01 & 0,01 & 0,02 & 0,001 \end{bmatrix},$$

$$E_G = \begin{bmatrix} 0,007 & 0,001 \end{bmatrix}, H = \begin{bmatrix} 0 \\ 0,001 \\ 0 \\ 0 \end{bmatrix}.$$

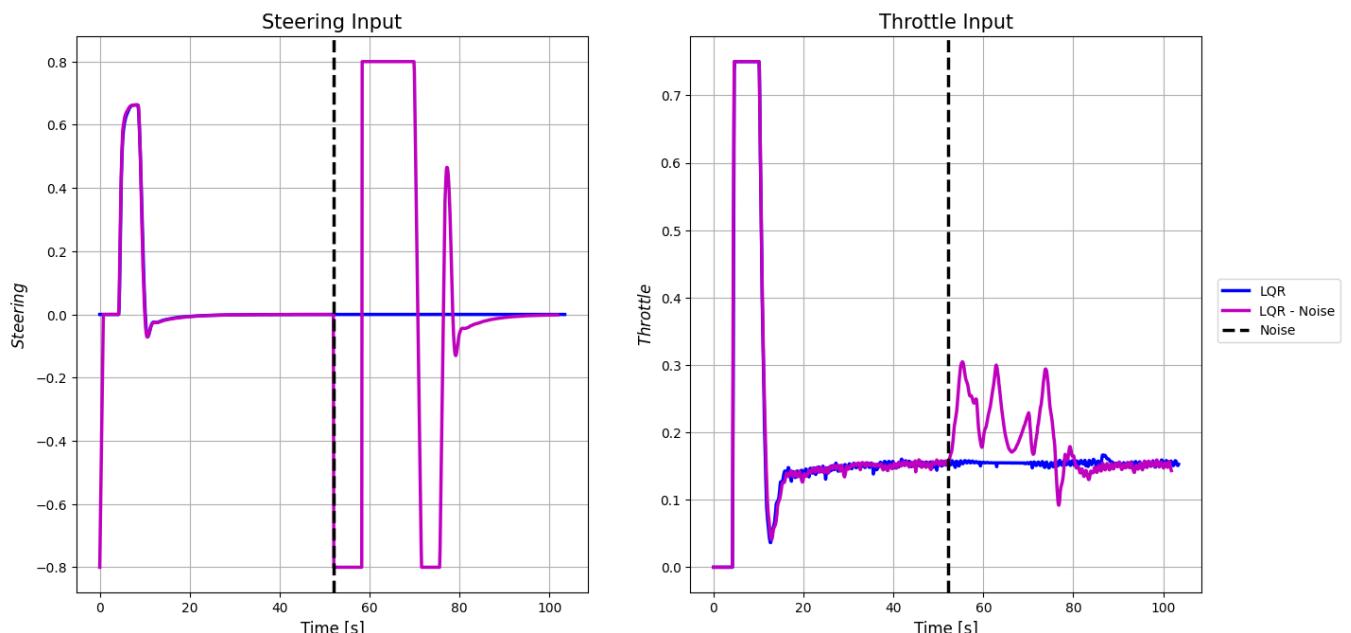
Os valores de velocidade foram escolhidos baseados nos valores reais de velocidades de colheitadeiras no cenário proposto, da operação de transbordo. Já as distâncias laterais

foram definidas de acordo com cada situação e suas restrições físicas. Por fim, as matrizes foram obtidas por tentativa e erro, de modo a otimizar os controladores.

Figura 11 – Cenário 1 - Linha reta - LQR



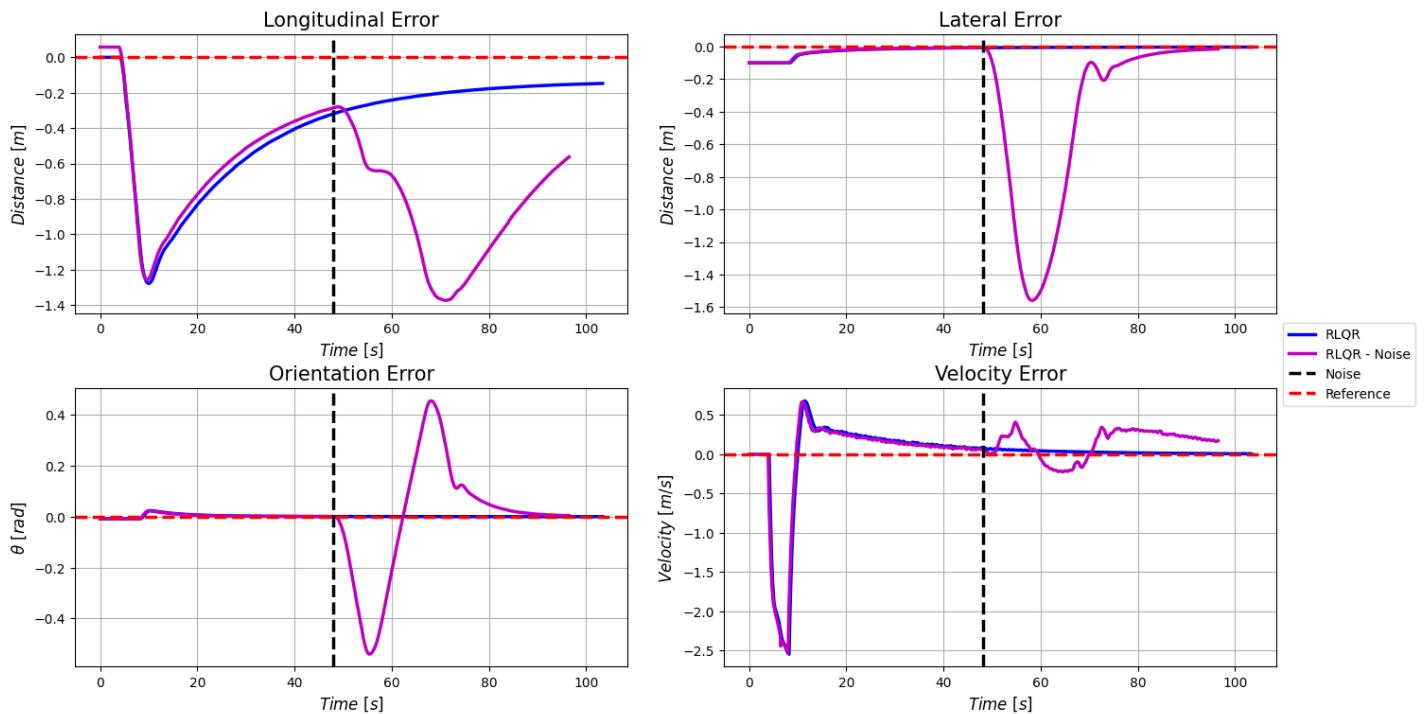
(a) Estados



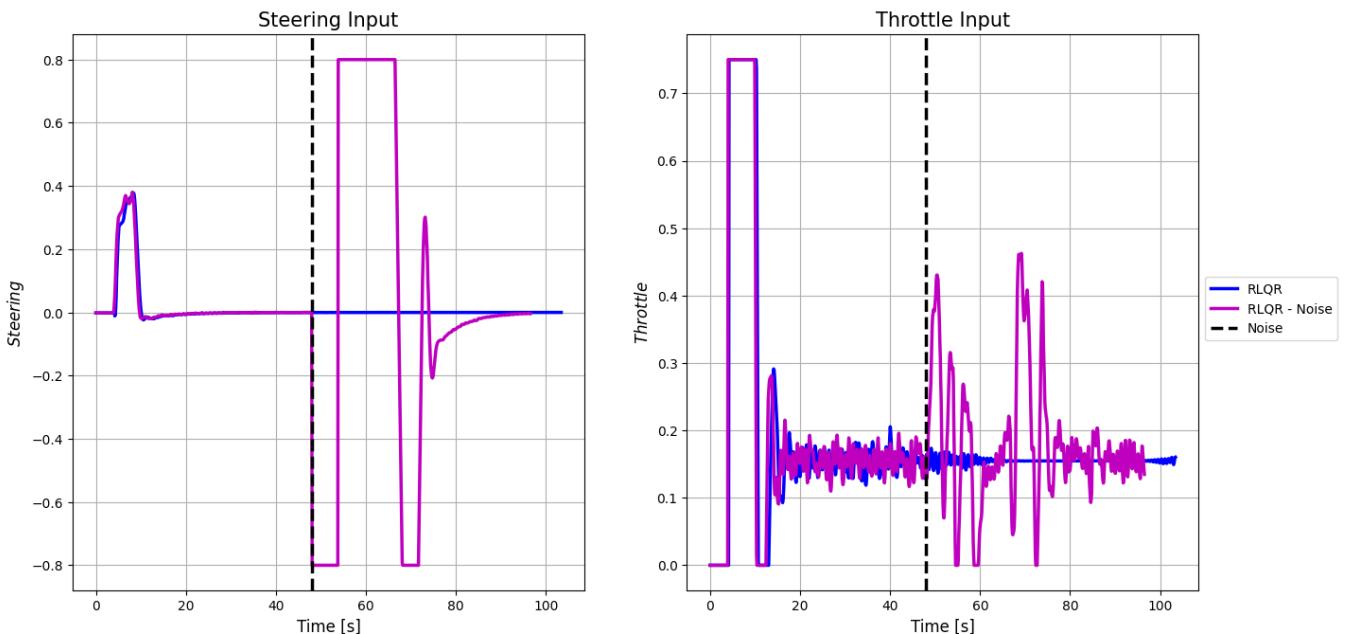
(b) Sinais de controle

Elaborado pelo autor.

Figura 12 – Cenário 1 - Linha reta - RLQR



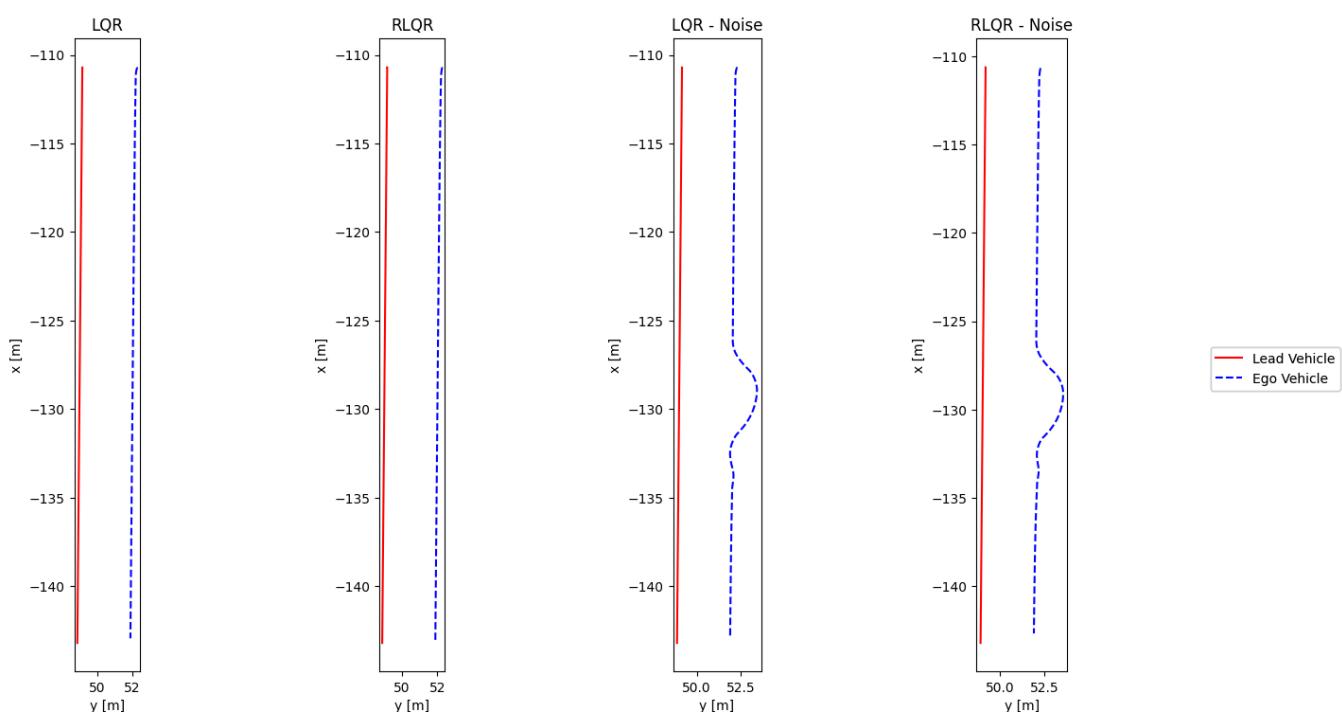
(a) Estados



(b) Sinais de controle

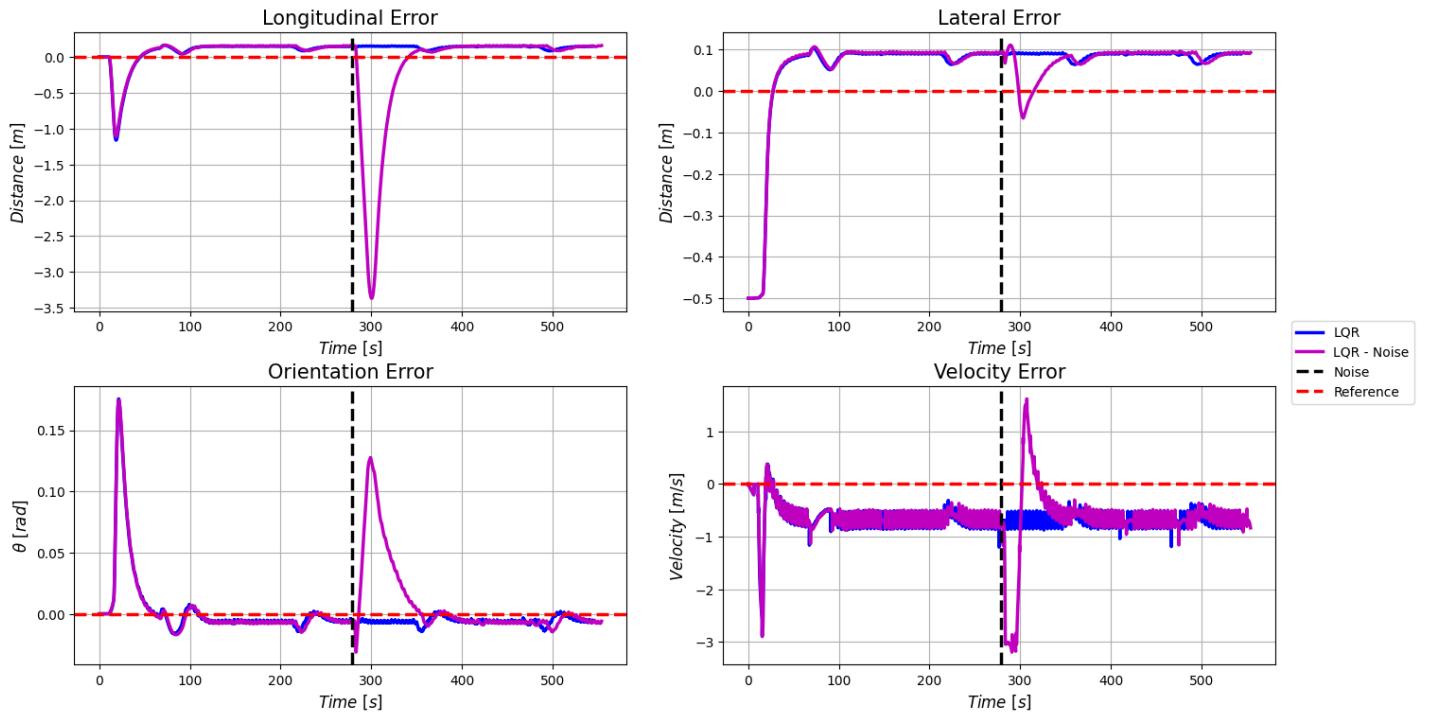
Elaborado pelo autor.

Figura 13 – Cenário 1 - Caminho percorrido

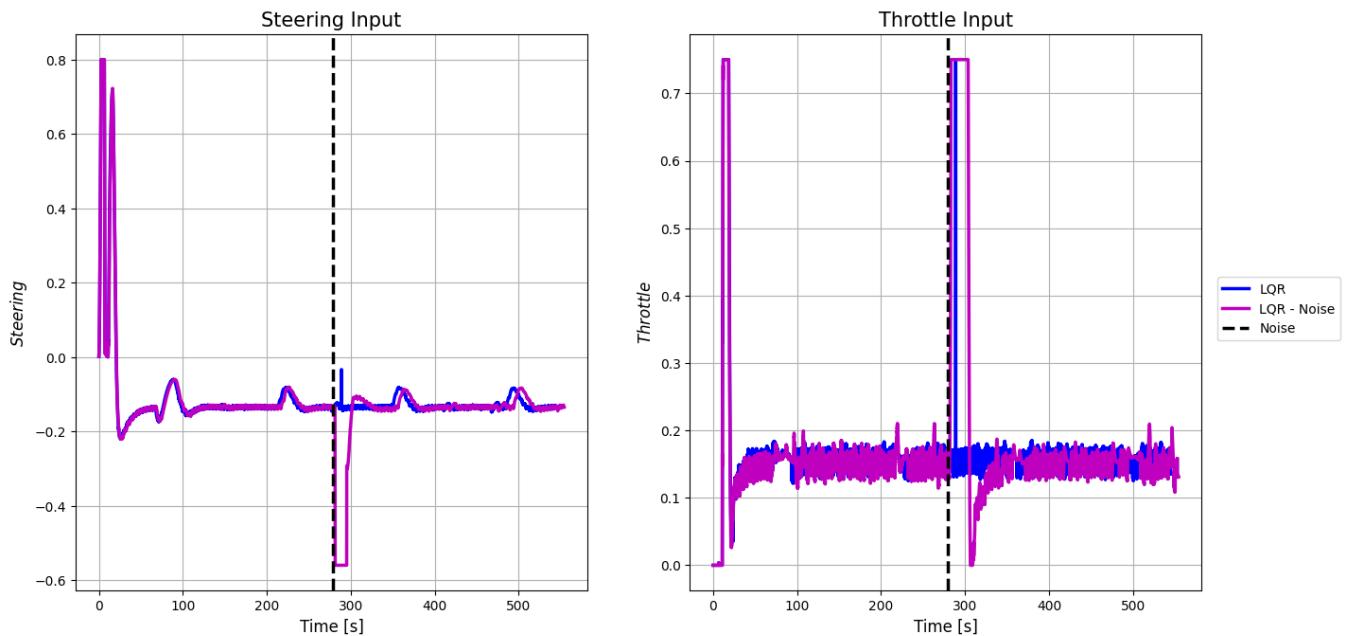


Elaborado pelo autor.

Figura 14 – Cenário 2 - Rotatória - LQR



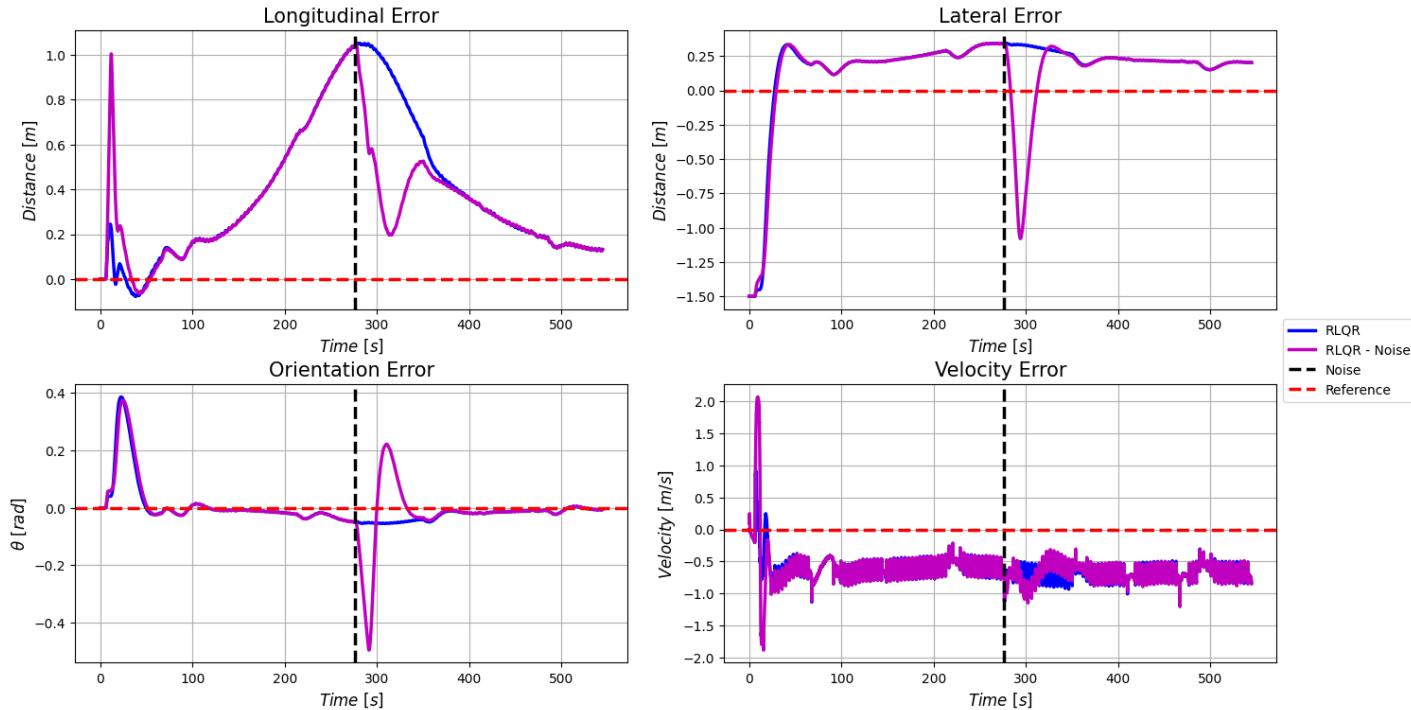
(a) Estados



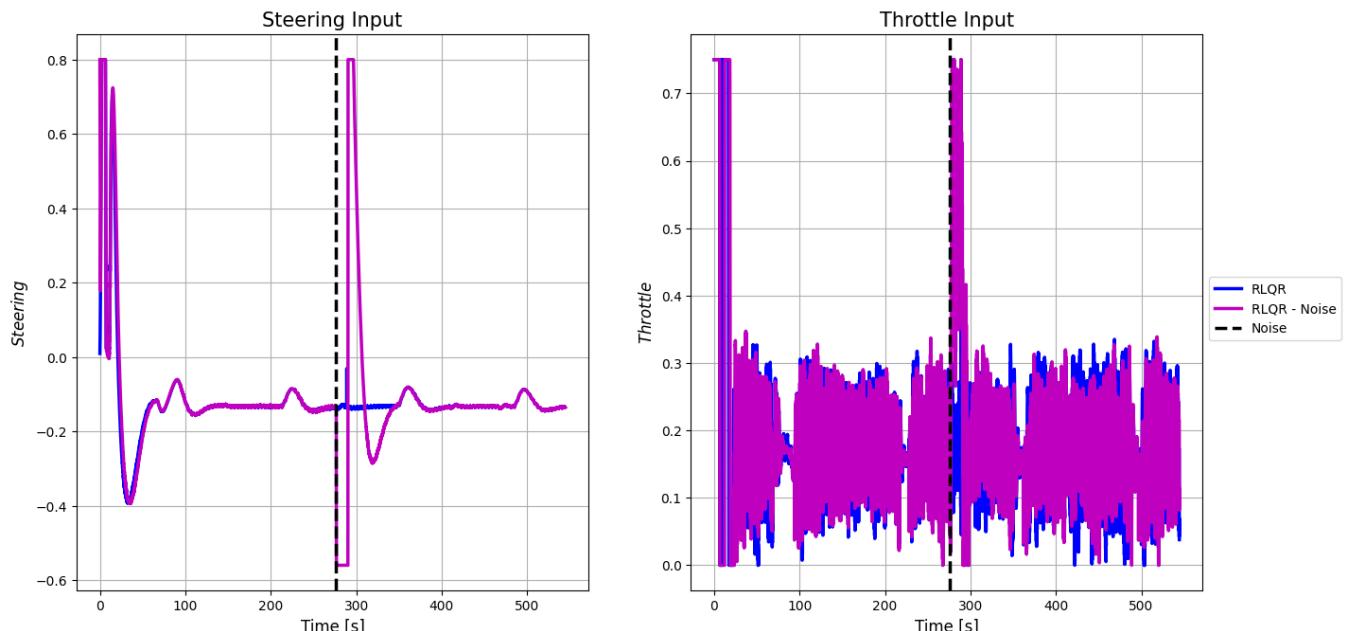
(b) Sinais de controle

Elaborado pelo autor.

Figura 15 – Cenário 2 - Rotatória - RLQR



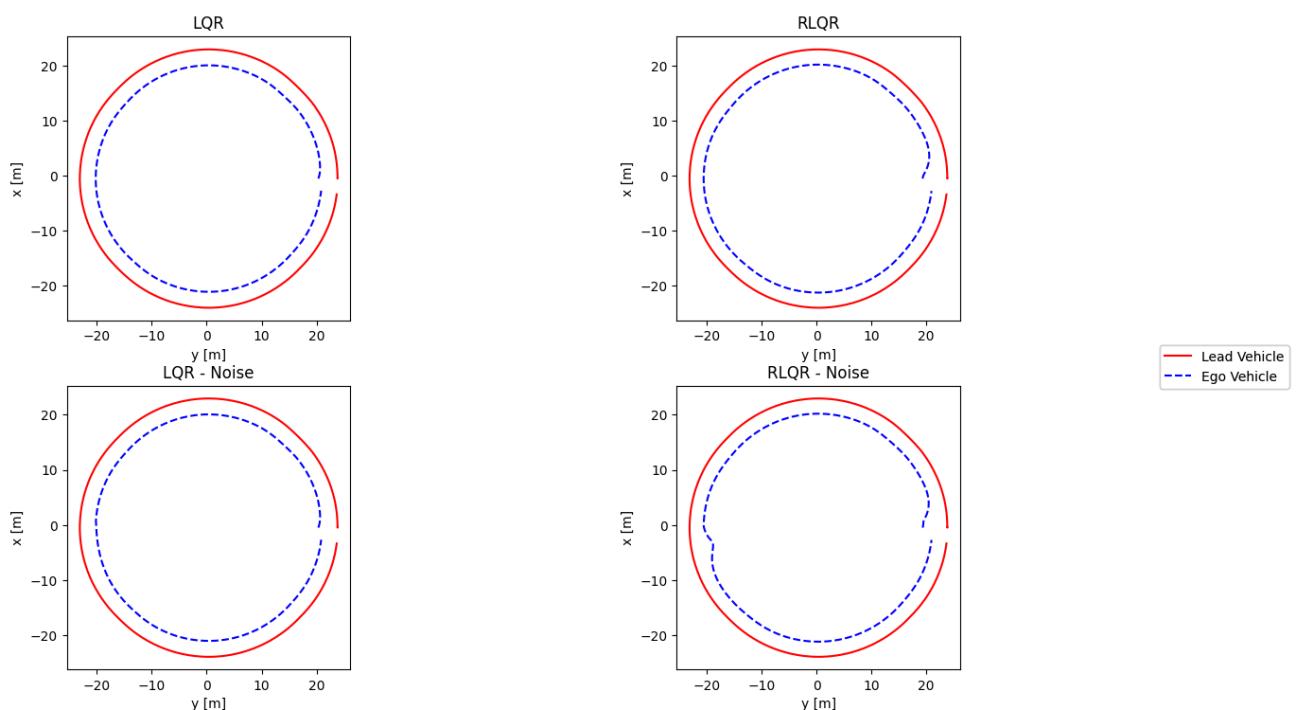
(a) Estados



(b) Sinais de controle

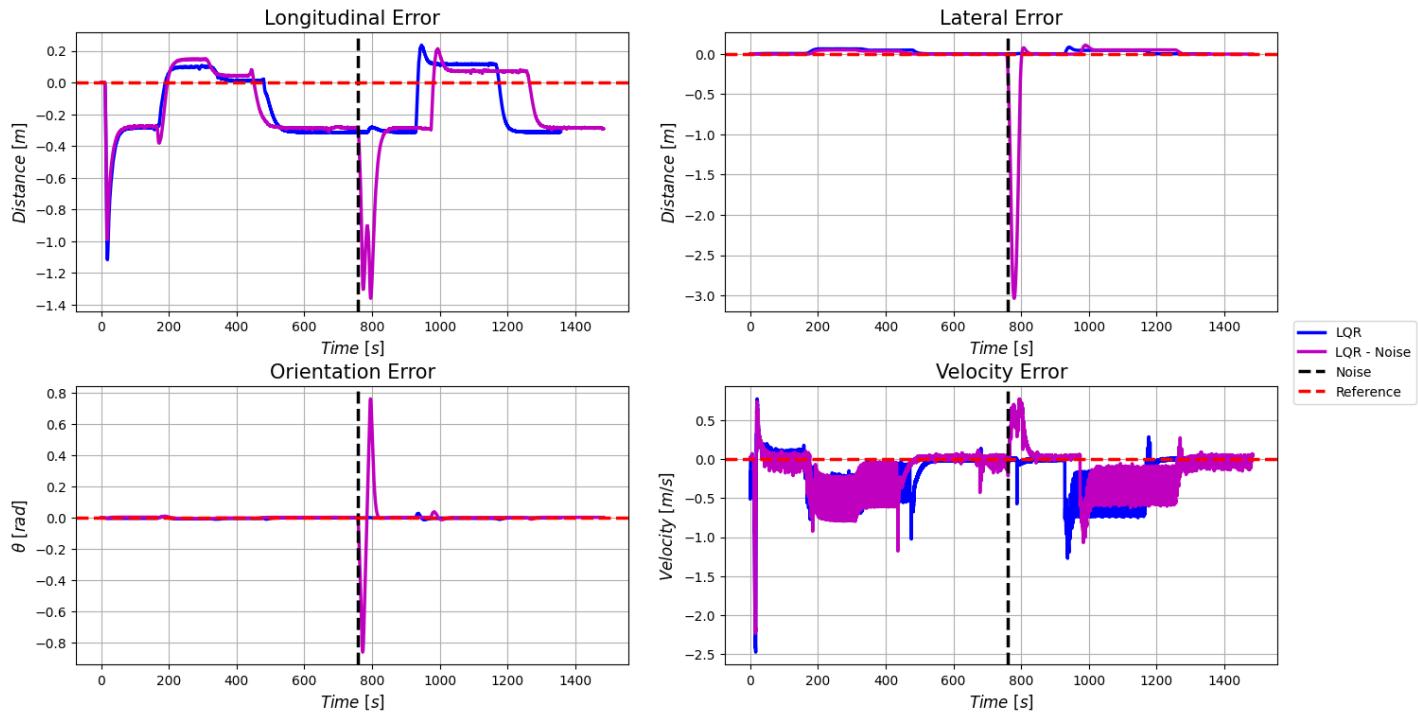
Elaborado pelo autor.

Figura 16 – Cenário 2 - Caminho percorrido

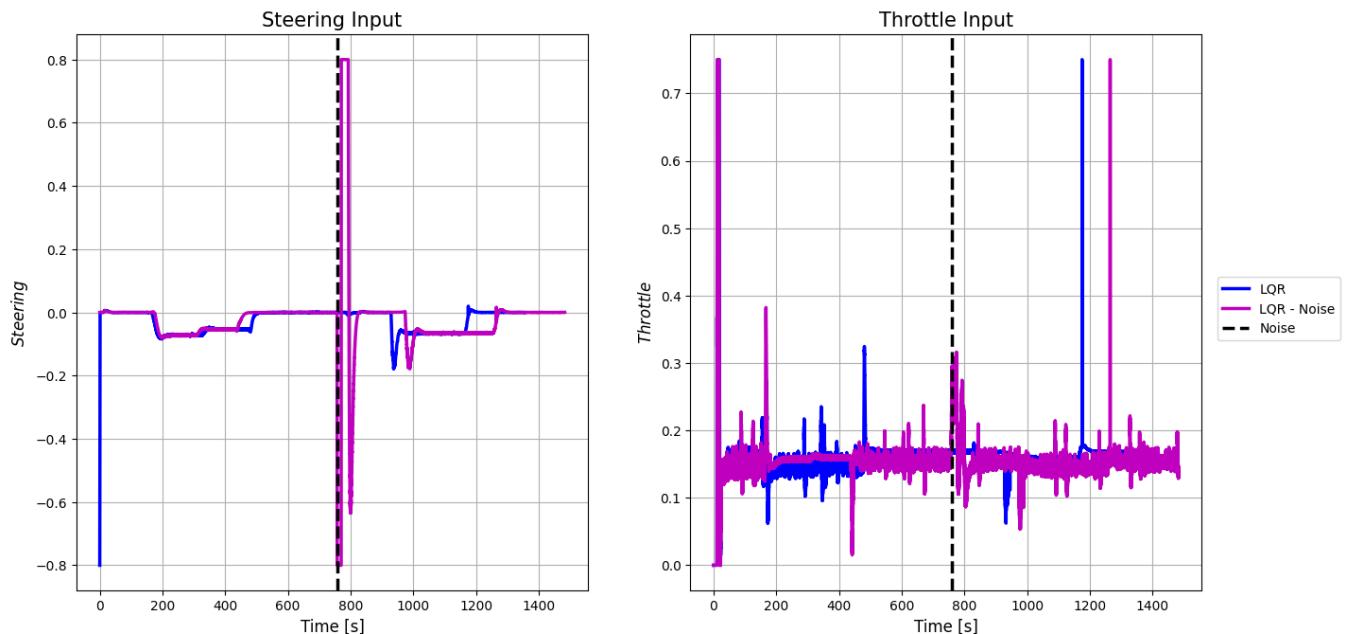


Elaborado pelo autor.

Figura 17 – Cenário 3 - Trajeto curto com retas e curvas - LQR



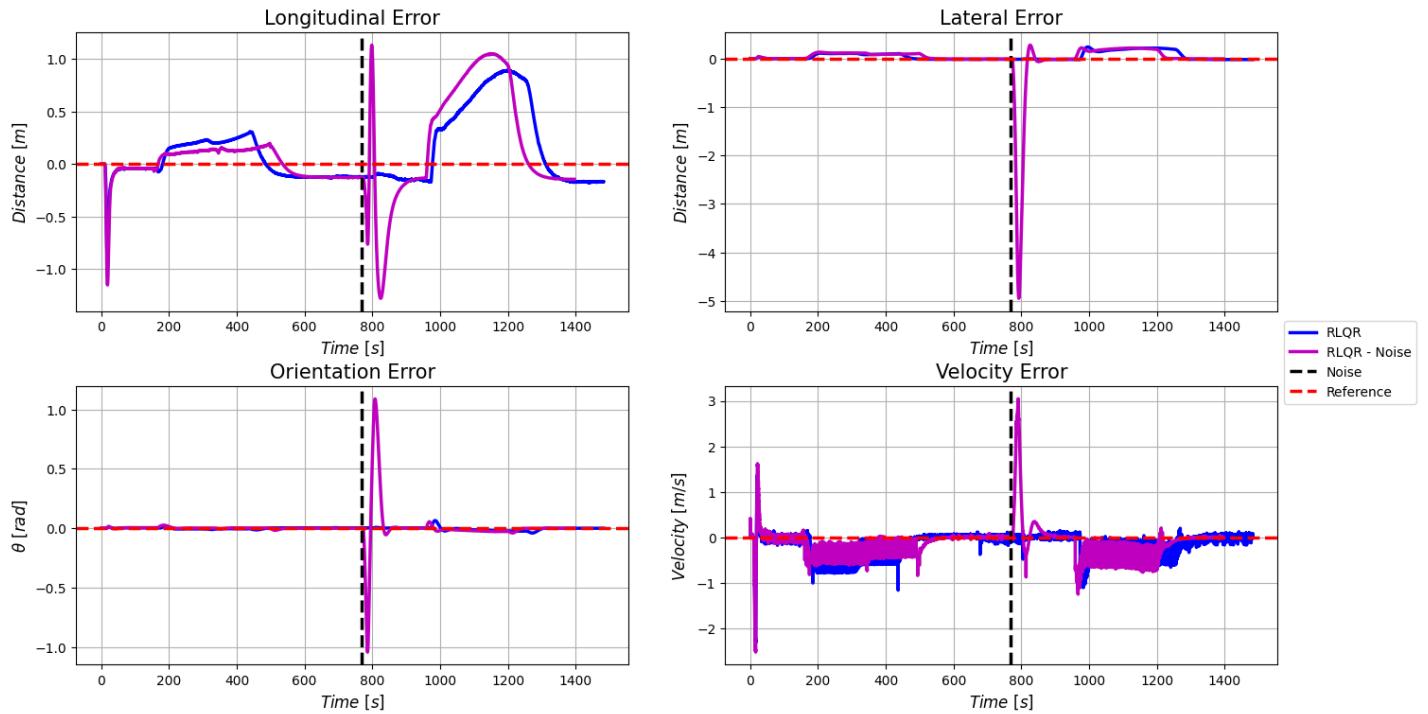
(a) Estados



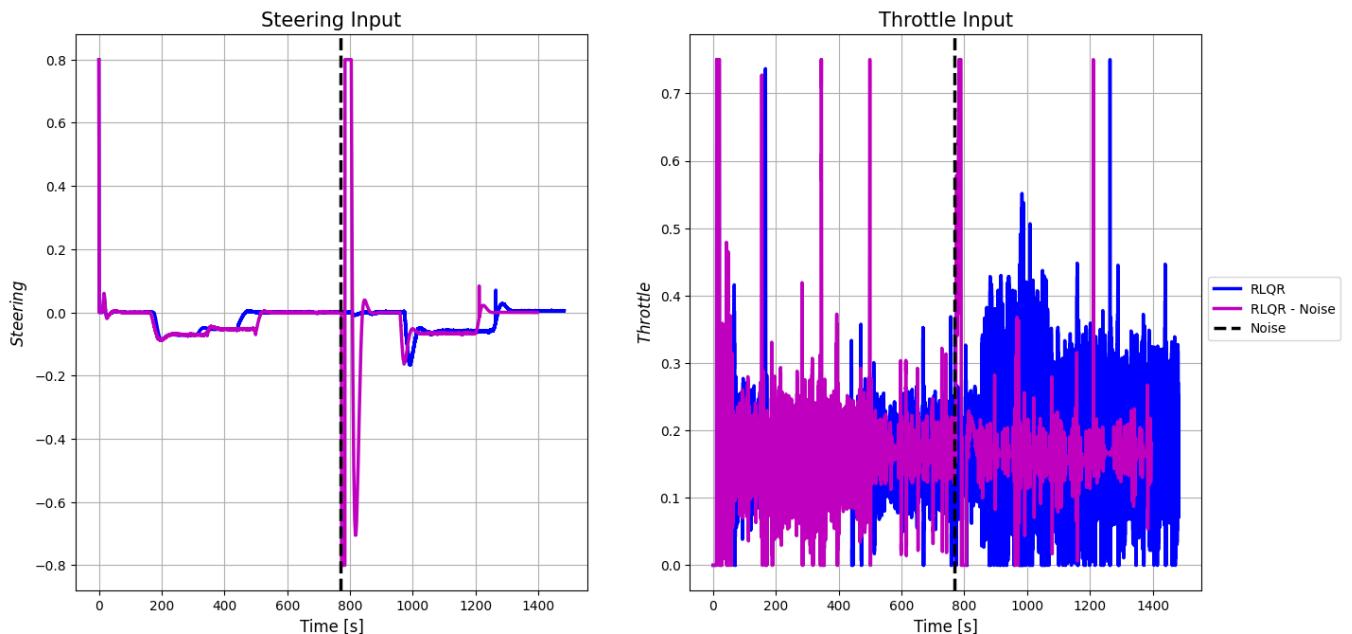
(b) Sinais de controle

Elaborado pelo autor.

Figura 18 – Cenário 3 - Trajeto curto com retas e curvas - RLQR



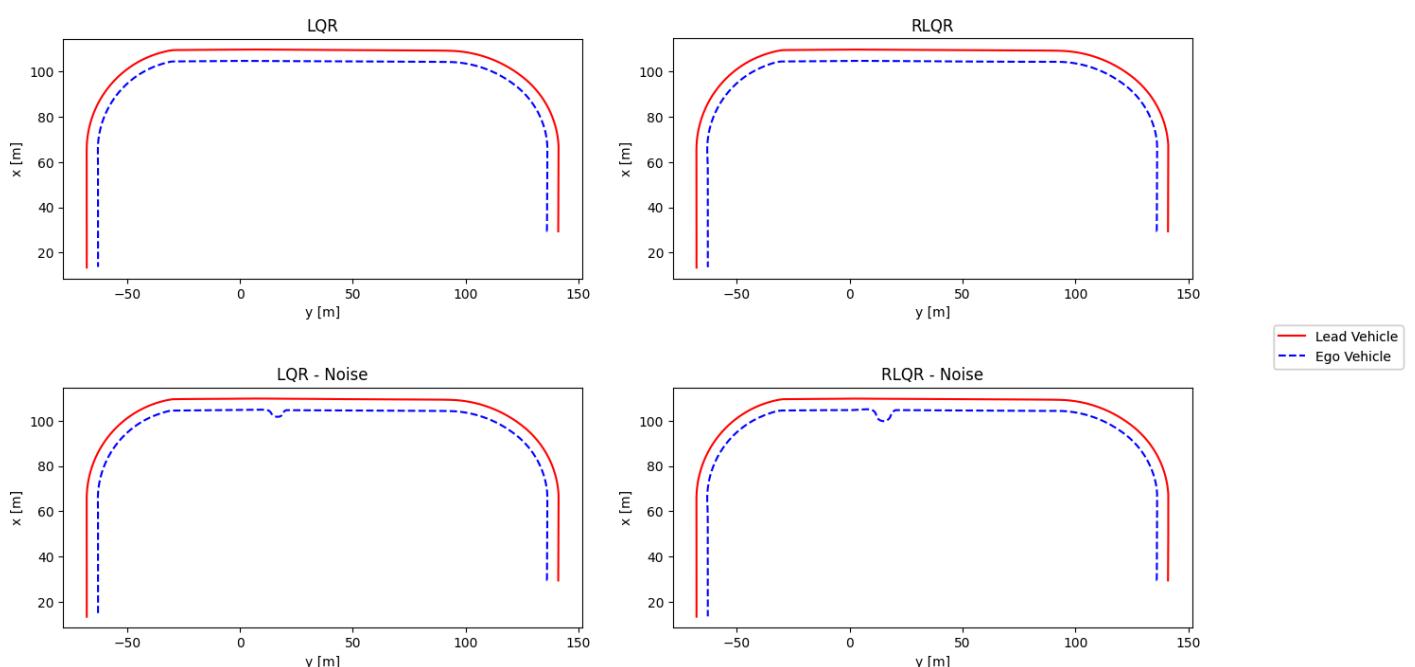
(a) Estados



(b) Sinais de controle

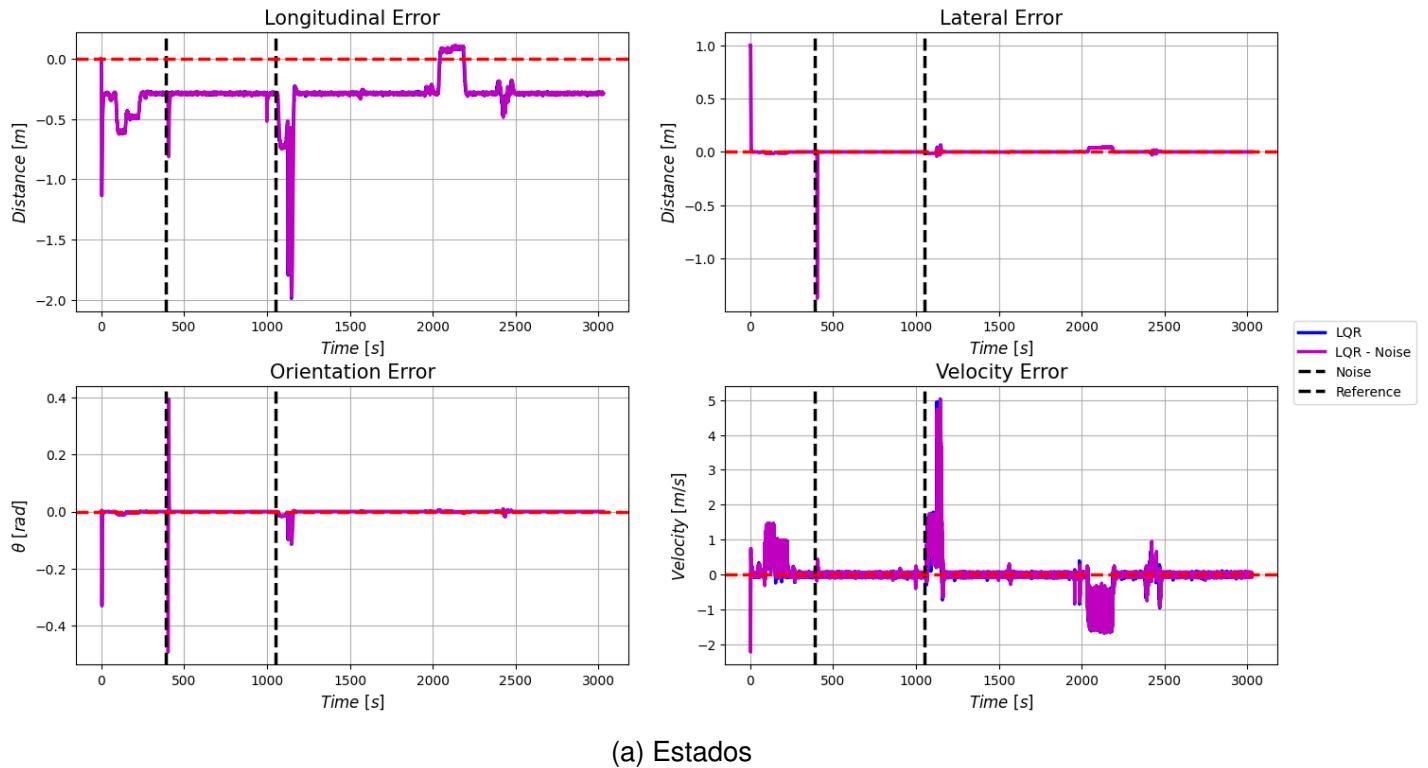
Elaborado pelo autor.

Figura 19 – Cenário 3 - Caminho percorrido

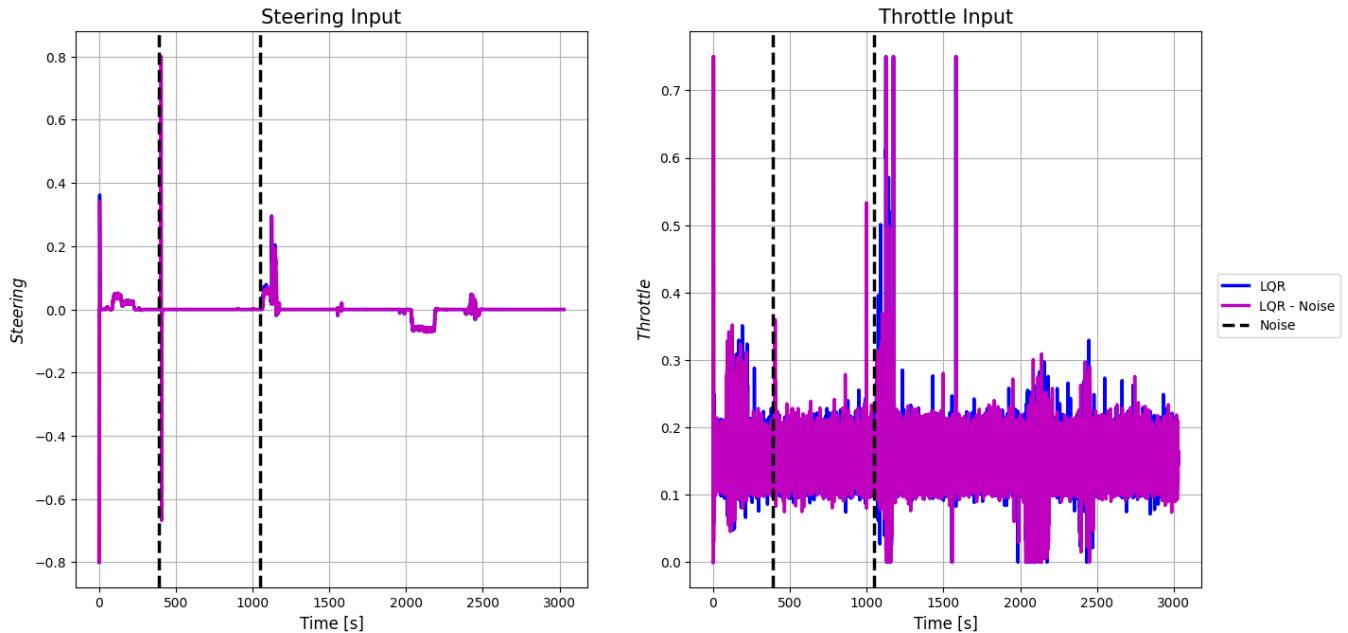


Elaborado pelo autor.

Figura 20 – Cenário 4 - Trajeto longo com retas e curvas - LQR



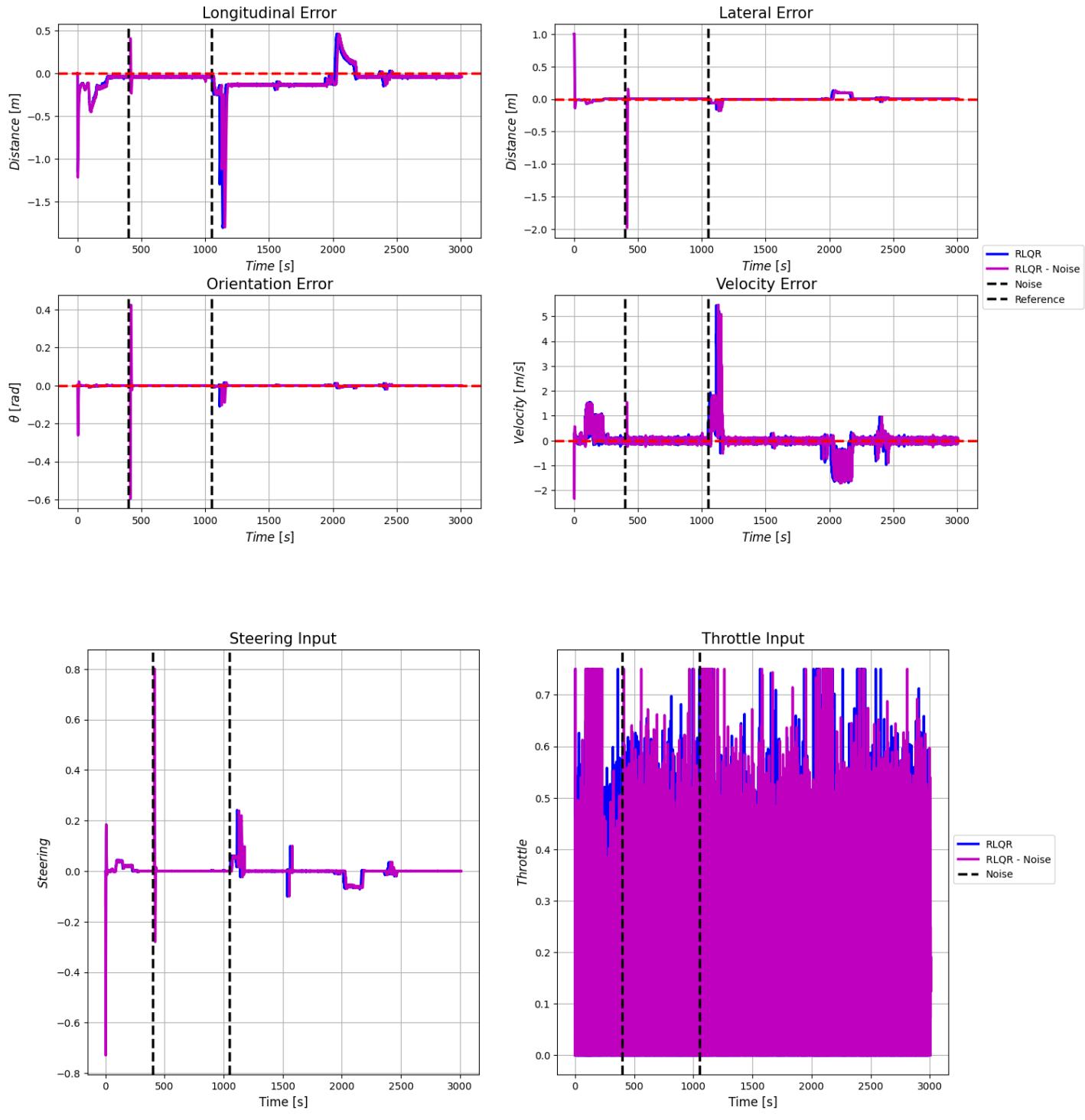
(a) Estados



(b) Sinais de controle

Elaborado pelo autor.

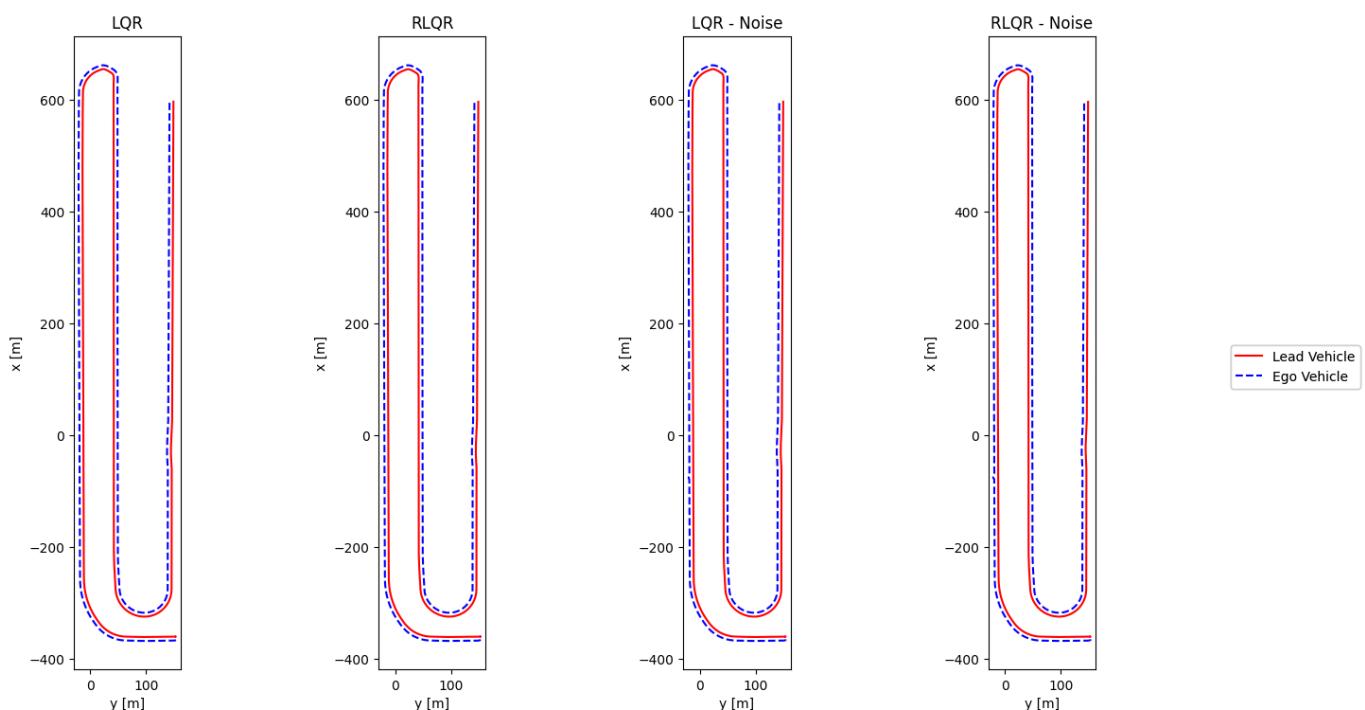
Figura 21 – Cenário 4 - Trajeto longo com retas e curvas - RLQR



(a) Sinais de controle

Elaborado pelo autor.

Figura 22 – Cenário 4 - Caminho percorrido

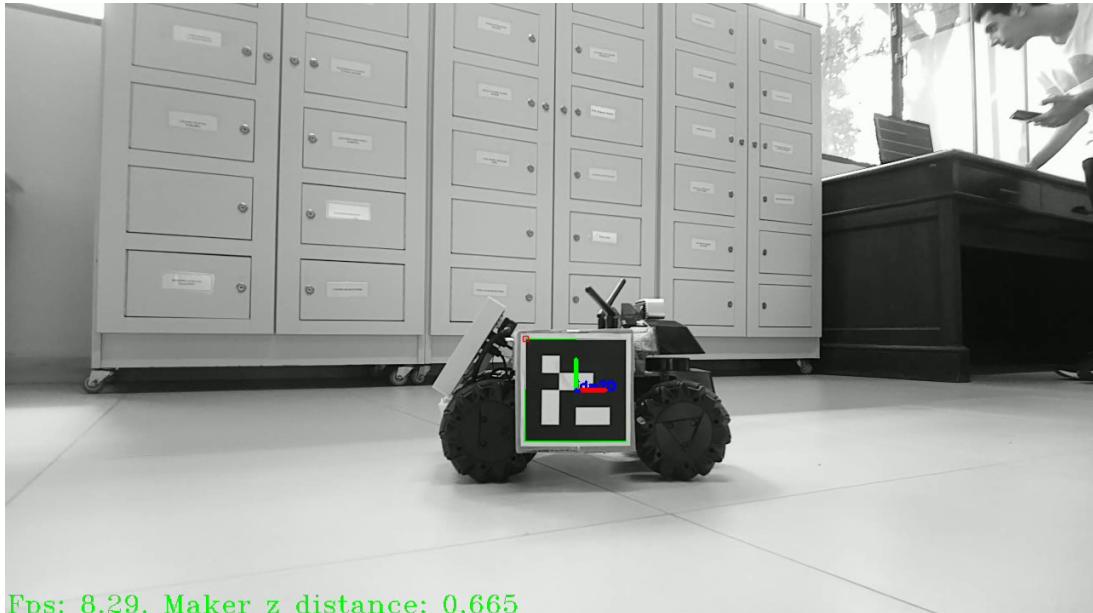


Elaborado pelo autor.

4.2 Testes em laboratório

Para os testes em laboratório, foi utilizada uma frequência de 10Hz no controlador. A Figura 23 mostra a visão da colheitadeira pela câmera.

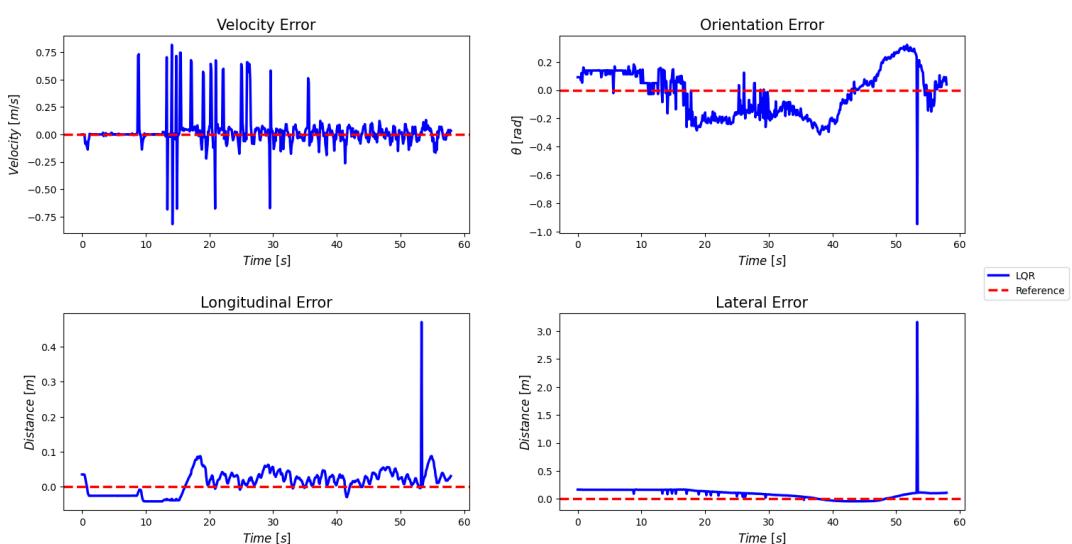
Figura 23 – Visão da câmera no teste em laboratório



Fonte: Elaborado pelo autor.

Um experimento envolvendo retas e curvas foi realizado com os controladores. Para o LQR, os valores dos estados são mostrados na Figura 24.

Figura 24 – Estados LQR - Laboratório



Fonte: Elaborado pelo autor.

É possível perceber que em um determinado momento há uma grande variação nos erros de orientação, posição longitudinal, e posição lateral. Isso ocorreu, pois houve uma

perda de referência no algoritmo de detecção do ArUco, que não identificou o marcador na imagem, como mostrado na Figura 25.

Figura 25 – Perda de referência do ArUco

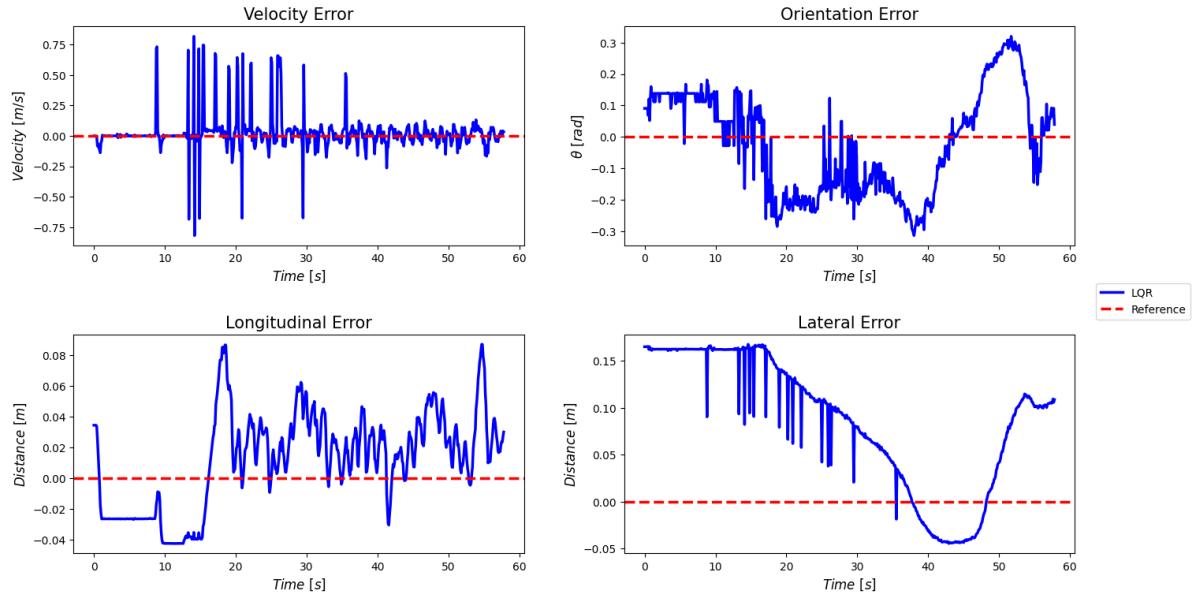


Fonte: Elaborado pelo autor.

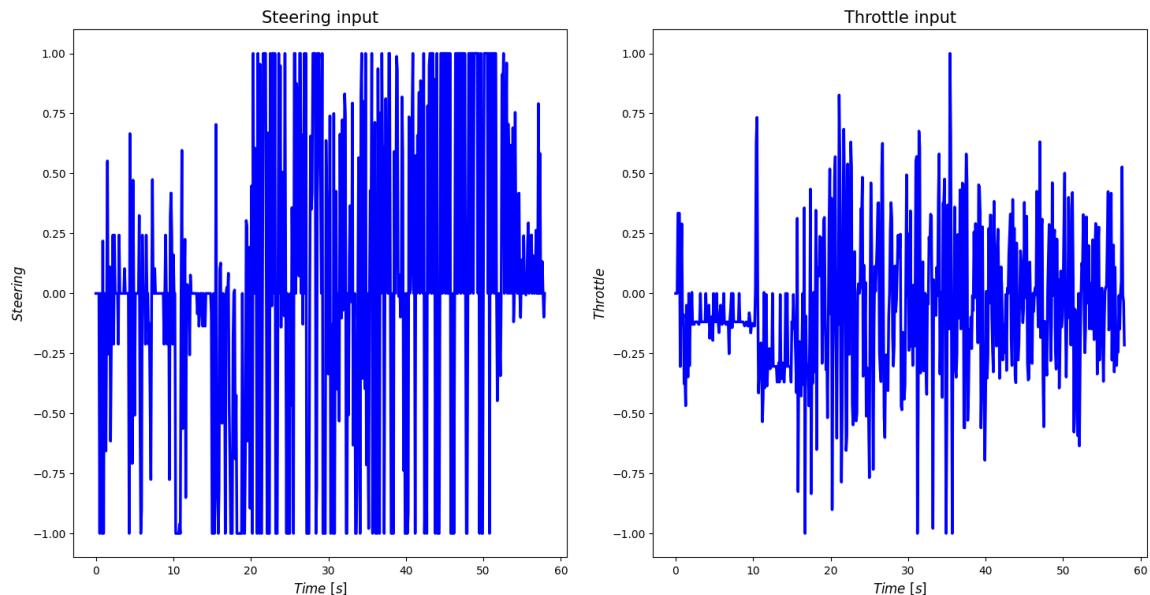
Removendo os valores relativos a essa detecção, tem-se 4 frames removidos. Nesse caso, o valor máximo da posição lateral relativa é de 0,67 m, aproximadamente, o que resulta em um erro máximo de 0,17 m. Já a posição longitudinal relativa não passa de 0,1 m, conforme visto na Figura 26.

Nesse cenário, o controle RLQR não teve um desempenho capaz de acompanhar o percurso. Simulando o cenário em linha reta, no qual a velocidade foi variada linearmente com o tempo, foram obtidos os seguintes resultados para os controladores, mostrados nas Figuras 27 e 28

Figura 26 – LQR - Laboratório



(a) Estados

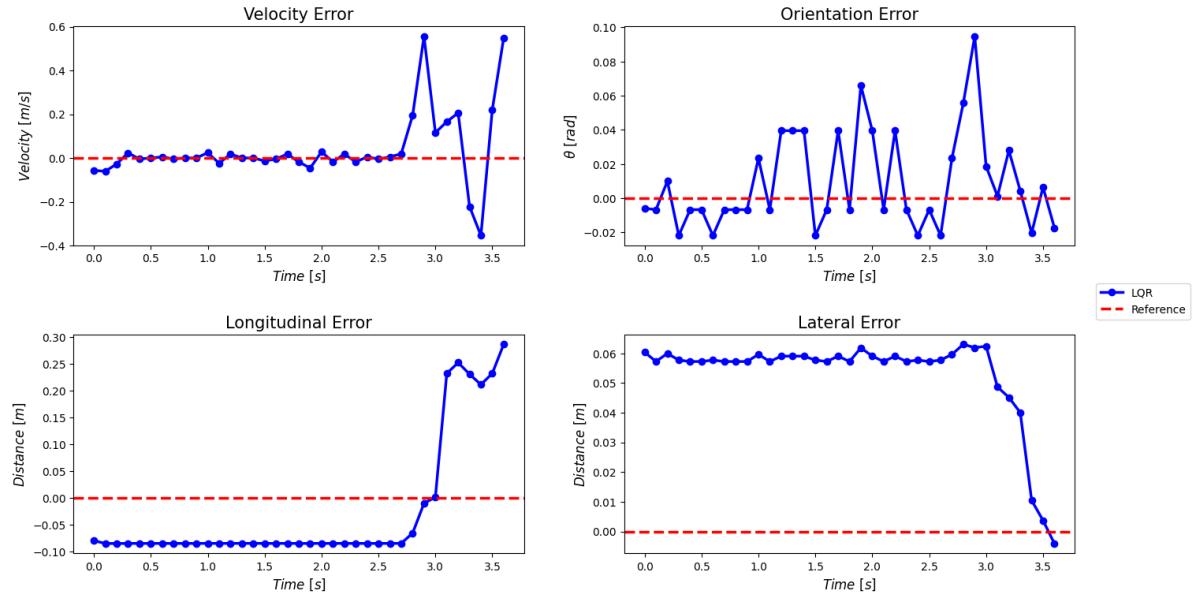


(b) Sinais de controle

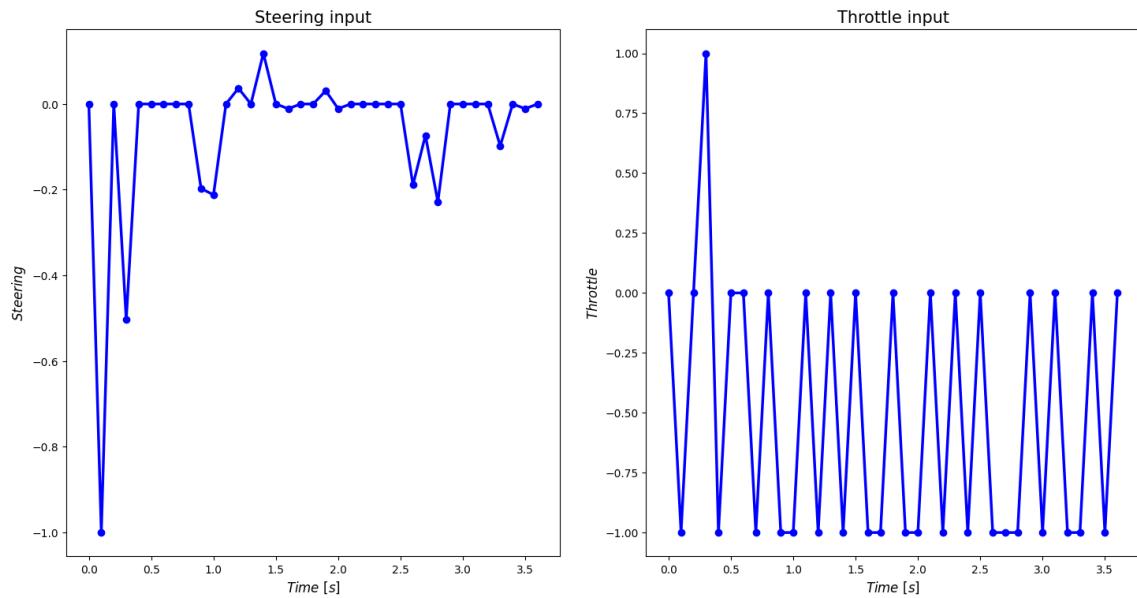
Elaborado pelo autor.

- LQR:

Figura 27 – LQR - Linha Reta - Laboratório



(a) Estados

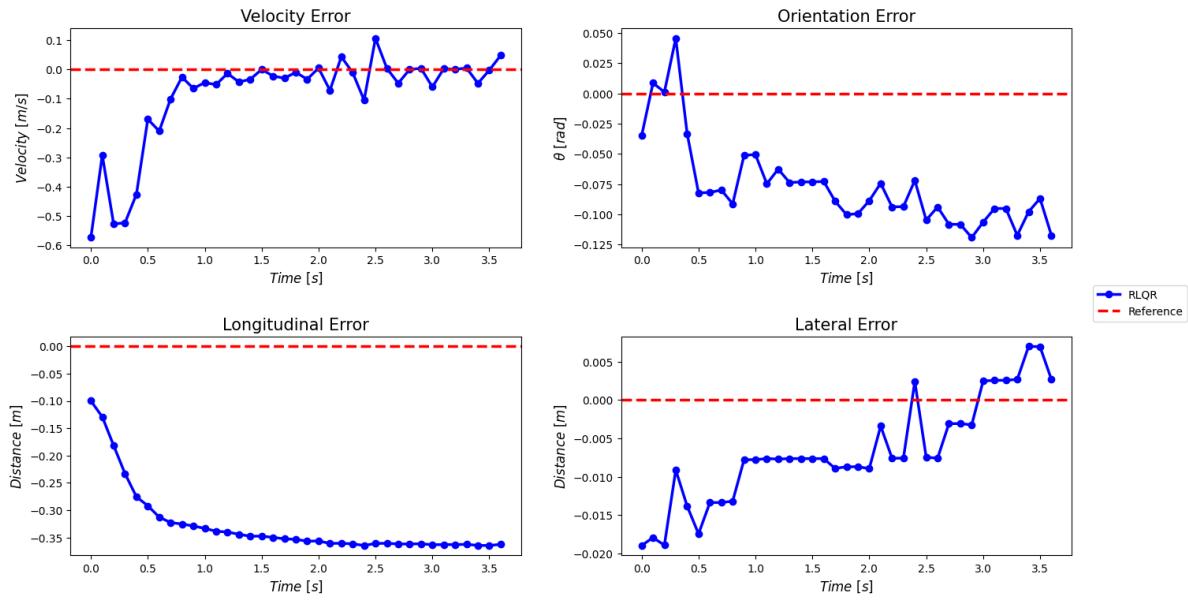


(b) Sinais de controle

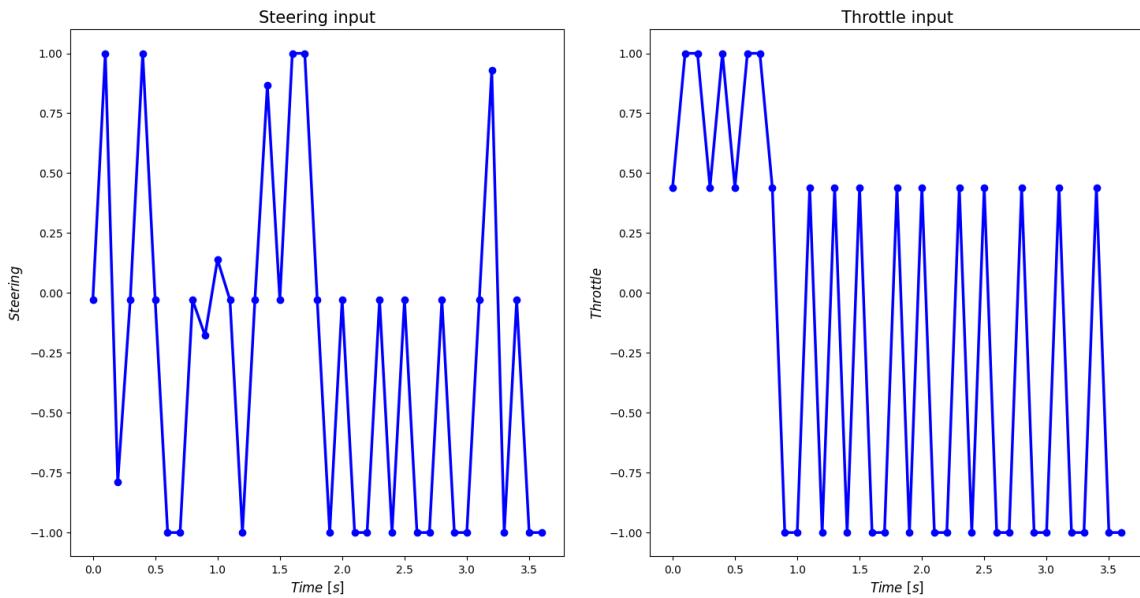
Elaborado pelo autor.

- RLQR:

Figura 28 – RLQR - Linha Reta - Laboratório



(a) Estados



(b) Sinais de controle

Elaborado pelo autor.

5 CONCLUSÃO

Com a implementação dos reguladores LQR e RLQR e testes em simulação e laboratório, foi possível observar que, os controladores tendem a seguir a referência e zerar o erro para os estados do modelo proposto. Entretanto, foram observados alguns pontos que impediram a aplicação prática e resultados esperados em simulações.

Nas simulações, percebe-se um erro de regime permanente na distância longitudinal, como no cenário 1, em linha reta. Isso se deve, provavelmente, ao ponto utilizado como referência para o erro nulo, obtido pelos estados da colheitadeira. Esse ponto é calculado e utilizado no algoritmo de controle, mas, enquanto isso acontece, a posição da colheitadeira já se atualizou e, consequentemente, também foi atualizada a posição desejada. Pelo movimento do veículo se dar principalmente no eixo longitudinal, este é um estado mais afetado por essa defasagem na estimativa. Uma alternativa seria estimar essa defasagem de tempo e estimar um ponto mais à frente considerando a velocidade da colheitadeira ou otimizar o processamento do controlador. Além disso, percebe-se que em curvas o controle não zera o erro de regime, e que, geralmente, apresenta erro positivo na distância longitudinal, erro negativo na velocidade e erro positivo na distância lateral. Isso pode ser consequência da estimativa da posição desejada, conforme mencionado anteriormente.

Na prática, o LQR foi capaz de seguir a referência relativamente bem, com erros pequenos. Isso se deve, principalmente, aos erros associados às estimativas de posições provenientes da câmera. O maior problema foi a perda de referência do marcador pelo algoritmo, o que se provou um problema consistente no decorrer dos diversos experimentos. Uma possível solução seria utilizar mais marcadores e determinar os estados por meio da detecção deles, oferecendo um mecanismo de redundância e maior precisão.

Outro problema prático foi a frequência de aquisição de imagens e do controlador. A frequência máxima de aquisição de imagens era próxima de 15 fps com o controlador operando. Desse modo, a frequência do controlador utilizada foi de 10Hz. Certamente uma frequência maior seria capaz de obter resultados mais próximos do esperado.

Por fim, um possível problema pode ser a modelagem cinemática utilizada. O desenvolvimento de uma modelagem dinâmica pode trazer resultados melhores ao considerar as forças atuantes no veículo.

Como continuação deste trabalho, é possível buscar otimizar os algoritmos, desenvolver formas mais robustas de aquisição e estimativa de dados por meio de odometria visual e utilizar a modelagem dinâmica.

REFERÊNCIAS

- AQEL, M. O. A. et al. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, v. 5, n. 1, p. 1897, out. 2016. ISSN 2193-1801. Disponível em: <https://doi.org/10.1186/s40064-016-3573-7>.
- AUTOINDUSTRIA, R. *Scania amplia oferta com caminhão autônomo para transbordo de cana*. 2022. Disponível em: <https://www.autoindustria.com.br/2022/04/26/scania-amplia-oferta-com-caminhao-autonomo-para-transbordo-de-cana/>.
- BELARDO, G. de C.; CASSIA, M. T.; SILVA, R. P. da. Processos agrícolas e mecanização da cana-de-açúcar. 1. ed. Jaboticabal - SP: Associação Brasileira de Engenharia Agrícola (SBEA), 2015. cap. 18. ISBN 978-85-64681-06-4.
- CERRI, J. P. *Regulador Robusto Recursivo para Sistemas Lineares de Tempo Discreto no Espaço de Estado*. Dissertação (Mestrado) — EESC-USP, 2009.
- DING, Y. *Simple Understanding of Kinematic Bicycle Model*. 2021. Disponível em: <https://dngyan89.medium.com/simple-understanding-of-kinematic-bicycle-model-81cac6420357>.
- DOSOVITSKIY, A. et al. *CARLA: An Open Urban Driving Simulator*. 2017. Disponível em: <https://arxiv.org/abs/1711.03938>.
- DREYFUS, S. Richard bellman on the birth of dynamic programming. *Operations Research*, Informs, v. 50, n. 1, p. 48–51, 2002.
- HIERONYMUS, J. Comparison of methods for geometric camera calibration. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B5, p. 595–599, 2012. Disponível em: <https://doi.org/10.5194/isprsarchives-XXXIX-B5-595-2012>.
- KALAITZAKIS, M. et al. Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers. *Journal of Intelligent & Robotic Systems*, Springer, v. 101, p. 1–26, 2021.
- LI, S.; XU, C.; XIE, M. A robust o (n) solution to the perspective-n-point problem. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 34, n. 7, p. 1444–1450, 2012.
- LIM, K. L.; BRAUNL, T. A review of visual odometry methods and its applications for autonomous driving. *arXiv*, arXiv:2009.09193, 2020. Accessed: 2024-11-30. Disponível em: <https://arxiv.org/abs/2009.09193>.
- LU, A. et al. Adaptive lqr path tracking control for 4ws electric vehicles based on genetic algorithm. In: *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*. 2022. p. 1–6.
- OECD and Food and Agriculture Organization of the United Nations. *OECD-FAO Agricultural Outlook 2021-2030*. Paris: OECD Publishing, 2021. 337 p. Disponível em: <https://www.oecd-ilibrary.org/content/publication/19428846-en>.

RONDELLI, V.; FRANCESCHETTI, B.; MENGOLI, D. A review of current and historical research contributions to the development of ground autonomous vehicles for agriculture. *Sustainability*, v. 14, n. 15, 2022. ISSN 2071-1050. Disponível em: <https://www.mdpi.com/2071-1050/14/15/9221>.

SALVI, J.; ARMANGUÉ, X.; BATLLE, J. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, v. 35, n. 7, p. 1617–1635, 2002. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320301001261>.

SAYED, A. H.; NASCIMENTO, V. H. Design criteria for uncertain models with structured and unstructured uncertainties. In: GARULLI, A.; TESI, A. (Ed.). *Robustness in identification and control*. 1999. p. 159–173. ISBN 978-1-84628-538-7.

TERRA, M. H.; CERRI, J. P.; ISHIHARA, J. Y. Optimal robust linear quadratic regulator for systems subject to uncertainties. *IEEE Transactions on Automatic Control*, v. 59, n. 9, p. 2586–2591, 2014.

ZHAO, J.-S. et al. Design of an ackermann-type steering mechanism. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, v. 227, n. 11, p. 2549–2562, 2013. Disponível em: <https://doi.org/10.1177/0954406213475980>.