Universidade Estadual Paulista
Júlio de Mesquita filho
UNESP
ICTS Sorocaba
Engenharia de Controle e Automação

João Victor Elias Martins
jv.martins@unesp.br

Trabalho Final: Tópicos em Sistemas Dinâmicos

Relatório Simulador Pêndulo Invertido

Sorocaba, 2024

João Victor Elias Martins

Trabalho Final: Tópicos em Sistemas Dinâmicos

Universidade Estadual Paulista

Orientador: Prof. Dr. Felix Mauricio Escalante Ortega
felix.escalante@unesp.br

Sorocaba
2024

Resumo

Este Relatório descreve o controle aplicado à um sistema de pêndulo invertido, para o controle desse sistema foi utilizado um rastreador com ação integral, a fim de, fazer com que o carrinho consiga seguir um sinal de referência.

O pêndulo invertido em questão trata-se de um simulador disponibilizado pelo Prof. Dr. Felix Mauricio Escalante Ortega, implementado em MATLAB. Tanto o modelo linearizado, quanto o controle, foram implementados dentro desse simulador, fazendo uso do modelo não linearizado disponibilizado pelo professor.

Palavras-chaves: Pêndulo invertido; Rastreado com ação integral; Matlab; Controle; Simulação;.

Abstract

This Report describes the control applied to an inverted pendulum system. To control this system, a tracker with integral action was used, in order to make the cart able to follow a reference signal.

The inverted pendulum in question is a simulator made available by Prof. Dr. Felix Mauricio Escalante Ortega, implemented in MATLAB. Both the linearized model and the control were implemented within this simulator, making use of the non-linearized model made available by the teacher.

Key-words:Inverted pendulum; Tracked with full action; Matlab; Control; Simulation.

Sumário

1	INTRODUÇÃO	5
1.1	Objetivos	5
I	DESENVOLVIMENTO	6
2	MATERIAIS E MÉTODOS	7
2.1	Linearização do modelo	7
2.2	Projeto Regulador	8
2.3	Rastreador por ação integral	15
2.4	Simulação no simulink	15
II	RESULTADOS E CONCLUSÕES	18
3	RESUTADOS	19
4	CONCLUSÕES	21
	Referências	22

1 Introdução

O pêndulo invertido é um sistema mecânico clássico com características dinâmicas intrinsecamente instáveis. Apesar de tratar-se de um sistema com estrutura bastante simples, tem sido utilizado em pesquisas com o objetivo de se avaliar o desempenho de diferentes estratégias de controle (POLO; MOLINA; CHICA, 2012) e (NG; CHANG; SONG, 2013). Além disso, o pêndulo invertido representa o princípio de inúmeras aplicações em vários campos do conhecimento, por exemplo: veículos de transporte urbano (PATHAK; FRANCH; AGRAWAL, 2005), robôs aéreos não tripulados em formação (YANG et al., 2013), robôs com duas rodas (GRASSER et al., 2002), exoesqueletos para analisar as forças de reação ao do solo com os membros inferiores (XIANG; ARORA; ABDEL-MALEK, 2011), como representação da coluna vertebral para pacientes com paraplegia de tronco (VANONCINI; HOLDERBAUM; ANDREWS, 2012), entre outras aplicações.

1.1 Objetivos

Neste trabalho prático final do curso, o objetivo é focar nas técnicas de controle abordadas durante a disciplina. Contudo, é necessário modelar o sistema adequadamente para posteriormente adotar métodos de controle mais avançados no futuro. O objetivo é utilizar um sistema real e simples, como o pêndulo invertido, que ilustra o princípio operacional de sistemas mais complexos (ESCALANTE et al., 2023).

Parte I

Desenvolvimento

2 Materiais e métodos

2.1 Linearização do modelo

Tendo em vista o modelo não linear disponível no simulador sem malha de controle (MARTINS, 2024a), foi possível realizar a linearização através do "Model Linearizer" do simulink figura 1, nesse sistema não linear foram definidas uma entrada, tensão de controle, e duas saídas, posição linear do carrinho e posição angular da haste.

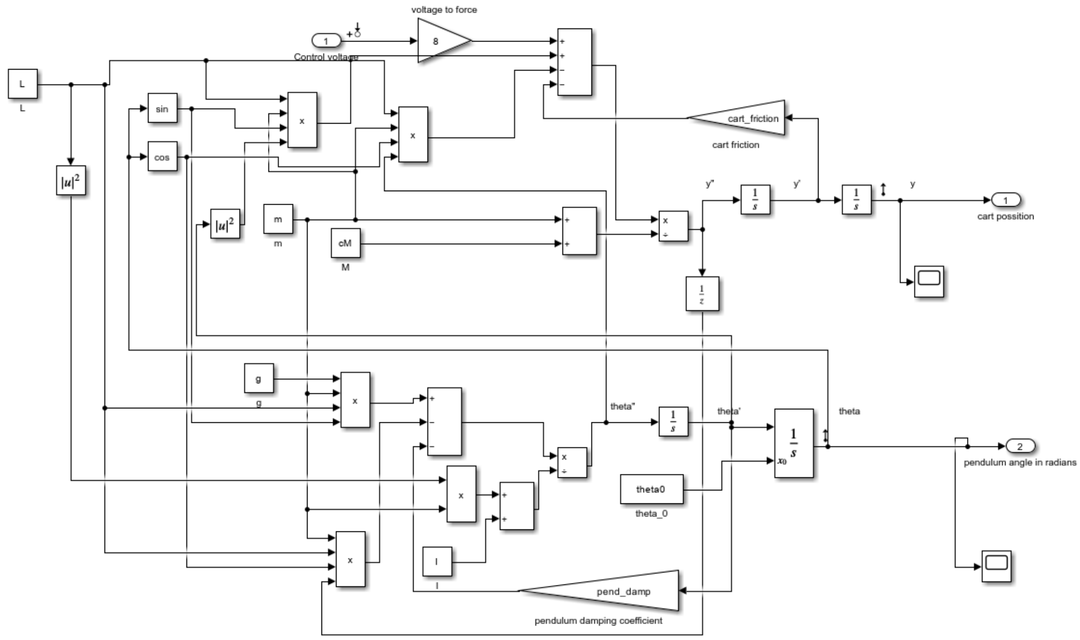

Figura 1 – Modelo não linear do pêndulo invertido.

General Information:

Operating point: Model initial condition

Size: 1 inputs, 2 outputs, 5 states

Linearization Result:

A =

	x1	x2	x3	x4	x5
x1	1	0	1.155e-06	-0.0002141	2.276e-05
x2	0.001	1	5.954e-10	-1.071e-07	1.138e-08
x3	0	0	1	0.006513	-0.000664
x4	0	0	0.001	1	-3.32e-07
x5	-0.01901	0	0.001262	-0.2141	0.02276

B =

	u1
x1	0.003042
x2	1.521e-06
x3	0
x4	0
x5	3.042

C =

	x1	x2	x3	x4	x5
y1	0	1	0	0	0
y2	0	0	0	1	0

D =

	u1
y1	0
y2	0

State Names:
x1 - Integrator
x2 - Integrator1
x3 - Integrator2
x4 - Integrator3
x5 - Unit Delay

Input Channel Names:
u1 - Control voltage

Output Channel Names:
y1 - Integrator1
y2 - Integrator3

Figura 2 – Modelo em espaço de estados devolvido pelo simulink.

2.2 Projeto Regulador

Com o modelo em espaço de estados em mãos é possível iniciar o projeto do controlador, para esse controlador foram definidos alguns parâmetros de controle, sendo esses parâmetros, o tempo de pico máximo de 4 s, e sobressinal máximo de 2%

```
% Polos desejados (tempo de pico Tp = 2s e sobressinal 2%)
Tp = 4;             % Tempo de pico m ximo (em segundos)
OS = 0.02;          % Sobressinal m ximo (2%)

% C lculo do coeficiente de amortecimento (zeta) e
   frequ ncia natural (wn)
zeta = sqrt(log(OS)^2 / (pi^2 + log(OS)^2));  % F rmula
   para zeta
wn = pi / (Tp * sqrt(1 - zeta^2));            % F rmula
   para wn

% Polos dominantes
p1 = (-zeta*wn + 1i*wn*sqrt(1-zeta^2))*1
p2 = (-zeta*wn - 1i*wn*sqrt(1-zeta^2))*1

% Polos adicionais (escolhidos para estabilidade)
poles = [p1, p2, -3, -4, -5, -6];  % 6 polos para um
   sistema aumentado (6 estados)
disp(poles);
```

Após a definição dos parâmetros desejados para o sistema, alocam-se os polos dominantes que garantem a resposta desejada ao sistema. alocam-se também mais quatro polos referentes as outras variáveis de estado, nota-se que um polo a mais foi alocado, isso se deve ao novo estado que será adicionado ao sistema, representado a ação integral do rastreador.

Com os polos definidos é possível calcular os ganhos do regulador, mas como para o sistema de controle deseja-se implementar um rastreador faz-se uso de matrizes aumentadas para considerar o ganho integral.

```
global MODEL
MODEL = {};

clc, close all, clear all;

% Matriz A (5x5)
```

```
MODEL.A = [1 0 1.155e-06 -0.0002141 2.276e-05;
           0.001 1 5.954e-10 -1.071e-07 1.138e-08;
           0 0 1 0.006513 -0.000664;
           0 0 0.001 1 -3.32e-07;
           -0.01901 0 0.001262 -0.2141 0.02276];

% Matriz B (5x1)
MODEL.B = [0.003042;1.521e-06;0;0;3.042];

% Matriz C (2x5)
MODEL.C = [0 1 0 0 0; 0 0 0 -1 0];

% Matriz D (2x1)
MODEL.D = [0; 0];

% Constru  o da matriz aumentada (A_aug: 7x6)
disp('Matrizes aumentadas')
MODEL.A_aug = [MODEL.A zeros(size(MODEL.A, 1), 2); -MODEL.
   C zeros(size(MODEL.C, 1), 2)];
MODEL.B_aug = [MODEL.B; zeros(size(MODEL.C, 1), size(MODEL
   .B, 2))];
MODEL.C_aug = [MODEL.C, zeros(size(MODEL.C, 1), 1)];
MODEL.D_aug = [MODEL.D];

disp('Aa');
disp(MODEL.A_aug)
disp('Bb');
disp(MODEL.B_aug)
disp('Cc');
disp(MODEL.C_aug)
disp('Dd');
disp(MODEL.D_aug)
disp('---');
```

O intuito no começo do projeto do controlador era rastrear as duas variáveis, theta e posição linear, mas observando a controlabilidade do sistema, tem-se que não é possível controlar dessa forma, já que theta depende da posição linear. Realizando uma troca de base, a matriz aumentada que antes era 7x7, se torna 6x6, dessa vez controlável e acomodando o estado do ganho integral.

```
%checar controlabilidade
ctrb_aug = ctrb(MODEL.A_aug, MODEL.B_aug);
rank_ctrb = rank(ctrb_aug);

if rank_ctrb < size(MODEL.A_aug, 1)
    disp('O sistema aumentado n o   control vel!');
    disp(MODEL.A_aug);
    disp(MODEL.B_aug);
    disp('---');
else
    disp('O sistema aumentado   control vel!');
    disp(MODEL.A_aug);
    disp(MODEL.B_aug);
    disp('---');
end

%tranformar em control vel, matriz controlabilidade
MODEL.C_ctr = ctrb(MODEL.A_aug, MODEL.B_aug);

%calculo base controlavel
T = orth(MODEL.C_ctr);

%matrizes a e b controlaveis
MODEL.A_ctr = T \ MODEL.A_aug * T;
MODEL.B_ctr = T \ MODEL.B_aug;

Co = ctrb(MODEL.A_ctr, MODEL.B_ctr);
rank_Co = rank(Co);

if rank_Co < size(MODEL.A_ctr, 1)
    disp('O sistema aumentado n o   control vel!');
    disp(MODEL.A_ctr);
    disp(MODEL.B_ctr);
    disp('---');
else
```

```
    disp('O sistema aumentado   control vel!');
    disp(MODEL.A_ctr);
    disp(MODEL.B_ctr);
    disp('---');
end
```

As figuras 3 e 4, apresentam as matrizes aumentadas A e B antes e depois da mudança de base para garantir controlabilidade

```
O sistema aumentado não é controlável!
    1.0000         0    0.0000   -0.0002    0.0000         0         0
    0.0010    1.0000    0.0000   -0.0000    0.0000         0         0
         0         0    1.0000    0.0065   -0.0007         0         0
         0         0    0.0010    1.0000   -0.0000         0         0
   -0.0190         0    0.0013   -0.2141    0.0228         0         0
         0   -1.0000         0         0         0         0         0
         0         0         0    1.0000         0         0         0

    0.0030
    0.0000
         0
         0
    3.0420
         0
         0
```

Figura 3 – Matrizes Aumentadas A e B antes da troca de base.

```
O sistema aumentado é controlável!
    0.0227   -0.0153    0.0847   -0.0331    0.0103    0.1938
    0.0012    1.0000    0.0015   -0.0018    0.0001    0.0043
   -0.0000    0.0014    1.0101   -0.3990    0.0162    0.3327
    0.0000   -0.0000    0.0257   -0.0085    0.0419    0.8413
   -0.0000    0.0000    0.0011    0.0408    0.9983   -0.0298
   -0.0000    0.0000    0.0000    0.0000    0.0011    1.0000

   -3.0420
    0.0001
   -0.0000
    0.0000
   -0.0000
   -0.0000
```

Figura 4 – Matrizes Aumentadas A e B após a troca de base.

Finalmente calcula-se os ganhos do regulador, para esse cálculo foi necessário utilizar a função "acker" ao invés da função "place", por conta da instabilidade e a dificuldade de controlar o sistema.

```
% C lculo do ganho K utilizando a fun  o 'acker',14
MODEL.K = (acker(MODEL.A_ctr, MODEL.B_ctr, poles)) / 1e+0;

%MODEL.K1_5 = MODEL.K(1:5);

% Exibi  o do ganho K
disp('Ganho do controlador K:');
%MODEL.K(1) = MODEL.K(1)
disp(MODEL.K);

% teste dos polos e zeros do sistema em malha fechada
sys_aug = ss(MODEL.A_ctr - (MODEL.B_ctr * MODEL.K), MODEL.
   B_ctr, MODEL.C_aug, MODEL.D_aug)
MODEL.A_exp = MODEL.A_ctr - (MODEL.B_ctr * MODEL.K);

step(sys_aug)

% zero
zeros_aug = tzero(sys_aug);

%polo
polos_aug = pole(sys_aug);

disp('Zeros do sistema controlado (com K):');
disp(zeros_aug);

disp('Polos do sistema controlado (com K):');
disp(polos_aug);
```

A figura 5 apresenta a resposta ao degrau do regulador, nota-se que após a troca de base as variáveis se inverteram, sendo theta o primeiro, retornando ao angulo 0°, e a posição linear tentando seguir a referência. como os ganhos são muito altos a resposta apresenta ordens de grandeza muito baixas.

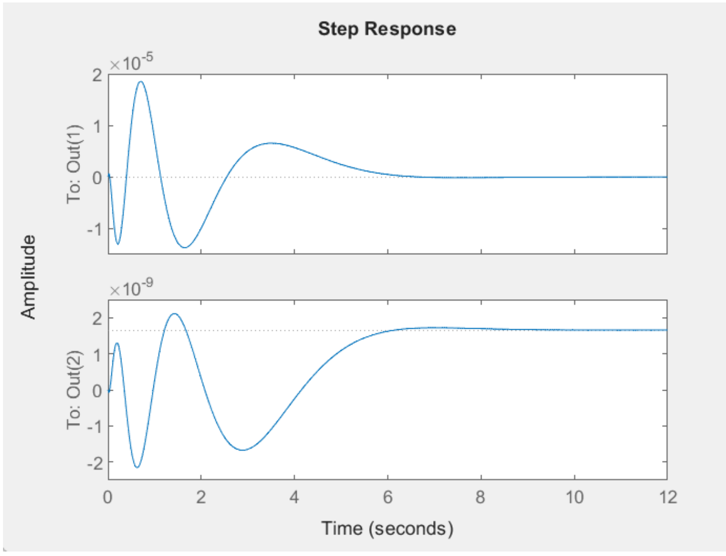

Figura 5 – Resposta ao degrau do regulador

As figuras 6 e 7 apresentam respectivamente os ganhos do controlador e os polos do sistema em malha fechada com o regulador, observa-se que foi possível alcançar os polos desejados para o sistema, e que os ganhos apresentam uma ordem de grandeza muito elevada.

```
Ganho do controlador K:
   1.0e+14 *

   -0.0000    -0.0000    -0.0000    -0.0003    -0.0091    -3.1951
```

Figura 6 – Ganhos K para o sistema.

```
Zeros do sistema controlado (com K):
Polos do sistema controlado (com K):
  -6.0000 + 0.0000i
  -5.0000 + 0.0000i
  -4.0000 + 0.0000i
  -3.0000 + 0.0000i
  -0.9780 + 0.7854i
  -0.9780 - 0.7854i
```

Figura 7 – Polos do sistema controlado.

2.3 Rastreador por ação integral

Por último projeta-se o rastreador, o código abaixo apresenta o cálculo para o ganho integral, nota-se que foi necessário diminuir a ordem de grandeza desse ganho, pois ele trazia instabilidade ao sistema caso fosse mantido alto. A figura 8 apresenta o ganho obtido.

```
a = (MODEL.A_ctr - (MODEL.B_ctr * MODEL.K))^-1;
b = MODEL.C_aug * a * MODEL.B_ctr;
MODEL.N = (-1/b)/1e+10;

disp('ganho rastreador');
disp(MODEL.N);
```

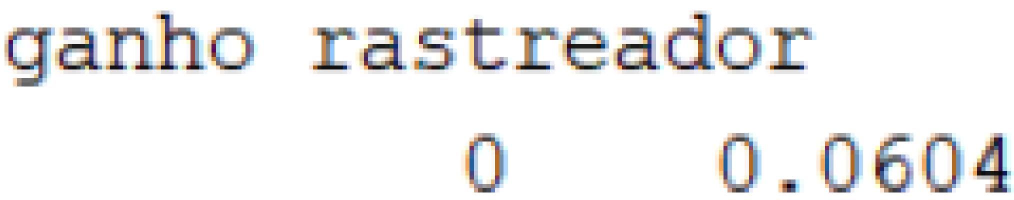

Figura 8 – Ganhos do rastreador.

2.4 Simulação no simulink

Com o modelo em mãos foi desenvolvido uma malha de controle no simulink, algumas observações importantes para a simulação foram, a necessidade de aplicar um ganho na saída do sistema de 0.5e+7 para que o sistema atingisse o setpoint. E por falta de um observador de estados foi necessário recuperar os estados direto do sistema de espaço de estados, sendo necessário reimplementar o bloco de espaço de estados no simulink.

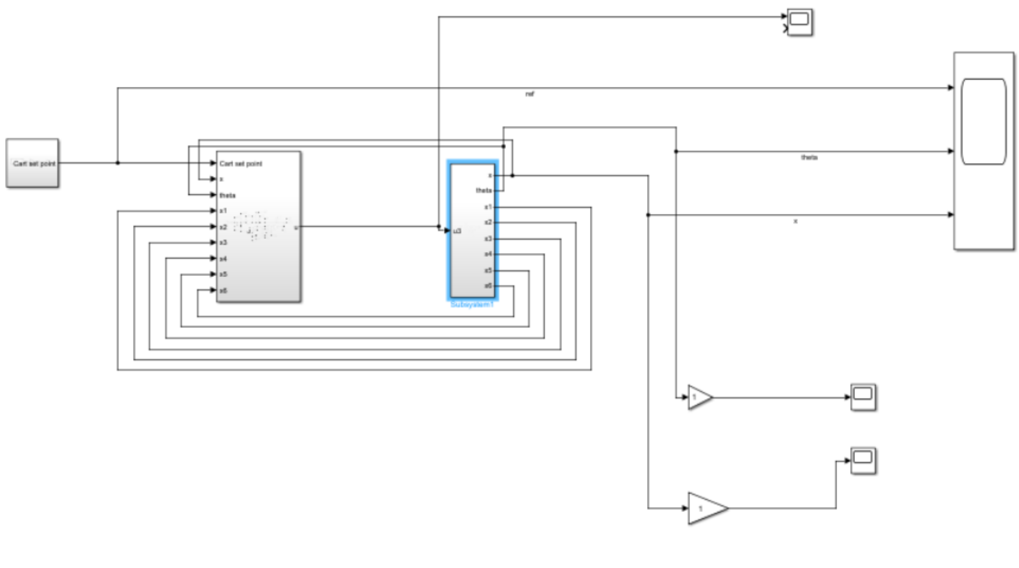

Figura 9 – Sistema montado.

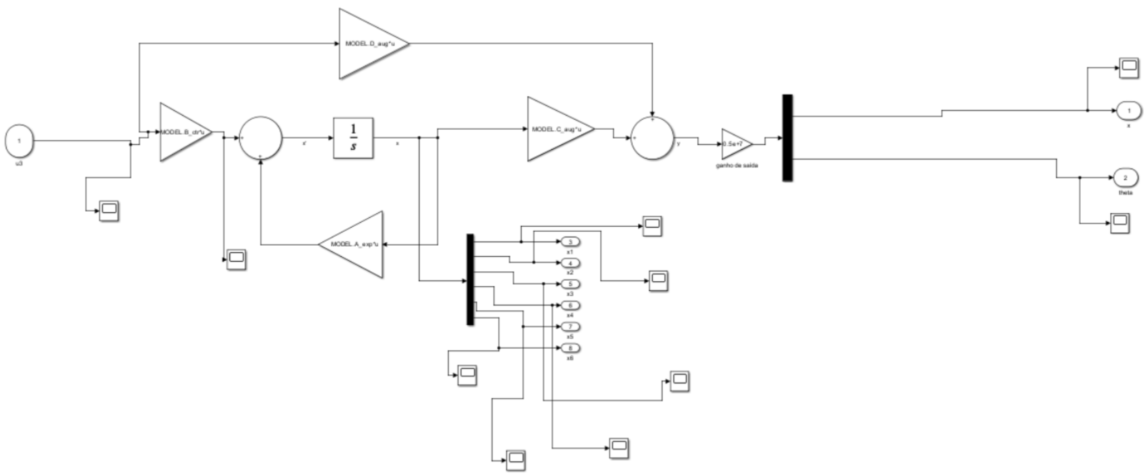

Figura 10 – Modelo em espaço de estados.

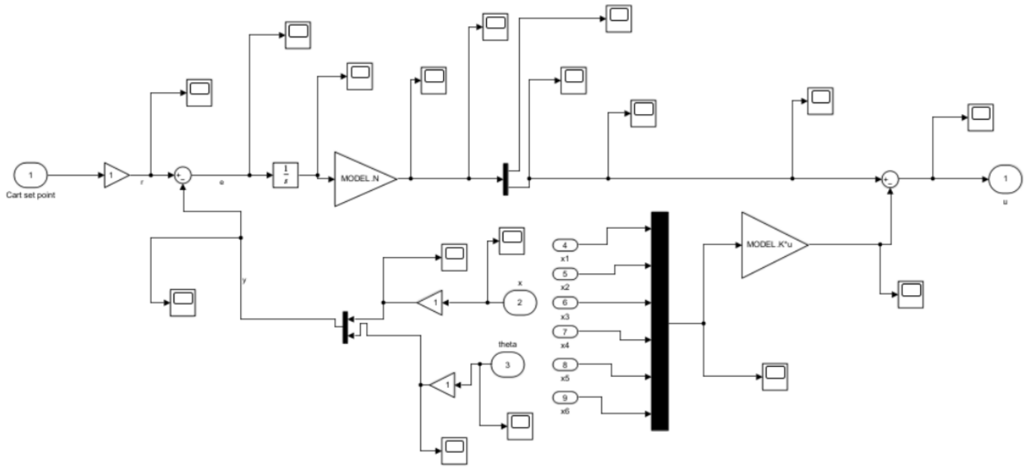

Figura 11 – Rastreador por ação integral.

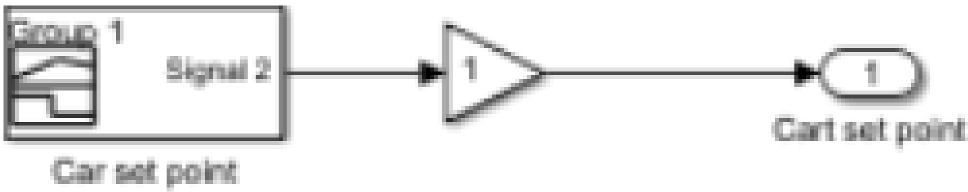

Figura 12 – Sinal de referência.

Parte II

Resultados e conclusões

3 Resutados

Após as simulações foi possível obter tanto os gráficos de posição angular e posição linear, junto dos sinais de controle e de estados. tanto a simulação do pêndulo funcionando disponível no Youtube (MARTINS, 2024b). O projeto completo está disponível no GitHub em (MARTINS, 2024a).

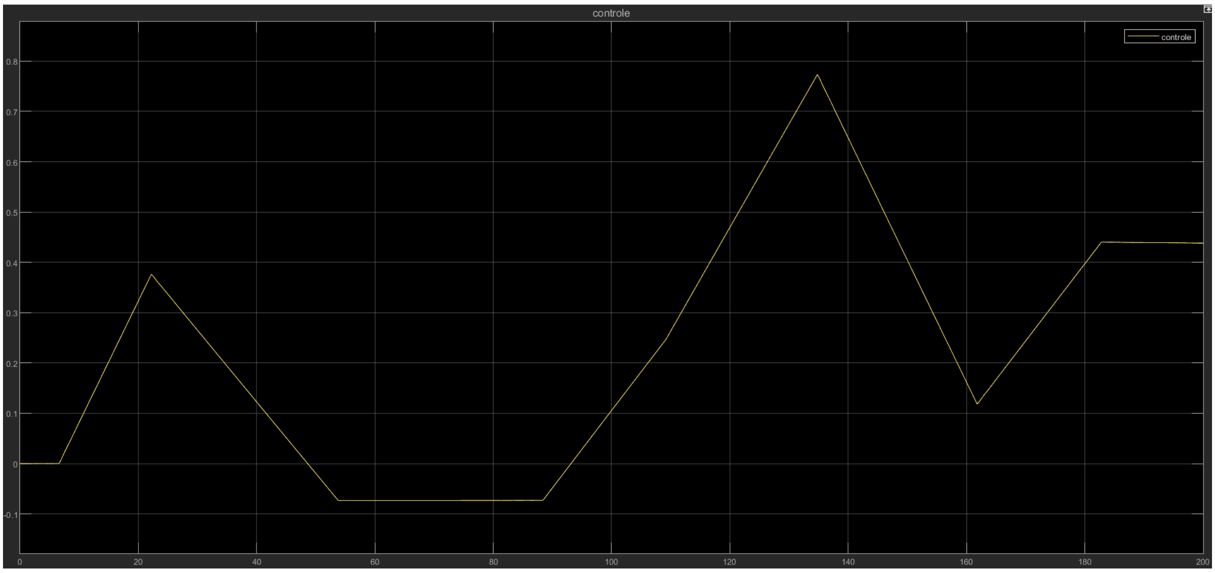

Figura 13 – Sinal de controle.

O sinal de controle apresenta um perfil similar a resposta do angulo, isso se deve ao fato de o objetivo do sistema ser controlar a posição angular da haste, não a deixando cair.

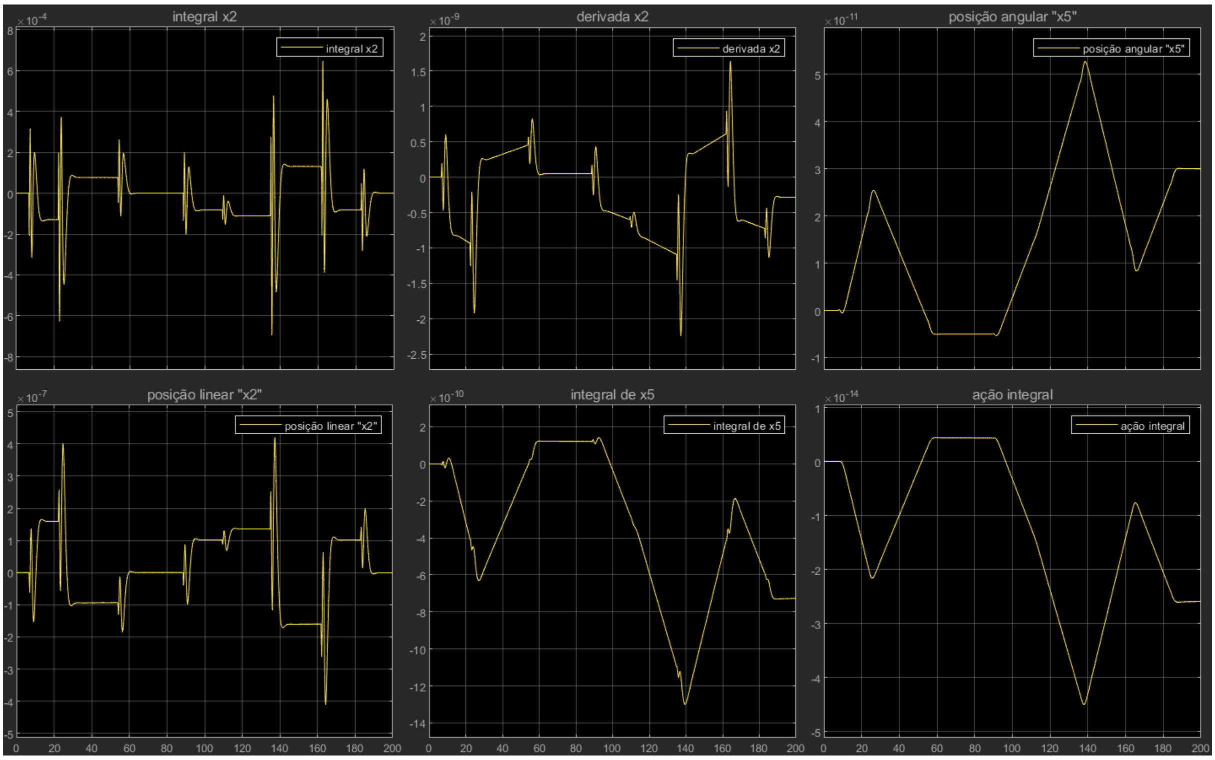

Figura 14 – Estados do sistema.

Para os estados do sistema observa-se uma troca do estado do angulo, sendo apresentado agora no estado x5, provavelmente isso ocorre por conta da troca de base realizada para garantir a controlabilidade.

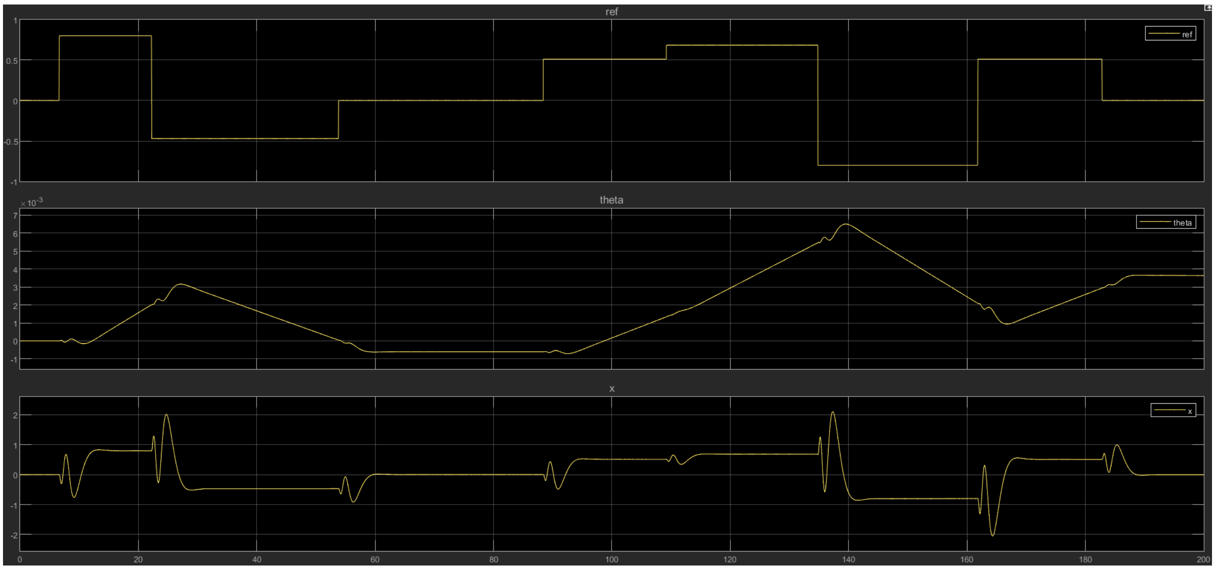

Figura 15 – Saídas do sistema controlado com ação integral, seguindo a referência.

A resposta final do sistema apresenta o erro nulo em regime, seguindo a referência, mas com muito overshoot nos momentos de troca do sinal, sendo necessário tomar cuidado em uma implementação física, isso ocorre principalmente por conta dos altos ganhos do regulador.

4 Conclusões

O sistema de controle desenvolvido conseguiu seguir a referência com o revés de adicionar um overshoot muito alto nas trocas de posição do carrinho, deve se ater a esse fator numa eventual implementação física.

Algumas coisas que poderiam corrigir esse comportamento seriam as alterações dos parâmetros desejados, remodelagem do sistema, linearizando o sistema já na modelagem, à fim de minimizar os ganhos do regulador.

Referências

ESCALANTE, Felix M et al. Robust linear quadratic regulator applied to an inverted pendulum. *Asian Journal of Control*, Wiley Online Library, v. 25, n. 4, p. 2564–2576, 2023. Citado 1 vez na página 5.

GRASSER, Felix et al. JOE: a mobile, inverted pendulum. *IEEE Transactions on industrial electronics*, IEEE, v. 49, n. 1, p. 107–114, 2002. Citado 1 vez na página 5.

MARTINS, João Victor Elias. *Controle Pêndulo Invertido GitHub*. Universidade Estadual Paulista. Nov. 2024. **urdedo**$_i$*ndicadorrom*: `https://github.com/VihEltins/Controle-de-pendulo-invertido`. Acesso em: 27 nov. 2024. Citado 2 vezes nas páginas 7, 19.

MARTINS, João Victor Elias. *Controle Pêndulo Invertido Vídeo*. Universidade Estadual Paulista. Nov. 2024. **urdedo**$_i$*ndicadorrom*: `https://youtu.be/zcNQ4ZTUe0Y`. Acesso em: 27 nov. 2024. Citado 1 vez na página 19.

NG, Wai Man; CHANG, Dong Eui; SONG, Seong-Ho. Four representative applications of the energy shaping method for controlled lagrangian systems. *Journal of Electrical Engineering and Technology*, The Korean Institute of Electrical Engineers, v. 8, n. 6, p. 1579–1589, 2013. Citado 1 vez na página 5.

PATHAK, Kaustubh; FRANCH, Jaume; AGRAWAL, Sunil Kumar. Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *IEEE Transactions on robotics*, IEEE, v. 21, n. 3, p. 505–513, 2005. Citado 1 vez na página 5.

POLO, Manuel F Pérez; MOLINA, Manuel Pérez; CHICA, Javier Gil. Swing-up and positioning control of an inverted wheeled cart pendulum system with chaotic balancing motions. *International Journal of Non-Linear Mechanics*, Elsevier, v. 47, n. 6, p. 655–665, 2012. Citado 1 vez na página 5.

VANONCINI, Michele; HOLDERBAUM, William; ANDREWS, Brian J. Electrical stimulation for trunk control in paraplegia: a feasibility study. *Control Engineering Practice*, Elsevier, v. 20, n. 12, p. 1247–1258, 2012. Citado 1 vez na página 5.

XIANG, Yujiang; ARORA, Jasbir S; ABDEL-MALEK, Karim. Optimization-based prediction of asymmetric human gait. *Journal of biomechanics*, Elsevier, v. 44, n. 4, p. 683–693, 2011. Citado 1 vez na página 5.

YANG, Aolei et al. Stability analysis and implementation of a decentralized formation control strategy for unmanned vehicles. *IEEE Transactions on Control Systems Technology*, IEEE, v. 22, n. 2, p. 706–720, 2013. Citado 1 vez na página 5.