

UNIVERSIDADE ESTADUAL PAULISTA "JULIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA

DIOGO CUNHA JOSÉ KARMOUCHE

CONTROLE COM *MACHINE LEARNING* VIA *PARTICLE*
SWARM OPTIMIZATION

UNIVERSIDADE ESTADUAL PAULISTA "JULIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA

DIOGO CUNHA JOSÉ KARMOUCHE

CONTROLE COM *MACHINE LEARNING* VIA *PARTICLE
SWARM OPTIMIZATION*

Dissertação apresentada ao Programa de
Pós-Graduação em Engenharia Mecânica,
Faculdade de Engenharia de Ilha Solteira -
UNESP como parte dos requisitos necessá-
rios para obtenção do título de Mestre em
Engenharia Mecânica

Orientador: Prof. Dr. Gustavo Luiz Chagas Manhães de Abreu

Ilha Solteira

2022

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

K18c Karmouche, Diogo Cunha José.
Controle com *Machine Learning* via particle swarm optimization / Diogo
Cunha José Karmouche. -- Ilha Solteira: [s.n.], 2022
97 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de
Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2022

Orientador: Gustavo Luiz Chagas Manhães de Abreu
Inclui bibliografia

1. Aprendizado de máquina. 2. Enxame de partículas. 3. Pêndulo invertido.
4. Controle adaptativo.


Raiane da Silva Santos

Supervisora Técnica de Seção
Seção Técnica de Referência, Atendimento ao usuário e Documentação
Diretoria Técnica de Biblioteca e Documentação
CRB/8 - 9999

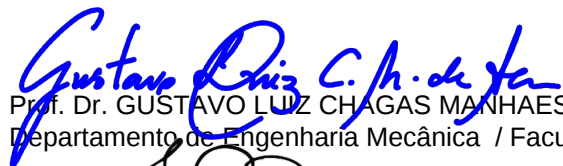
CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Controle com Machine Learning via Particle Swarm Optimzation

AUTOR: DIOGO CUNHA JOSÉ KARMOUCHE

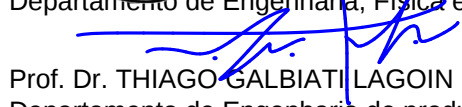
ORIENTADOR: GUSTAVO LUIZ CHAGAS MANHAES DE ABREU

Aprovado como parte das exigências para obtenção do Título de Mestre em ENGENHARIA MECÂNICA, área: Mecânica dos Sólidos pela Comissão Examinadora:



Prof. Dr. GUSTAVO LUIZ CHAGAS MANHAES DE ABREU (Participação Virtual)
Departamento de Engenharia Mecânica / Faculdade de Engenharia de Ilha Solteira - UNESP


Prof. Dr. FABIO ROBERTO CHAVARETTE (Participação Virtual)
Departamento de Engenharia, Física e Matemática / Instituto de Química de Araraquara - UNESP


Prof. Dr. THIAGO GALBIATI LAGOIN (Participação Virtual)
Departamento de Engenharia de produção / Universidade Federal de Mato Grosso do Sul - UFMS

Ilha Solteira, 10 de março de 2022

Dedico este trabalho ao meu pai José por me apoiar no caminho que decidi trilhar.

AGRADECIMENTOS

Gostaria de começar agradecendo a Deus pela oportunidades que me foram fornecidas e também por me dar força e sabedoria nos momentos que mais precisei.

Ao professor Sanderson Manoel da Conceição por acreditar em mim e me indicar ao professor Gustavo Luiz Chagas Manhães de Abreu, que também agradeço por toda paciência, disposição, atenção e por tornar este trabalho possível. Também gostaria de agradecer a todos os professores que me ajudaram em minha jornada por meio de aulas, reuniões e dicas para contribuir ao meu trabalho.

Gostaria também de agradecer a toda minha família, em especial minha mãe Sandra Lucia Brandão Karmouche e meu pai José Roberto Jorge Karmouche por toda força, apoio e paciência quando mais precisei. Aos meus irmãos que sempre estiveram comigo, mesmo com as barreiras de distância entre as cidades.

Agradeço a minha namorada Ana Paula Fleitas Sachelaride pelo com amor, carinho e incentivo que também se manteve firme com as barreiras de distância entre as cidades.

Aos meus amigos que tenho e fiz durante o percurso da minha vida, que mesmo distantes me deram apoio.

Por último, gostaria de agradecer a Universidade Estadual Paulista - UNESP (Ilha Solteira) e todos os seus membros novamente por possibilitarem a execução deste trabalho.

*"Nunca perfeito. Objetivo da perfeição muda. Nunca para de se mover. Pode perseguir, não
pode alcançar"*

— Abathur - StarCraft II

RESUMO

Aprendizado de máquina é uma área em grande desenvolvimento e promissora, sendo utilizada na resolução de problemas com o mínimo ou nenhuma interferência humana. Em teoria de controle diferentes abordagens para elaboração de leis de controle são utilizadas, com aprendizado de máquina tais abordagens podem ser divididas em, identificação de parâmetros do sistema para aplicar técnicas de controle convencionais e elaborar diretamente a estrutura e parâmetros da lei de controle. Neste contexto, a elaboração direta da estrutura de controle permite uma abordagem sem conhecimento prévio do sistema em questão. Assim, este trabalho visa apresentar o controle de um sistema não linear como pêndulo invertido utilizando aprendizado de máquina por meio do algoritmo otimização por enxame de partículas, onde os parâmetros do sistema, como massa do carro, comprimento da haste do pêndulo e os coeficientes de atrito, serão alterados. Para isso, foram utilizadas 5 populações de vetores de ganho, que foram geradas de maneiras diferentes, uma de maneira aleatória, duas pelo regulador quadrático linear e duas pelo regulador quadrático linear descrito por desigualdade de matriz linear. Deste modo, foi visto como que o aprendizado de máquina utilizando o enxame de partículas gera vetores de controle para controlar um sistema que sofrerá alterações com o decorrer das iterações.

Palavras-chave: Aprendizado de máquina; enxame de partículas; pêndulo invertido; controle adaptativo;

ABSTRACT

Machine learning is an area of great development and promise, being used to solve problems with minimal human interference. In theory, the applied control approaches for the construction of control laws are used, with learning methods of construction methods of parameters can be defined as methods of construction of control methods and, directly, to a structure of parameters of elaboration of parameters. In this context, the direct elaboration of the control structure allows an approach without prior knowledge of the system in question. Thus, this work aims to present the control of a system shed as a pendulum used linearly to use the machine through the use of the particle method, where the system parameters, such as car mass, pendulum rush length and the methods of friction calculation, will be changed. For this, 5 different gain alternatives were used, which were created in the same way, a random way, two by the linear quadratic regulator and two by the linear matrix regulator described by linear. In this way, it was seen as the prediction of a machine using or swarming control vectors for a system that will change over the course of the iterations.

Keywords: Machine learning; particle swarming; inverted pendulum; adaptive control;

LISTA DE FIGURAS

Figura 1 - Design do controlador PID baseado em PSO	18
Figura 2 - Design do controlador PID baseado em PSO	20
Figura 3 - <i>Overfitting</i>	23
Figura 4 - Agrupamento de objetos	24
Figura 5 - Controle com <i>Machine Learning</i>	26
Figura 6 - Diagrama do Algoritmo PSO	29
Figura 7 - Decisão vetorial para novo movimento da partícula	31
Figura 8 - Esquema de Controle via PSO	32
Figura 9 - Diagrama Modificado do Algoritmo PSO	33
Figura 10 - Sistema mecânico pêndulo invertido	34
Figura 11 - Diagrama elétrico do motor de corrente contínua	35
Figura 12 - Índice ISE para $\mathbf{Q}_1, \mathbf{R}_1$, $\mathbf{Q}_2, \mathbf{R}_1$, $\mathbf{Q}_3, \mathbf{R}_1$ e $\mathbf{Q}_3, \mathbf{R}_2$	45
Figura 13 - Índice CE para $\mathbf{Q}_1, \mathbf{R}_1$, $\mathbf{Q}_2, \mathbf{R}_1$, $\mathbf{Q}_3, \mathbf{R}_1$ e $\mathbf{Q}_3, \mathbf{R}_2$	45
Figura 14 - Sistema Pêndulo invertido focando a massa do carro	47
Figura 15 - Sistema Pêndulo invertido focando nos coeficientes de atrito	48
Figura 16 - Simulação das velocidades do Sistema pêndulo invertido sem controle	50
Figura 17 - População A	51
Figura 18 - Combinação de Todos os Pontos de Operação	52
Figura 19 - Possíveis populações B	53
Figura 20 - Populações B e C	54
Figura 21 - Populações D e E	57
Figura 22 - Progressão das posições da população A	60
Figura 23 - Custo das melhores posições da população A para todos os casos de mudanças de parâmetros	61
Figura 24 - CE e ISE das melhores posições da população A	61
Figura 25 - Situação sem mudanças das melhores posições da população A	62
Figura 26 - Progressão das posições da população B	65
Figura 27 - Análise das melhores posições da população B	66
Figura 28 - Situação sem mudanças das melhores posições da população B	67
Figura 29 - Progressão das posições da população C	70
Figura 30 - Análise das melhores posições da população C	71

Figura 31 - Situação sem mudanças das melhores posições da população C . . .	72
Figura 32 - Progressão das posições da população D	75
Figura 33 - Análise das melhores posições da população D	76
Figura 34 - Situação sem mudanças das melhores posições da população D . .	77
Figura 35 - Progressão das posições da população E	80
Figura 36 - Análise das melhores posições da população E	81
Figura 37 - Situação sem mudanças das melhores posições da população E . .	82
Figura 38 - Progressão das populações na mudança de massa	84
Figura 39 - Progressão das populações na mudança de comprimento	86
Figura 40 - Progressão das populações na mudança de atrito	88
Figura 41 - Progressão do custo das partículas	90
Figura 42 - Custo das melhores posições para situação sem mudança	91

LISTA DE TABELAS

Tabela 1 -	Hiperparâmetros do Algoritmo PSO	30
Tabela 2 -	Dados do Sistema Pêndulo Invertido	43
Tabela 3 -	Dados do Algoritmo PSO	46
Tabela 4 -	Tabela de Máximos e Mínimos dos Pontos de Operação	49
Tabela 5 -	Dados do Sistema para População C	52
Tabela 6 -	Tabela do Desvio Padrão ao quadrado das Possíveis Populações B	54
Tabela 7 -	Tabela das Melhores Posições da População A - Mudança na Massa	59
Tabela 8 -	Tabela das Melhores Posições da População A - Mudança na Comprimento	59
Tabela 9 -	Tabela das Melhores Posições da População A - Mudança na Atrito	59
Tabela 10 -	Tabela das Melhores Posições da População B - Mudança na Massa	63
Tabela 11 -	Tabela das Melhores Posições da População B - Mudança na Comprimento	64
Tabela 12 -	Tabela das Melhores Posições da População B - Mudança na Atrito	64
Tabela 13 -	Tabela das Melhores Posições da População C - Mudança na Massa	68
Tabela 14 -	Tabela das Melhores Posições da População C - Mudança na Comprimento	68
Tabela 15 -	Tabela das Melhores Posições da População C - Mudança no Atrito	69
Tabela 16 -	Tabela das Melhores Posições da População D - Mudança na Massa	73
Tabela 17 -	Tabela das Melhores Posições da População D - Mudança na Comprimento	73
Tabela 18 -	Tabela das Melhores Posições da População D - Mudança no Atrito	74

Tabela 19 -	Tabela das Melhores Posições da População E - Mudança na Massa	78
Tabela 20 -	Tabela das Melhores Posições da População E - Mudança na Comprimento	78
Tabela 21 -	Tabela das Melhores Posições da População E - Mudança no Atrito	79

SUMÁRIO

	Página
1	INTRODUÇÃO
1.1	Objetivos 21
2	<i>MACHINE LEARNING</i>
2.1	Aprendizado Supervisionado 23
2.2	Aprendizado Não Supervisionado 24
2.3	Aprendizado Reforçado 24
2.4	Controle com <i>Machine Learning</i> 25
3	<i>SWARM INTELLIGENCE</i>
3.1	<i>Particle Swarm Optimization</i> 28
3.2	Controle PSO 31
4	MODELAGEM
4.1	Motor 34
4.2	Carro 36
4.3	Pêndulo 37
4.4	Equações Não Lineares 38
4.5	Linearização 38
5	TÉCNICAS DE CONTROLE
5.1	Controle Ótimo 40
5.2	Robusto Ótimo 41
6	<i>METODOLOGIA</i>
6.1	Sistema 43
6.2	Função de Custo 44
6.3	Parâmetros do Algoritmo 45
6.4	Populações 49
6.4.1	População A 50
6.4.2	Populações B e C 51
6.4.3	População D e E 55

7	RESULTADOS	58
7.1	População A	58
7.2	População B	63
7.3	População C	67
7.4	População D	73
7.5	População E	78
7.6	Comparação Populacional	83
8	CONSIDERAÇÕES FINAIS	92
	Apêndice A – PROGRAMAÇÃO	97

1 INTRODUÇÃO

O desenvolvimento de leis de controle para sistemas dinâmicos sempre foi um grande desafio devido a incertezas e situações inesperadas no ambiente ou no próprio sistema. Por isso, diferentes técnicas e abordagens de controle estão sendo pesquisadas. Na atualidade o *machine learning* tem sido utilizado para diferentes problemas obtendo soluções favoráveis em diferentes áreas como financeira, médica, controle de sistemas e dentre muitas outras (DURIEZ et al., 2017; LI et al., 2018).

Aprendizado de Máquina ou *Machine Learning* é uma área em rápido desenvolvimento com grande impacto em problemas de engenharia e ciências naturais que utiliza estatística, ciência da computação e matemática aplicada para trabalhar os dados de modo que haja o mínimo ou nenhuma interferência humana sem ser expressamente programado para que possa chegar a resultados sem ser explicitamente programado para tal, como gerar modelos de predição ou classificação a partir de uma base de dados. (BONACCORSO, 2017; DURIEZ et al., 2017; PHIL, 2017; LI et al., 2018). Para realizar tal tarefa são utilizadas abordagens de aprendizado que se diferenciam como são tratados os dados de "input/entrada" no algoritmo utilizado e como os dados de treinamento são abordados. Popularmente essas abordagens de aprendizado são categorizadas em aprendizado supervisionado, não supervisionado e aprendizado reforçado. Na abordagem supervisionada os dados de treinamento são devidamente rotulados, deste modo o modelo é adaptado aos dados que são inseridos no algoritmo, na abordagem não supervisionada os dados de treinamento não possuem um rótulo e os dados de entrada podem ser utilizados para investigação de características do modelo. E na abordagem reforçada os dados de entrada são os dados de treinamento do modelo (PHIL, 2017; LI et al., 2018). Junto a essas abordagens são utilizadas diferentes técnicas para o treinamento como redes neurais, árvores de decisões, análise de regressão, redes bayesianas e algoritmos evolutivos (DURIEZ et al., 2017).

Em teoria de controle o aprendizado de máquina é utilizado como uma técnica de otimização permitindo novas abordagens na elaboração de leis de controle. Como o desenvolvimento de um modelo matemático que represente o modelo real, para então aplicar as técnicas de controle conhecidas. Além disso, o aprendizado de máquina é utilizado no mapeamento de entradas e saídas do sistema para desenvolver diretamente a lei de controle

(DURIEZ et al., 2017).

Nas últimas décadas algoritmos metaheurísticos, como *swarm intelligence* e algoritmo evolutivo tornaram-se populares métodos de otimização (LIN, 2006; JANPROM et al., 2020), inclusive para técnicas de aprendizado de máquina, devido a sua capacidade de solução inspirada em eventos e comportamentos naturais. Na área da ciência da computação, algoritmos metaheurísticos são métodos de otimização que podem possuir caráter estocástico e podem produzir resultados suficientemente bons para problemas, mas não garantem resultados ótimos globais, logo, seus resultados podem variar para cada problema e condição.

Dentre os algoritmos metaheurísticos, especificamente os algoritmos evolutivos, destaca-se o *swarm intelligence* (SI), que estuda o comportamento social e as características de organização descentralizadas e complexas de conjuntos determinados como “enxames” (“*swarm*”) (KIRANYAZ et al., 2014). De acordo com Parsopoulos e Vrahatis (2010) algoritmos baseados em “enxames” tem apresentado uma grande abrangência de aplicações em diferentes áreas e soluções de problemas complexos com robustez, eficiência e com uma implementação simplificada.

Dentre os diversos algoritmos baseados em “enxame” o algoritmo *particle swarm optimization* (PSO) é destacado, pois desde de sua criação em 1995 houve um crescimento exponencial no número de publicações e aplicações em diversas áreas (PARSOPOULOS; VRAHATIS, 2010), tornando-o um algoritmo bem conhecido. Na área de controle o algoritmo PSO é utilizado como um otimizador de vetor de ganhos arbitrários que minimizam uma função de custo que pode considerar atributos de performance como *rise time*, tempo de acomodação, *overshoot*, *steady estate error* e consumo energético, e também pode ser utilizado como um identificador de parâmetros do sistema a ser controlado, para então aplicar técnicas de controle conhecidas (LATHA et al., 2013; ROZALI et al., 2014; OLIVEIRA et al., 2020) ou na identificação de parâmetros de equações não lineares para elaboração de leis de controle.

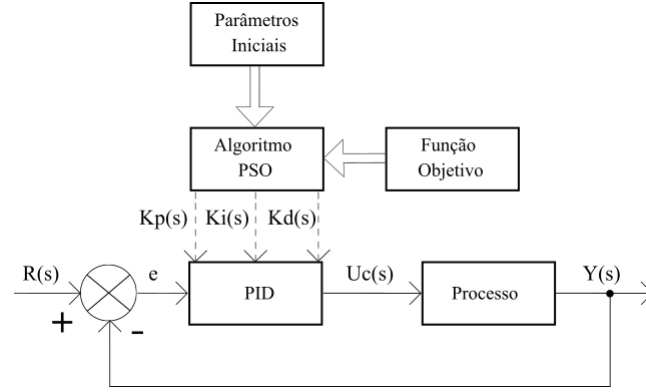
Em Latha et al. (2013) é descrito que o controle convencional PID é muito dependente de um modelo, requerendo a redução da ordem dos modelos para primeira ordem ou de segunda ordem com *delay*. Para isso é proposto um ajuste nos parâmetros do controlador PID através do algortimo PSO para diferentes sistemas, apontando que sua principal vantagem é falta de necessidade de um modelo matemático apenas sendo guiado por uma função objetivo, que neste trabalho foi utilizado a equação 1, que considera três parâmetros de performance como ISE - *integral squared error*, pico de *overshoot* e tempo

de acomodação:

$$J(\theta) = W_1 ISE + W_2 M_p + W_3 t_s \quad (1)$$

Para atualização do controle PID via PSO Latha et al. (2013) utilizou o diagrama de blocos ilustrado na Figura 1:

Figura 1- Design do controlador PID baseado em PSO



Fonte: Adaptado de Latha et al. (2013)

Para o primeiro sistema, representado pela equação 2, foi apenas analisado o ajuste de hiperparâmetros para diferentes fatores de pesos do algoritmo PSO ($W = 0.5$, $W = 0.75$ e $W = 1$) e foi observado que mesmo que o peso $W = 1$ tenha tomado maior tempo para convergência foi o que apresentou melhor resultado quando comparado aos demais pesos.

$$G(s) = \frac{1}{s+1} e^{-0.5s} \quad (2)$$

Para o segundo sistema, representando pela equação 3, foi proposto diferentes algoritmos evolutivos, como otimização por colônia de formigas (ACO) e algoritmo genético (GA), e também a técnica de controle clássica Ziegler-Nichols para o ajuste de parâmetros do ganho PID e comparar com o controle PSO.

$$G(s) = \frac{1}{(s+1)^2} e^{-0.5s} \quad (3)$$

O segundo sistema Latha et al. (2013) apontou que a resposta dos parâmetros ajustados pelo algoritmo PSO obteve melhor desempenho em *overshoot* e uma resposta mais suave quando comparado as outras técnicas. Para o terceiro sistema, descrito pela equação 4 que representa um modelo estável de um bioreator, foi proposto o algoritmo PSO para multiobjetos, o algoritmo genético e a técnica Ziegler-Nichols para ajustar os parâmetros

do ganho PID e analisar sua resposta.

$$G(s) = \frac{-1.53s - 0.4588}{s^2 + 2.564s + 0.6792} e^{-0.1s} \quad (4)$$

É apontado por Latha et al. (2013) que para este sistema os métodos propostos providenciaram uma rejeição de distúrbio e uma *reference tracking* melhor quando comparado aos demais métodos nos outros sistemas e o algoritmo PSO obteve o menor valor ISE quando comparado ao Ziegler-Nichols e o algoritmo genético. Para o quarto sistema analisado é utilizado o modelo instável de um bioreator, representado pela equação 5, por ser considerado um problema *bench mark* para o design de controladores e é comparado o ajuste de parâmetros via PSO com o algoritmo de otimização de forrageamento bacteriano (BFO).

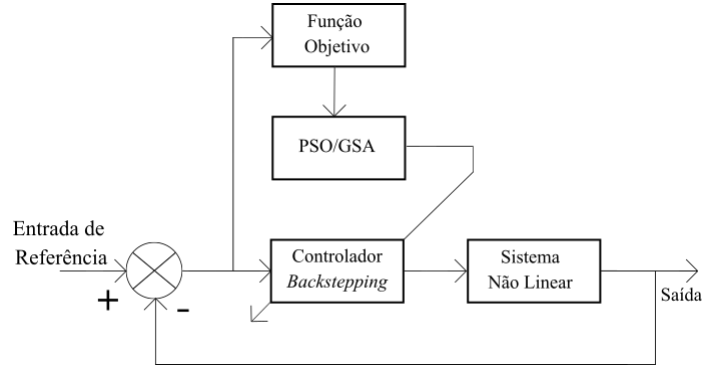
$$G(s) = \frac{-5.8644}{5.89s - 1} e^{-0.1s} \quad (5)$$

De acordo com Latha et al. (2013) o algoritmo PSO destacou-se em relação ao BFO na otimização de valores. Para o último sistema analisado foi modelado um tanque esférico não linear, representado pela equação 6:

$$G(s) = \frac{3.6215}{330.46s + 1} e^{-11.7s} \quad (6)$$

Ao final da simulação desse sistema Latha et al. (2013) concluiu que o ajuste de parâmetros pelo algoritmo PSO possui uma boa adaptabilidade para diferentes sistemas utilizando a mesma função objetivo, uma boa *reference tracking* e rejeição de distúrbios.

Em Rozali et al. (2014) foi realizado o controle *backstepping* de um sistema não linear com parâmetros ajustados por dois algoritmos o PSO e o *Gravitational Search Algorithm* (GSA). De acordo com os autores, o controle *backstepping* foi escolhido pois permite obter estabilidade global em casos quando o método de linearização em *feedback* possibilita estabilidade local. A aplicação desse controle varia desde controle de trajetória de vôo, sistema automáticos, máquinas elétricas, e etc. No algoritmo GSA, são considerado agentes que interagem baseados nas leis de atração Newtonianas da gravidade e sua performance são medidas através de massas. Para ambos os algoritmos desse trabalho o diagrama de blocos do ajuste de controle é ilustrado pela Figura 2:

Figura 2- Design do controlador PID baseado em PSO

Fonte: Adaptado de Rozali et al. (2014)

Assim, no trabalho de Rozali et al. (2014) o sistema escolhido para elaboração dos parâmetros de controle foi um atuador eletrohidráulico, regido pelo modelo em espaço de estados na equação 7:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= -\frac{k}{m}x_1 - \frac{f}{m}x_2 + \frac{S}{m}x_3 - \frac{F_L}{m} \\
 \dot{x}_3 &= -\frac{S}{k_c}x_2 - \frac{k_l}{k_c} + \frac{c}{k_c}\sqrt{\frac{p_a - x_3}{2}}k_v u
 \end{aligned} \tag{7}$$

Rozali et al. (2014) observou na simulação dos dois algoritmos que o ganho parametrizado pelo PSO obteve melhor desempenho quando comparado ao algoritmo GSA com uma resposta mais suave e com menor *tracking error*.

No trabalho de Oliveira et al. (2020) foram utilizadas as técnicas Cohen-Coon e Amigo para determinar limites de exploração para o algoritmo PSO elaborar os parâmetros de controle PID para um sistema de controle de temperatura e comparar os resultados com outro algoritmo de otimização lobo cinzento ou “*Grey Wolf Optimization*” (GWO). O sistema desse trabalho consiste em um *kit* de Arduino possuindo dois sensores de temperatura, dois transistores atuando como aquecedores em uma placa de circuito impresso e um Arduino. Os dois algoritmos utilizam o método de otimização por enxame, mas se diferenciam em como trabalham seus melhores valores da função de custo. Sendo assim, nesse trabalho foi observado que o desempenho do controle desenvolvido por PSO é similar ao GWO. O algoritmo PSO em relação as técnicas Cohen-Coon e Amigo apresenta uma variação menos agressiva e menos irregular no sinal de controle. O que verifica a boa competência do algoritmo PSO ao lidar com vários critérios para o design do controlador e a independência do mesmo em relação a dinâmica do sistema ao desenvolver as leis de controle.

1.1 Objetivos

O objetivo deste trabalho é realizar o controle com *machine learning* via o algoritmo PSO para um sistema instável e não linear utilizando cinco diferentes populações iniciais de ganhos de controle e ao decorrer das iterações o sistema a ser controlado terá seus parâmetros físicos alterados. Algumas populações utilizaram o sistema em sua forma linear e serão geradas com leis de controle conhecidas e as demais populações serão geradas para sistema com parâmetros diferentes do sistema a ser controlado, ademais, as mudanças aplicadas durante as iterações impõem novas situações para o algoritmo, onde as populações de ganhos não foram elaboradas para abranger. Assim, com as diferentes populações e as mudanças do sistema inicial, será possível avaliar qual população proporciona regiões nos espaços de busca com custo mínimo e ao mesmo tempo robustos. Esta técnica de controle com *machine learning* apresenta uma alternativa na elaboração e refinamento de leis de controle quando não há muito conhecimento específico sobre o sistema.

2 MACHINE LEARNING

Atualmente, o uso de máquinas inteligentes tem se tornado rotineiro, sendo difícil não ter tido contato com algum sistema que utilize um sistema artificial inteligente para realizar alguma tarefa. Um exemplo de um sistema inteligente é a capacidade do *smartphone* reconhecer o som específico da voz do usuários ou o sistema de propagandas que a partir do histórico de pesquisa do usuário apresenta produtos que provavelmente chamariam atenção do mesmo (MUELLER; MASSARON, 2021). Tais sistemas inteligentes fazem parte de subdivisões da inteligência artificial.

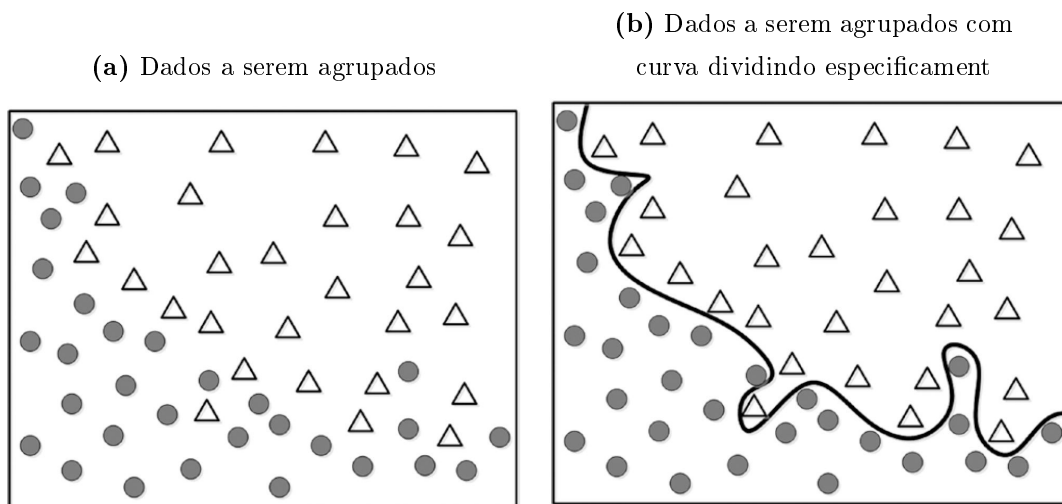
Acredita-se que o nascimento do ramo da inteligência artificial, como conhecida atualmente, começou com a publicação “*Computing Machinery and Intelligence*” de Alan Turing em 1950, onde é explorado se a máquina é capaz ou não de pensar (MUELLER; MASSARON, 2021), e que com o decorrer do tempo e desenvolvimento da tecnologia o ramo da inteligência artificial foi se subdividindo, onde uma dessas subdivisões é o *machine learning* (LANGLEY; LAIRD, 2006). Aprendizado de máquina ou *machine learning* é uma tecnologia onde a máquina tem a capacidade de “aprender” conforme dados são fornecidos e gerar um modelo por meio de um processo de “aprendizagem” (PHIL, 2017). De acordo com Bonaccorso (2017) o processo de “aprendizagem” é a habilidade de mudança de acordo com estímulos externos e mantendo a lembrança de experiências passadas. Sendo assim, o objetivo do *machine learning* pode ser definido como o estudar, desenvolver e/ou melhorar modelos matemáticos por meio do treinamento com dados para inferir previsões e tomar decisões sem ser explicitamente programado e com o mínimo ou sem interferência humana (BONACCORSO, 2017; DURIEZ et al., 2017; PHIL, 2017; LI et al., 2018).

Diferentes abordagens de *machine learning* foram desenvolvidas para resolver diferentes problemas em diferentes áreas, como financeira, climática, robótica e entre outras (BONACCORSO, 2017; DURIEZ et al., 2017). Tais abordagens podem ser classificadas de acordo o método de treinamento, sendo divididas em supervisionado, não supervisionado e aprendizado reforçado.

2.1 Aprendizado Supervisionado

Na abordagem supervisionada é definido no algoritmo dados de treinamento nos quais o modelo desenvolvido pelo algoritmo deve minimizar a discrepância entre o valor dos dados do modelo e os dados de treinamento (BONACCORSO, 2017). Contudo, o modelo desenvolvido é requerido trabalhar também com dados nunca vistos antes pelo algoritmo, sendo assim, é necessário que o algoritmo possua uma habilidade de generalização para evitar um problema comum em algoritmos de *machine learning* chamado de *overfitting*, que consiste no modelo desenvolvido ser muito específico para os dados de treinamento e não poder ser usado para outros dados (BONACCORSO, 2017; PHIL, 2017). Em Phil (2017) é apresentado um exemplo de *overfitting* por meio da classificação de dois grupos, como apresentado na Figura 3a, assim o objetivo é criar uma curva que divide os dois grupos e na Figura 3b é mostrado uma curva que separa muito bem os dois grupos, mas caso inserisse mais pontos além dos pontos usados para traçar a curva, a mesma poderia não ser mais apropriada para o bjetivo de classificar.

Figura 3- *Overfitting*



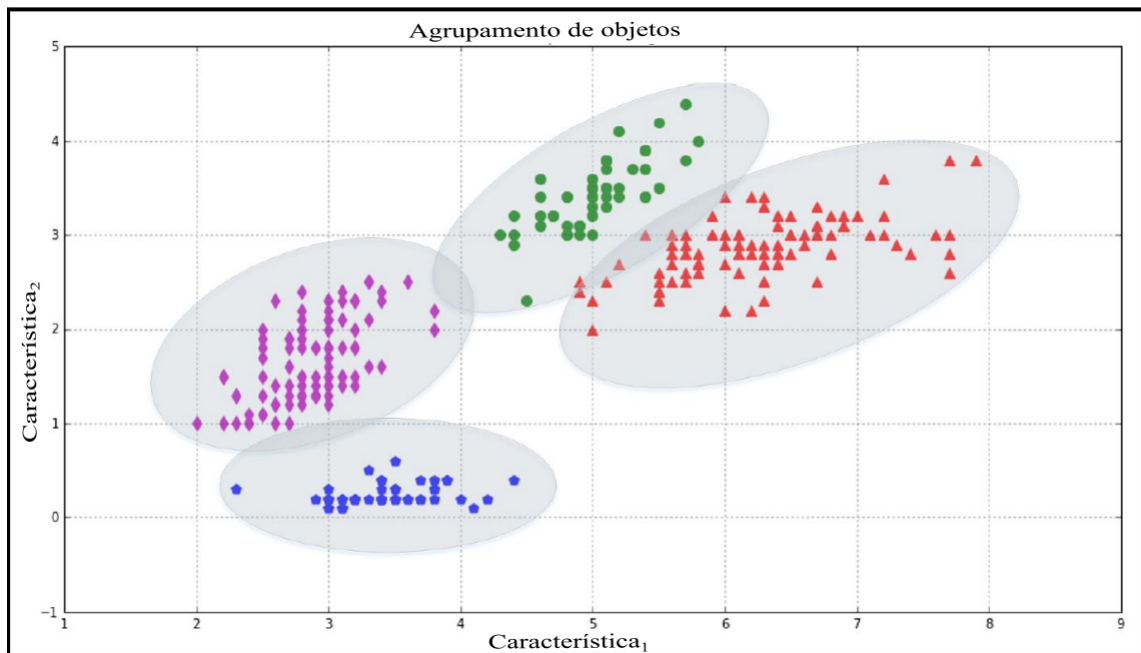
Fonte: Adaptado de Phil (2017)

As aplicações desta aprendizagem variam desde detecção de spam na caixa de email até detecção de falhas em equipamentos mecânicos possibilitando a previsão de probabilidade de falha do equipamento.

2.2 Aprendizado Não Supervisionado

Na abordagem não supervisionada não há medidas de erro, ou seja, não há referência para o algoritmo determinar se a resposta do modelo está correta (BONACCORSO, 2017; PHIL, 2017). Esta abordagem é utilizada para realizar agrupamentos de acordo com as similaridades entre os dados fornecidos ao algoritmo. Um exemplo é apresentado na Figura 4, onde são colocados dados em um espaço onde os dados são posicionados de acordo com a $caracteristica_1$ e $caracteristica_2$, assim, o algoritmo não supervisionado cria agrupamentos para os dados por meio de similaridades entre eles, representados por elipses na Figura 4 (BONACCORSO, 2017).

Figura 4- Agrupamento de objetos



Fonte: Adaptado de Bonaccorso (2017)

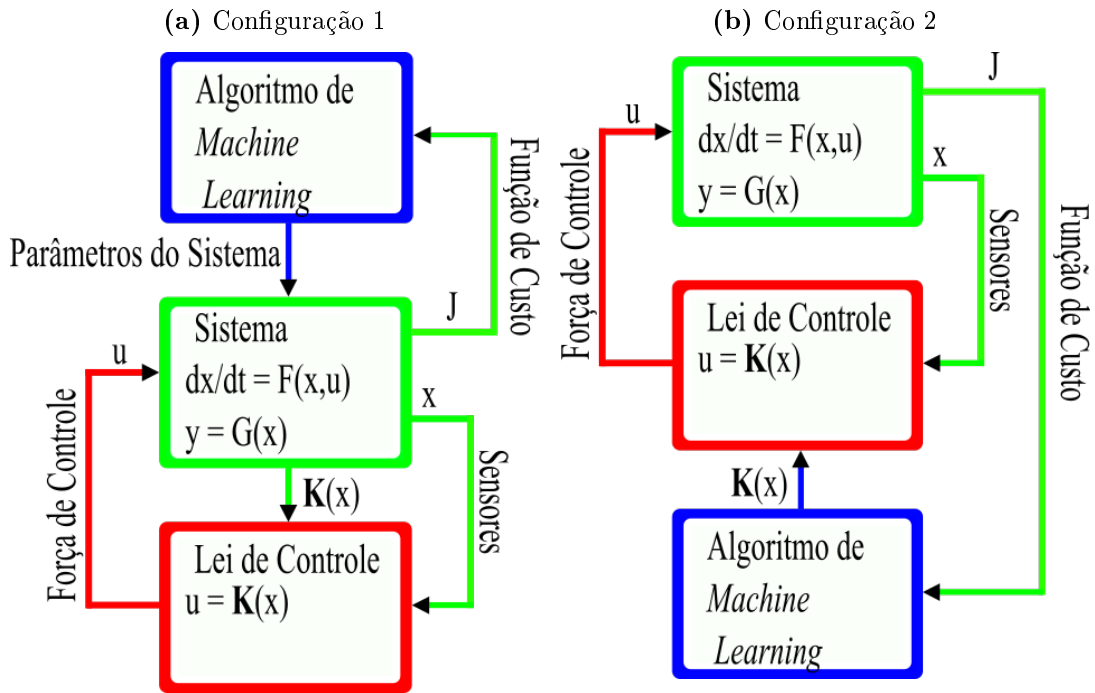
2.3 Aprendizado Reforçado

Na abordagem de aprendizado reforçado os dados inseridos no algoritmo são considerados os dados de treinamento dando recompensas positivas ou negativas da resposta do modelo gerado. Deste modo, o algoritmo busca maximizar a nota produzindo resultados mais qualitativos, mas sem uma noção de erro precisa (BONACCORSO, 2017; PHIL, 2017). De acordo com Bonaccorso (2017) esta técnica é aplicada eficientemente em ambientes onde é impossível obter uma mensuração precisa do erro, devido o ambiente onde o modelo é aplicado ser muito dinâmico.

2.4 Controle com *Machine Learning*

Ao pesquisar sobre *machine learning* é visto que sua aplicabilidade abrange diferentes áreas como na área financeira realizando previsões no mercado de ações (BONACCORSO, 2017), na área comercial auxila o *e-commerce* a obter vantagens competitivas e instalar uma presença digital (THOBANI, 2018), na área médica desenvolvendo modelos para o diagnóstico de câncer (SIDEY-GIBBONS; SIDEY-GIBBONS, 2019) e na área de sistema de controle identificando parâmetros ou modelando o próprio vetor de controle (DURIEZ et al., 2017; WU et al., 2018)

Em Duriez et al. (2017) é apresentado a utilização de *machine learning* para elaboração de leis de controle as quais são divididas em duas configurações representados na Figura 5, onde $\dot{x} = F(x, u)$ é um sistema arbitrário, com uma força de controle, u , e estados, x , onde uma outra função, $y = G(x)$, representa os estados adquiridos por sensores e também apresenta uma lei de controle representada pela função $u = K(x)$ e que por fim é avaliada por uma função de custo, J . Assim, a primeira configuração, representada na Figura 5a, utiliza do *machine learning* para gerar um modelo que se aproxima da dinâmica das entradas e das respostas do sistema real a ser controlado, para então aplicar técnicas de controle clássicas ou modernas, como PID e LQR, para gerar o vetor de controle (DURIEZ et al., 2017). Na segunda configuração, representado na Figura 5b, o *machine learning* gera o modelo do vetor de controle, o que não necessita de qualquer conhecimento sobre a dinâmica do sistema a ser aplicado o vetor de controle. Esta configuração possui algumas vantagens sobre a primeira configuração, como a falta de necessidade do modelo dinâmico do sistema real, pois, alguns sistemas possuem uma dinâmica muito complexa e que pode sofrer alterações no decorrer do tempo tornando muito trabalhoso a identificação de um modelo para então aplicar uma técnica de controle. Outra vantagem é que em muitos casos um vetor de controle é robusto o suficiente para controlar diferentes situações do modelo real (DURIEZ et al., 2017).

Figura 5- Controle com *Machine Learning*

Fonte: Autor Próprio

As configurações apresentadas na Figura 5 abordam dois métodos como o *machine learning* pode ser utilizado na elaboração de leis de controles efetivas considerando as limitações do problema para o qual será aplicado o controle, tais configurações utilizam de aprendizados supervisionados ou não supervisionados para configurar o modelo do controlador. Com o aprendizado reforçado, o problema de controle é formulado como um processo de decisão de Markov, que considera os estados do sistema, uma força de controle, uma recompensa e controlador, chamado usualmente de agente, que toma decisões de acordo com uma política, Π (BUŞONIU et al., 2018). No controle com aprendizado reforçado o controlador toma decisões de modo estatístico, tendendo para direção que gerará mais recompensas, de forma genérica a lei de controle é formulada $u = \Pi(x)$, onde a cada simulação ou teste o algoritmo de aprendizado do aprendizado reforçado irá determinar uma recompensa, onde na próxima simulação ou teste a lei de controle irá pender para direção que resulta em uma recompensa maior (BUŞONIU et al., 2018).

3 *SWARM INTELLIGENCE*

Durante milhões de anos sistemas biológicos solucionaram problemas por meio do compartilhamento de informações entre os membros do grupo, analisando o comportamento individual de cada membro e combinando com o comportamento do grupo, desenvolvendo uma ferramenta que simula o comportamento coletivo na resolução de problemas de sistemas biológicos (BLUM; MERKLE, 2008). Tal ferramenta é o *swarm intelligence*, descrita como o comportamento coletivo inteligente de sistemas alto-organizadores e descentralizados artificiais (WAHAB et al., 2015), inspirados em comportamentos biológicos de grupos de animais e insetos (ABRAHAM et al., 2006), que por meio da interação entre os membros do grupo de forma direta e/ou indireta é realizado um “*feedback*” sobre as decisões tomadas, realizando um movimento coordenado (BLUM; MERKLE, 2008). Devido o *swarm intelligence* abranger diferentes grupos biológicos vários tipos de algoritmos foram construídos. Os algoritmos são diferenciados por qual animal está imitando, ou seja, como são tratados os melhores resultados encontrados, a dinâmica dos membros no grupo, a quantidade de hiperparâmetros a serem ajustados, e entre muitas outras diferenças. Alguns exemplos de algoritmos de *swarm intelligence* são apresentados a seguir.

Introduzido em 1992 o algoritmo otimização por colônia de formigas (ACO) imita a comunicação realizada por formigas que baseia-se na comunicação indireta medida por feromônios, deste modo, quanto mais química uma trilha possuir mais formigas serão atraídas para a mesma (SERAPIÃO, 2009). Inicialmente, as formigas são colocadas no espaço de busca para procurar de forma aleatória a “comida”, que representa o resultado do problema, após encontrar a “comida”, a formiga retorna ao formigueiro marcando o caminho com feromônios para atrair as demais formigas, mesmo assim, algumas formigas irão continuar explorando diferentes caminhos e caso encontrem um melhor atrairá mais formigas, ao final da simulação o caminho com mais formigas é o que obteve a melhor resposta (DORIGO; BLUM, 2005). Sua aplicação engloba desde ajuste de parâmetros PID, problemas de manufatura, de linha de montagem e de *layout* industrial, processamento de imagens, planejamento de rotas e dentre muitas outras mencionadas em Serapião (2009).

Mirjalili et al. (2014) introduziu o algoritmo otimização por lobo cinzento onde o algoritmo imita a dinâmica de busca, caça e hierarquia de uma alcateia de lobos cinzentos. Onde no algoritmo é dividido 4 classes de lobo na pirâmide de hierarquia, o alfa, beta,

delta e omega. O algoritmo é definido em três partes, simulando a situação de caça do lobo cinzento, para isso é simulado a localização e o rodeamento ao redor da presa durante a caça e o ataque. Inicialmente, a alcateia artificial é posta em um espaço de busca e é determinado os lobos alfa, beta e delta por meio do qual está menos distante da presa, ou seja, do objetivo da otimização. Os demais lobos são denominados como omegas realizando o circulamento da presa. O ataque é simulado pela redução da diferença entre o valor do alfa e o valor da presa, quando a mesma para de se movimentar. O problema deste algoritmo é estar propenso a estagnação em soluções locais.

Outro algoritmo a ser apresentado é o *particle swarm optimization*. Introduzido por Kennedy e Eberhart (1995), este algoritmo gera uma população de partículas gerais nos quais sua característica de exploração e refino da busca ótima são definidas por poucos parâmetros quando comparado aos demais algoritmos de *swarm intelligence*. Dentre os membros do grupo é escolhido o valor da melhor partícula como referência onde todos os membros serão influenciados a seguir para tal direção não descartando o melhor valor obtido individualmente.

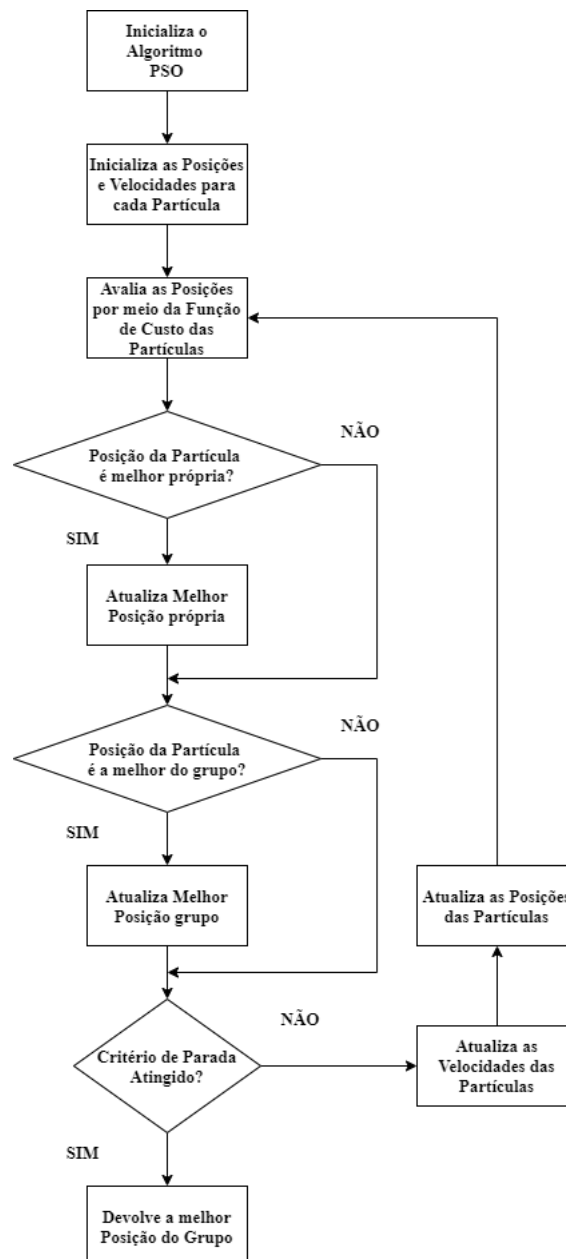
Além dos algoritmos apresentados muitos outros foram desenvolvidos como otimização de morcego, lobo cinzento, macaco, leão e entre outros mais (WAHAB et al., 2015, 2015; ROY et al., 2014; CHAKRABORTY; KAR, 2017). Como mencionado anteriormente, os algoritmos de *swarm intelligence* possuem diferentes características que influenciam diretamente como a resposta final é tratada, a velocidade na qual se obtém a melhor resposta, a robustez da mesma, a facilidade de sua implementação (YU et al., 2015). De acordo com o teorema de *No Free Lunch*, não há um algoritmo de otimização que abranja todos os problemas, pois, para diferentes problemas alguns algoritmos poderão se sobressair, ademais, diferentes configurações do mesmo algoritmo podem impactar no desempenho da resolução do problema. Neste trabalho é ressaltado o algoritmo *particle swarm optimization*, pois, mesmo possuindo desvantagens similares aos demais algoritmos como a possibilidade de ficar preso em um ótimo local, ele possui uma fácil implementação, uma eficiência na otimização para espaços de busca de grandes dimensões, uma resposta robusta (YU et al., 2015; WAHAB et al., 2015) e por ser bem conhecido na comunidade científica.

3.1 Particle Swarm Optimization

Introduzido inicialmente por Kennedy e Eberhart (1995), o algoritmo *Particle Swarm Optimization* (PSO) é um processo de otimização de procura estocástica baseada em

população (ABRAHAM et al., 2006). Este algoritmo funciona criando uma população de partículas em uma dimensão de parâmetros, N , de um problema ou função que buscará a melhor posição entre as partículas, sendo refinada a cada iteração através da atração das demais partículas para melhor posição (POLI, 2008), mantendo salvo sua individual melhor posição encontrada (LIN, 2006), como é representado na Figura 6.

Figura 6- Diagrama do Algoritmo PSO



Fonte: Autor Próprio

De acordo com o diagrama na Figura 6 a velocidade é atualizada antes da posição,

pois, a equação da posição (8) utiliza a equação da velocidade (9).

$$x_{k+1,N} = x_{k,N} + v_{k+1,N} \quad (8)$$

$$v_{k+1,N} = w_k v_{k,N} + c_1 r_1 (p_{k,N} - x_{k,N}) + c_2 r_2 (g_{k,N} - x_{k,N}) \quad (9)$$

Os termos de (8) e (9) são apresentados na tabela 1.

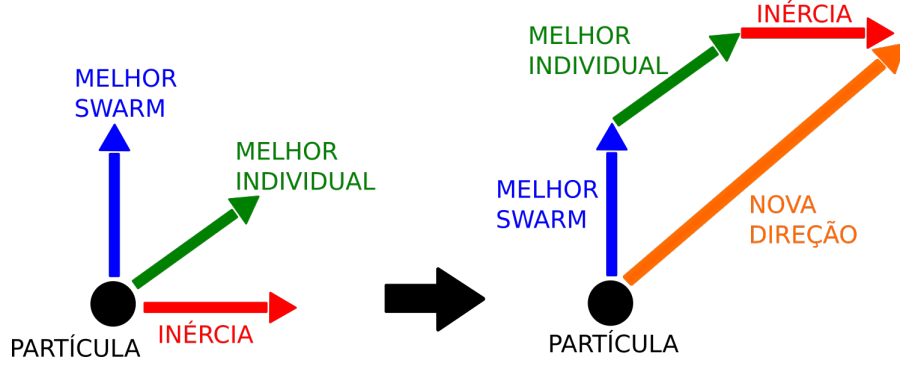
Tabela 1- Hiperparâmetros do Algoritmo PSO

Variável	Definição
Posição da partícula	$x_{k,N}$
Velocidade da partícula	$v_{k,N}$
Melhor posição individual	$p_{k,N}$
Melhor posição do <i>swarm</i>	$g_{k,N}$
Constante inercial de peso	w_k
Parâmetro cognitivo	c_1
Parâmetro Social	c_2
Número Aleatório entre 0 e 1	r_1, r_2

Fonte: Autor Próprio

Os termos c_1 e c_2 representam o caráter que o algoritmo PSO irá possuir, caso o termo c_1 for maior que c_2 , a população irá possuir um caráter exploratório podendo extrapolar os valores da posição no espaço de busca e se caso c_2 for maior que c_1 , a população possuirá um caráter refinador buscando acompanhar, sem muita exploração do espaço de busca, o melhor valor da população (PARSOPOULOS; VRAHATIS, 2010). Os termos aleatórios r_1 e r_2 além de ajudarem as partículas não ficarem nas mesmas posições, eles inserem um termo de característica para cada partícula a cada iteração, variando a partícula em exploratória e refinadora, independente da característica da população (ABRAHAM et al., 2006). O parâmetro w_k representa o quanto a velocidade anterior influenciará na velocidade atual, impactando diretamente na habilidade de exploração global e local no espaço de busca (ABRAHAM et al., 2006). Deste modo a direção que a partícula seguirá pode ser representada por uma soma vetorial como na Figura 7.

Figura 7- Decisão vetorial para novo movimento da partícula



Fonte: Autor Próprio

A posição da partícula no espaço de procura é representado por meio da matriz de posição $x_{k,N} = [x_{k,1}, x_{k,2}, \dots, x_{k,N}]$ e a melhor posição individual ($p_{k,N}$), a melhor posição do *swarm* ($g_{k,N}$) e a velocidade ($v_{k,N}$) são representados de maneira similar matricialmente (ZHAO et al., 2005).

Durante a simulação há a possibilidade de extrapolação das posições e velocidades das partículas devido os parâmetros cognitivos, social e de peso, deste modo, para evitar esse comportamento são utilizados diferentes meios de controle, como a definição de condições de contorno no espaço de busca da posição e velocidade,

$$\begin{aligned} x_{k,N}^{min,max} &= \begin{bmatrix} (x_{k,1}^{min}, x_{k,1}^{max}) & (x_{k,2}^{min}, x_{k,2}^{max}) & \dots & (x_{k,N-1}^{min}, x_{k,N-1}^{max}) & (x_{k,N}^{min}, x_{k,N}^{max}) \end{bmatrix} \\ v_{k,N}^{min,max} &= \begin{bmatrix} (v_{k,1}^{min}, v_{k,1}^{max}) & (v_{k,2}^{min}, v_{k,2}^{max}) & \dots & (v_{k,N-1}^{min}, v_{k,N-1}^{max}) & (v_{k,N}^{min}, v_{k,N}^{max}) \end{bmatrix} \end{aligned} \quad (10)$$

E também pode ser definido um decaimento linear no parâmetro de peso,

$$w_k = w_{max} - Ite_k \left(\frac{w_{max} - w_{min}}{Ite_{max}} \right) \quad (11)$$

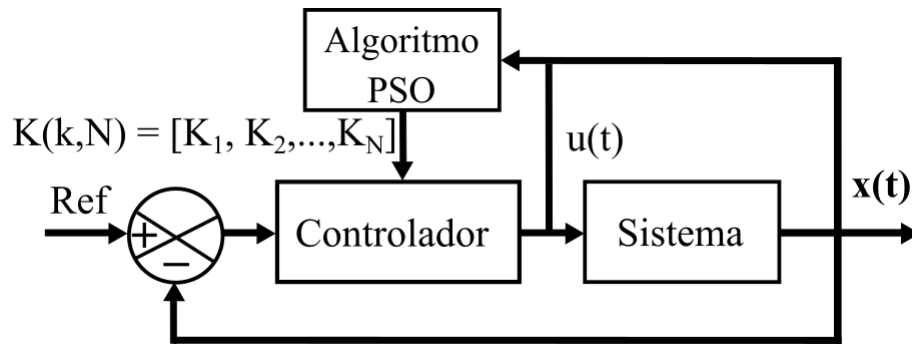
w_{max} é o valor inicial de peso, w_{min} é o valor final do peso, Ite_{max} é o número de iterações máxima e Ite_k é a iteração atual. Tais meios mencionados anteriormente influenciam diretamente nas características exploratório e na velocidade de conversão do algoritmo (KIRANYAZ et al., 2014).

3.2 Controle PSO

O controle por meio do algoritmo PSO pode ser realizada por duas configurações, identificando parâmetros físicos do sistema para aplicar leis de controle conhecidas e modelando diretamente o vetor de controle independente das configurações físicas do sistema, tais configurações são representadas na Figura 5, do controle com *machine learning*. Tendo

um foco na segunda configuração 5b, o algoritmo PSO utiliza uma população partículas, sendo cada partícula um vetor de controle e cada elemento do vetor representa uma posição no espaço de busca do algoritmo, deste modo, podendo ser retratado pela seguinte matriz $\mathbf{K}(k, N) = [K_{k,1}, K_{k,2}, \dots, K_{k,N}]$. Desta forma, cada partícula irá simular a resposta do ganho para um sistema e calcular um custo para sua posição por meio de uma função de custo que engloba os estados $\mathbf{x}(t)$ e a força de controle $u(t)$, e enviará para o algoritmo. Em sequência o algoritmo avalia o custo e determina qual a melhor posição/vetor de controle do grupo e calcula novas posições e direções para as partículas buscarem posições melhores ou convergirem na posição melhor encontrada. Este esquema de controle é representado na Figura 8.

Figura 8- Esquema de Controle via PSO

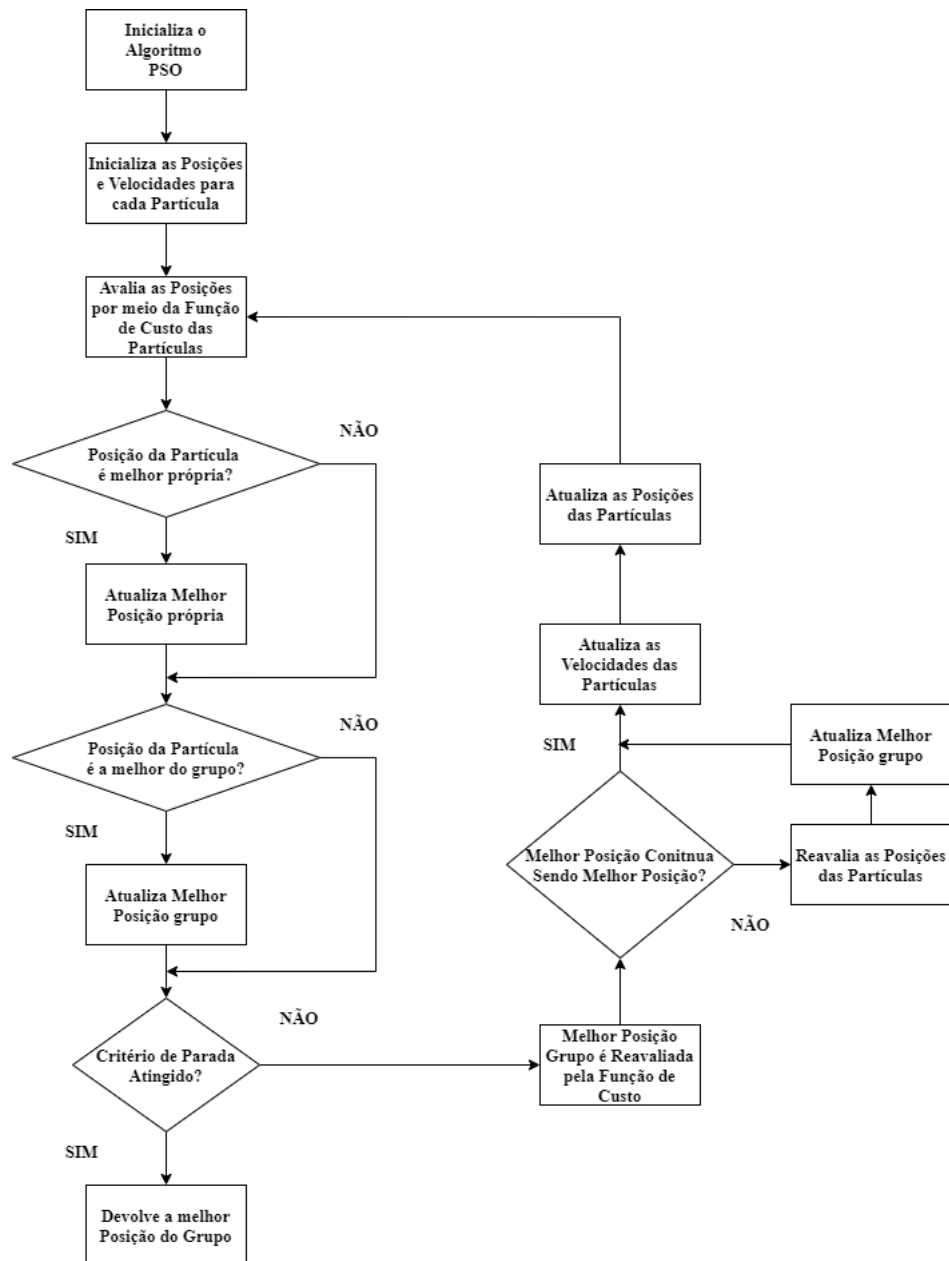


Fonte: Autor Próprio

Assim, o algoritmo determina a melhor posição e explora o espaço de busca com visando de minimizar a função de custo. Durante as iterações do algoritmo parâmetros do sistema irão ser alterados para simular mudanças inesperadas para avaliar o desempenho do melhor ganho encontrado pelo algoritmo. Contudo, o algoritmo básico do PSO, Figura 6, não contempla mudanças no sistema ou ambiente, o que ocasiona nas partículas serem direcionadas para uma direção não ótima (RINI et al., 2011). Diferentes soluções são estudadas para mudanças no ambiente ou no sistema, como detecção de mudança no ambiente, mudar a atualização do parâmetro w_k , reinicialização das partículas e busca local de acordo com Rini et al. (2011).

Neste trabalho é implementado uma detecção de mudanças no ambiente reavaliando a melhor posição ao final da iteração para determinar se o custo desta posição continua sendo a melhor dentre as partículas no grupo, caso contrário as partículas são reavaliadas e é escolhido uma nova melhor posição. Deste modo o fluxograma da Figura 6 é reestruturado para o fluxograma da Figura 9, para o algoritmo PSO contemplar mudanças no ambiente ou no sistema que influenciam diretamente na direção que o grupo deve seguir.

Figura 9- Diagrama Modificado do Algoritmo PSO

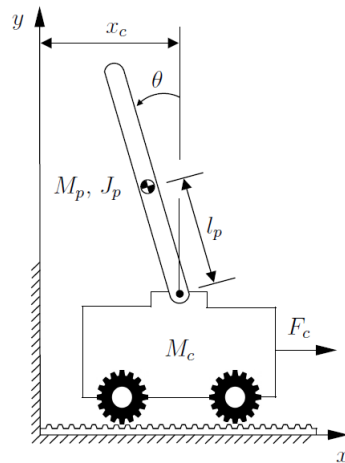


Fonte: Autor Próprio

4 MODELAGEM

O sistema pêndulo invertido é um problema clássico em teoria de controle devido a sua dinâmica não linear de alta ordem e fase mínima. Sua modelagem se aplica em problemas de controle de altitude de foguete espacial e de satélite, pouso automático em aeronaves, estabilização de aeronaves em fluxos turbulentos e estabilização de uma cabine de navio (AKHTARUZZAMAN; SHAFIE, 2010). Para a modelagem do pêndulo invertido é considerado o sistema a seguir:

Figura 10- Sistema mecânico pêndulo invertido



Fonte: Adaptado de Netto et al. (2018)

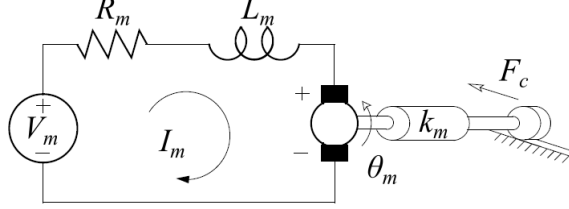
Com o sistema dinâmico apresentado na Figura 10, este capítulo apresentará a modelagem do motor que aplicará a força de controle sob o sistema, a modelagem da dinâmica do carro e do pêndulo e por fim a dinâmica do sistema em espaço de estados de forma linearizada e não linear.

4.1 Motor

Como atuador, exercendo a força de controle sobre o sistema, é considerado um motor de corrente contínua acoplado no carro. Para uma modelagem realista do sistema em questão é necessário compreender a dinâmica do atuador. Um motor de corrente contínua convencional é um sistema eletromecânico que converte tensão em torque ao redor de seu eixo, que pode ser representado pelo diagrama elétrico da Figura 11 onde V_m é a tensão

de entrada, R_m resistência de armadura, I_m é a corrente de armadura, L_m a indutância do motor, θ_m é ângulo de rotação, k_m é a constante eletromotriz e F_c é a força de controle exercida no sistema (NETTO et al., 2018):

Figura 11- Diagrama elétrico do motor de corrente contínua



Fonte: Adaptado de Netto et al. (2018)

Na Figura 11 é perceptível que V_m é convertida pelo motor em F_c , deste modo, é possível construir uma relação entre a tensão de entrada no motor e a força que é aplicada no sistema, através da Lei de Kirchhoff, (12) é obtida:

$$V_m(t) - R_m I_m(t) - L_m \dot{I}_m(t) - k_m \dot{\theta}_m(t) = 0 \quad (12)$$

Considerando a indutância do motor desprezível, $L_m \approx 0$, (NETTO et al., 2018) a corrente de armadura pode ser isolada tal que:

$$I_m(t) = \frac{V_m(t) - k_m \dot{\theta}_m(t)}{R_m} \quad (13)$$

Em motores de corrente contínua o torque gerado é proporcional à corrente de armadura:

$$\tau_m(t) = \eta_m k_t I_m(t) \quad (14)$$

Sendo η_m é a eficiência do motor e o k_t é a constante de proporcionalidade corrente-torque. Substituindo (13) em (14) é possível obter o torque em função da tensão de entrada e da velocidade angular:

$$\tau_m(t) = \eta_m k_t \frac{V_m(t) - k_m \dot{\theta}_m(t)}{R_m} \quad (15)$$

O torque pode ser, também, representado em função da força de controle, onde é considerado k_g uma constante de redução realizada pela caixa de transmissão, η_g uma

eficiência de transmissão de movimento e r_{pm} o raio do pinhão:

$$\tau_m(t) = \frac{F_c(t)r_{pm}}{\eta_g k_g} \quad (16)$$

Deste modo, a força de controle (16) pode ser isolada como:

$$F_c(t) = \frac{\eta_g k_g \tau_m(t)}{r_{pm}} \quad (17)$$

Em (12), a velocidade angular é outro componente a ser determinado. Determinado uma relação entre a posição angular e a posição linear utilizando r_{pm} e k_g :

$$\theta_m(t) = \frac{k_g x_c(t)}{r_{pm}} \quad (18)$$

Derivando (18) é obtida a seguinte expressão para velocidade angular:

$$\dot{\theta}_m(t) = \frac{k_g \dot{x}_c(t)}{r_{pm}} \quad (19)$$

Assim, a força de controle pode ser descrita em função da tensão de entrada, $V_m(t)$, e da velocidade linear, $x_c(t)$, substituindo (17) e (15) em (19):

$$F_c(t) = \frac{\eta_g \eta_m k_g k_t}{r_{pm} R_m} \left(V_m(t) - \frac{k_m k_g}{r_{pm}} \dot{x}_c(t) \right) \quad (20)$$

Com (20) é possível obter a força que o motor de corrente contínua exercerá sobre o carro que terá a dinâmica discutida na seção a seguir.

4.2 Carro

O carro é movimentado através da força exercida sobre ele através do motor de corrente contínua que através da rotação do pinhão desloca-se horizontalmente sobre a cremalheira. No carro são consideradas algumas forças como uma força de atrito viscosa de constante B_c , uma força inercial de armadura $F_a(t)$ gerada devido ao torque de inércia e uma força gerada pela haste do pêndulo conectada ao carro $T_x(t)$ (NETTO et al., 2018). Sendo assim, adotando os referenciais da Figura 10 e aplicando o princípio de D'Alembert a equação de movimento do carro é dado por:

$$M_c \ddot{x}_c(t) + F_a(t) + T_x + B_c \dot{x}_c(t) = F_c(t) \quad (21)$$

A força inercial da armadura é equacionada em função do torque de inércia :

$$F_a(t) = \frac{\eta_g k_g \tau_{inercia}(t)}{r_{pm}} \quad (22)$$

Sendo o torque de inércia representado por:

$$\tau_{inercia}(t) = J_m \ddot{\theta}_m(t) = J_m \left(\frac{k_g \ddot{x}_m(t)}{r_{pm}} \right) \quad (23)$$

Substituindo (23) em (22) é obtido:

$$F_a(t) = \frac{\eta_g k_g^2 J_m \ddot{x}_c(t)}{r_{pm}^2} \quad (24)$$

Substituindo a força de controle (20), e a força inercial de armadura (24), na equação de movimento do carro (21), é possível obter:

$$M_c \ddot{x}_c(t) + \frac{\eta_g k_g^2 J_m \ddot{x}_c(t)}{r_{pm}^2} + T_x + B_c \dot{x}_c(t) = \frac{\eta_g \eta_m k_g k_t}{r_{pm} R_m} \left(V_m(t) - \frac{k_m k_g}{r_{pm}} \dot{x}_c(t) \right) \quad (25)$$

Simplificando (25):

$$M_{eq} \ddot{x}_c(t) + B_{eq} \dot{x}_c(t) + T_x = A_{eq} V_m(t) \quad (26)$$

Sendo, M_{eq} , B_{eq} e A_{eq} são a massa equivalente, coeficiente de atrito viscoso equivalente e o ganho de atuação equivalente, respectivamente, tais quais são representados por:

$$M_{eq} = M_c + \frac{\eta_g k_g^2 J_m}{r_{pm}^2} \quad (27)$$

$$B_{eq} = B_c + \frac{\eta_g \eta_m k_g^2 k_t k_m}{r_{pm}^2 R_m} \quad (28)$$

$$A_{eq} = \frac{\eta_g \eta_m k_g k_t}{r_{pm} R_m} \quad (29)$$

4.3 Pêndulo

Diferente do carro, que apenas movimenta-se horizontalmente, o pêndulo possui dois tipos de movimento, rotação e translação. Como seu momento de inércia, J_p , é localizado a uma distância l_p do seu centro de gravidade ao ponto de pivoteamento, o movimento de

translação pode ser equacionado como:

$$M_p \frac{d^2}{dt^2} (x_c(t) - l_p \sin(\theta(t))) - T_x = 0 \quad (30)$$

E seu movimento de rotação gera um atrito no ponto de pivoteamento com coeficiente de atrito, B_p resultando na seguinte equação:

$$(J_p + M_p l_p^2) \ddot{\theta}(t) - M_p l_p \cos(\theta(t)) \ddot{x}_c(t) - M_p g l_p \sin(\theta(t)) + B_p \dot{\theta}(t) = 0 \quad (31)$$

4.4 Equações Não Lineares

Com (26), (30) e (31) a dinâmica do conjunto pêndulo-carro é descrita completamente. Entretanto (26) e (30) possuem um coeficiente em comum, T_x , podendo ser combinadas para formar uma nova equação, e por fim generalizar a dinâmica do sistema de forma não linear com as seguintes equações:

$$(M_p + M_{eq}) \ddot{x}_c(t) + B_{eq} \dot{x}_c(t) - M_p l_p \cos(\theta(t)) \ddot{\theta}(t) + M_p l_p \sin(\theta(t)) \dot{\theta}(t)^2 = A_{eq} V_m(t) \quad (32)$$

$$(M_p l_p^2 + J_p) \ddot{\theta}(t) + B_p \dot{\theta}(t) - M_p l_p \cos(\theta(t)) \ddot{x}_c(t) - M_p g l_p \sin(\theta(t)) = 0 \quad (33)$$

4.5 Linearização

A linearização é um processo no qual é encontrado um modelo linear que possua uma boa aproximação do sistema não linear em uma pequena região, conhecida como ponto de operação (SEBORG et al., 2017). O ponto de operação pode coincidir com o ponto de equilíbrio de um sistema, considerando que um ponto $\bar{x} \in \mathbb{R}^n$ é um ponto de equilíbrio se existir um $\bar{u} \in \mathbb{R}^m$ de tal modo que $f(\bar{x}, \bar{u}) = 0_n$, sendo f uma função de mapeamento $\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ (PACKARD et al., 2005). Para linearizar um sistema em um ponto de operação $\bar{x}$ e $\bar{u}$, é necessário definir funções f para os estados do sistema e por meio da matriz Jacobiana (34), encontrar as matrizes $\mathbf{A}$ e $\mathbf{B}$ obtendo o sistema em sua forma linearizada ao redor do ponto de operação $\bar{x}$ e $\bar{u}$.

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_n} \end{bmatrix} \quad (34)$$

Neste trabalho os estados do sistema pêndulo invertido são respectivamente o deslo-

camento do carro, x_c , o deslocamento angular do pêndulo, θ , a velocidade do carro, $\dot{x}_c$, a velocidade angular do pêndulo, $\dot{\theta}$ e a força de controle $u = V_m$. As funções para cada estado é representado matricialmente em (35).

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} x_c \\ \theta \\ \dot{x}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x}_c \\ \dot{\theta} \\ \frac{cB_{eq}\dot{x}_c + c\dot{\theta}^2 d - cA_{eq}V_m + B_p\dot{\theta}b - eb}{J_{Tn}} \\ \frac{aB_{eq}\theta - ea + B_{eq}\dot{x}_c b - b\dot{\theta}^2 d - bA_{eq}V_m}{J_{Tn}} \end{bmatrix} \quad (35)$$

Sendo $a = M_p + M_{eq}$, $b = M_p l_p \cos(\theta)$, $c = M_p l_p^2 + J_p$, $d = M_p l_p \sin(\theta)$, $e = M_p g l_p \sin(\theta)$ e $J_{Tn} = (M_p l_p \cos(\theta))^2 - (M_p + M_{eq})(M_p l_p^2 + J_p)$. Por meio da matriz Jacobian, (36), é possível obter as matrizes **A** e **B**.

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \\ \frac{\partial f_4}{\partial u} \end{bmatrix} \quad (36)$$

Deste modo, realizando as derivadas parciais da Jacobiana para os pontos de operação $\bar{\mathbf{x}} = [\bar{x}_c \quad \bar{\theta} \quad \bar{\dot{x}}_c \quad \bar{\dot{\theta}}]$ e $\bar{u} = \bar{V}_m$, é obtido as seguintes matrizes **A** e **B** apresetandas em (37).

$$\mathbf{A}(\bar{\mathbf{x}}, \bar{u}) = \frac{1}{J_{Tn}(\bar{\mathbf{x}})} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & c\bar{\theta}^2 b - B_p \bar{\theta} d - b^2 g - d^2 g & cB_{eq} & 2c\bar{\theta} d + B_p b \\ 0 & aB_{eq} - bga - B_{eq}\bar{\dot{x}}_c d + d^2 \bar{\theta}^2 - b^2 \bar{\theta}^2 + dA_{eq}\bar{V}_m & B_{eq}b & 2b\bar{\theta} d \end{bmatrix} \quad (37)$$

$$\mathbf{B}(\bar{\mathbf{x}}) = \frac{A_{eq}}{J_{Tn}(\bar{\mathbf{x}})} \begin{bmatrix} 0 \\ 0 \\ -(M_p l_p^2 + J_p) \\ -(M_p l_p \cos(\bar{\theta})) \end{bmatrix}$$

Sendo $J_{Tn}(\bar{\mathbf{x}}) = (M_p l_p \cos(\bar{\theta}))^2 - (M_p + M_{eq})(M_p l_p^2 + J_p)$. Deste modo o sistema é representado em sua forma linear em espaço de estados:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(x) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t) \end{cases} \quad (38)$$

Com a representação dinâmica do sistema em sua forma linear é possível desenvolver leis de controles efetivas que funcionam bem próximos ao ponto de operação o qual o sistema foi linearizado (PACKARD et al., 2005).

5 TÉCNICAS DE CONTROLE

Neste capítulo é apresentado as formulações de leis de controle utilizadas para gerar as populações iniciais do algoritmo PSO. Tais formulações são importantes no algoritmo para desenvolver populações de ganhos estáveis, pois, deste modo auxiliará na diversidade racional da população e orientação para uma região ótima com o menor número de gerações possíveis (HE et al., 2010).

5.1 Controle Ótimo

De acordo com Jha et al. (2016) o regulador quadrático linear (LQR) é um dos principais métodos em teoria de controle ótimo, sendo (39) o controle em *feedback*.

$$u(t) = -\mathbf{K}_{lqr}\mathbf{x}(t) \quad (39)$$

Para determinar o ganho $\mathbf{K}_{lqr}$ que minimize a função de custo quadrático,

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + u^T \mathbf{R} u) dt \quad (40)$$

onde as matrizes $\mathbf{Q}$ e $\mathbf{R}$ penalizam o erro e o gasto energético, é substituído (39) na primeira parcela do equacionamento em espaço de estados de (38) obtendo,

$$\dot{\mathbf{x}} = \mathbf{A} - \mathbf{BK}\mathbf{x} = (\mathbf{A} - \mathbf{BK})\mathbf{x} \quad (41)$$

, assim, substituindo (41) o em (40), a função de custo é reescrita em (42).

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{x}^T \mathbf{K}^T \mathbf{R} \mathbf{K} \mathbf{x}) dt = \int_0^\infty \mathbf{x}^T (\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}) \mathbf{x} dt \quad (42)$$

Assim, realizando as devidas manipulações algébricas, substituições e considerando $\mathbf{A} - \mathbf{BK}$ uma matriz estável, é possível obter a equação de matriz reduzida de Riccati, (43) (OGATA, 2010):

$$\mathbf{AP} + \mathbf{PA} - \mathbf{PBR}^{-1}\mathbf{BP} + \mathbf{Q} = 0 \quad (43)$$

Onde é possível encontrar um matriz positiva definida, $\mathbf{P}$, que satisfaça (43) (OGATA,

2010). Deste modo, utilizando a matriz $\mathbf{P}$ é obtido o ganho ótimo por meio de (44).

$$\mathbf{K}_{lqr} = \mathbf{R}^{-1}\mathbf{B}\mathbf{P} \quad (44)$$

5.2 Robusto Ótimo

Com a técnica de controle robusto ótimo é projetado um controlador que contempla incertezas de parâmetros, Olalla et al. (2009) sugere projetar um controlador LQR que através de *Linear Matrix Inequalities* (LMI) seja possível adicionar incertezas ao projeto do controlador.

De acordo com o teorema de Lyapunov é uma condição suficiente para assegurar que um sistema é quadraticamente estável, para uma dado sistema invariante no tempo linear, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, se existir uma matriz $\mathbf{P}$ de tal modo que uma função quadrática $V(x) = x^T \mathbf{P} x > 0 \forall x \neq 0$ satisfaça $\dot{V}(t) = x^T (\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A}) x < 0 \forall x \neq 0$ (BOYD et al., 1994).

Assim, dado o sistema em espaço de estados em sua forma linear (38), aplicando o operador traço matricial $T_r(\cdot)$, que satisfaz $\mathbf{a}^T \mathbf{X} \mathbf{b} = T_r(\mathbf{X} \mathbf{b} \mathbf{a}^T)$ em (42) é obtido (45).

$$J = \int_0^\infty T_r((\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}) \mathbf{x}^T \mathbf{x}) dt \quad (45)$$

Resumindo o termo $\int_0^\infty \mathbf{x}^T \mathbf{x}$ em uma matriz $\mathbf{P}$ que satisfaz a seguinte equação:

$$(\mathbf{A} - \mathbf{B} \mathbf{K}) \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{B} \mathbf{K})^T + \mathbf{x}_0 \mathbf{x}_0^T = 0 \quad (46)$$

Deste modo, utilizando a matriz $\mathbf{P}$ equação de custo (45) é reescrita como:

$$J = T_r((\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}) \mathbf{P}) \quad (47)$$

O ganho do controle ótimo pode ser encontrado minimizando a seguinte função:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{K}} T_r(\mathbf{Q} \mathbf{P}) + T_r(\mathbf{R}^{1/2} \mathbf{K} \mathbf{P} \mathbf{K}^T \mathbf{R}^{1/2}) \\ \text{sujeito a} \\ (\mathbf{A} - \mathbf{B} \mathbf{K}) \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{B} \mathbf{K})^T + \mathbf{x}_0 \mathbf{x}_0^T < 0 \end{aligned} \quad (48)$$

Entretanto, de acordo com Olalla et al. (2009) (48) não é linear devido a multiplicação entre as matrizes $\mathbf{K}$ e $\mathbf{P}$, então definindo uma nova quantidade $\mathbf{Y} = \mathbf{K} \mathbf{P}$ a (48) é reescrita

como (49) (FERON et al., 1992).

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{Y}} \quad & T_r(\mathbf{QP}) + T_r(\mathbf{R}^{1/2} \mathbf{Y} \mathbf{P}^{-1} \mathbf{Y}^T \mathbf{R}^{1/2}) \\ & \text{sujeito a} \\ & \mathbf{AP} - \mathbf{BY} + \mathbf{PA}^T - \mathbf{Y}^T \mathbf{B}^T + \mathbf{x}_0 \mathbf{x}_0^T < 0, \text{ sendo } \mathbf{P} = \mathbf{P}^T > 0 \end{aligned} \quad (49)$$

Adicionalmente, o termo $\mathbf{R}^{1/2} \mathbf{Y} \mathbf{P}^{-1} \mathbf{Y}^T \mathbf{R}^{1/2}$ pode ser substituído por uma segunda variável auxiliar $\mathbf{X}$, que pode ser decomposto com o complemento de Shur,

$$\mathbf{X} > \mathbf{R}^{1/2} \mathbf{Y} \mathbf{P}^{-1} \mathbf{Y}^T \mathbf{R}^{1/2} \Leftrightarrow \begin{bmatrix} \mathbf{X} & \mathbf{R}^{1/2} \mathbf{Y} \\ \mathbf{Y}^T \mathbf{R}^{1/2} & \mathbf{P} \end{bmatrix} > 0 \quad (50)$$

deste modo a formulação em LMI do problema ótimo é representado por (51).

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{Y}, \mathbf{X}} \quad & T_r(\mathbf{QP}) + T_r(\mathbf{X}) \\ & \text{sujeito a} \\ & \mathbf{AP} - \mathbf{BY} + \mathbf{PA}^T - \mathbf{Y}^T \mathbf{B}^T + \mathbf{x}_0 \mathbf{x}_0^T < 0 \\ & \begin{bmatrix} \mathbf{X} & \mathbf{R}^{1/2} \mathbf{Y} \\ \mathbf{Y}^T \mathbf{R}^{1/2} & \mathbf{P} \end{bmatrix} > 0 \quad \mathbf{P} > 0 \end{aligned} \quad (51)$$

Com o problema formulado em LMI, o ganho ótimo pode ser obtido com $\mathbf{K} = \mathbf{P}^{-1} \mathbf{Y}$. Para adicionar incertezas ao cálculo do ganho, inicialmente os termos incertos são agrupados em um vetor $\mathbf{p}$ que consiste em n_p parâmetros incertos, $\mathbf{p} = (p_1, \dots, p_{n_p})$, onde cada um é limitado a um valor máximo e mínimo. Generalizando, os valores admissíveis para o vetor $\mathbf{p}$ são restritos em um hiper-retângulo no espaço de parâmetro $\mathbb{R}^{n_p}$ com $N = 2^{n_p}$ vértices, $\{v_1, \dots, v_N\}$. A imagem da matriz $\begin{bmatrix} \mathbf{A}(p) & \mathbf{B}(p) \end{bmatrix}$, para cada vértice v_i , corresponde a um conjunto $\{G_1, \dots, G_N\}$ que são os extremos de um politopo convexo que contém as imagens para todos os valores de $\mathbf{p}$ se a matriz $\begin{bmatrix} \mathbf{A}(p) & \mathbf{B}(p) \end{bmatrix}$ depender linearmente de $\mathbf{p}$ (SARKAWI; OHTA, 2018). Deste modo, a formulação em LMI, 51, pode ser reescrita considerando o modelo politópico:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{Y}, \mathbf{X}} \quad & T_r(\mathbf{QP}) + T_r(\mathbf{X}) \\ & \text{sujeito a} \\ & \mathbf{A}_i \mathbf{P} - \mathbf{B}_i \mathbf{Y} + \mathbf{PA}_i^T - \mathbf{Y}^T \mathbf{B}_i^T + \mathbf{x}_0 \mathbf{x}_0^T < 0. \quad (i = 1, 2, 3, 4) \\ & \begin{bmatrix} \mathbf{X} & \mathbf{R}^{1/2} \mathbf{Y} \\ \mathbf{Y}^T \mathbf{R}^{1/2} & \mathbf{P} \end{bmatrix} > 0 \quad \mathbf{P} > 0 \end{aligned} \quad (52)$$

Sendo que, a solução da formulação (52) produzirá uma matriz $\mathbf{P}$ que satisfaz a inequação para todos os vértices do politopo convexo.

6 METODOLOGIA

Para realizar as simulações de controle deste trabalho é necessário, inicialmente definir os parâmetros do sistema pêndulo invertido, definir a função de custo e seus parâmetros para avaliar o desempenho do controle, as populações a serem testadas, as mudanças realizadas durante as iterações do algoritmo e por fim os parâmetros do algoritmo para então realizar o controle.

6.1 Sistema

Neste trabalho foi utilizado o sistema pêndulo invertido, apresentado na seção 4, e seus parâmetros foram selecionados a partir de dados da empresa *Quanser* que fornece diferentes sistemas de controle, como é visto na tabela 2.

Tabela 2- Dados do Sistema Pêndulo Invertido

Variável	Valor	Unidade
M_c, M_p	0.94, 0.23	$[kg]$
l_p, r_{pm}	0.3302, $6.35 \cdot 10^{-3}$	$[m]$
J_p	$7.88 \cdot 10^{-3}$	$[kg.m^2]$
J_m	$3.9 \cdot 10^{-7}$	$[kg.m^2]$
B_{eq}, B_p	5.4, 0.0024	$[N.m/m^2]$
r	2.6	$[ohm]$
k_t	0.00767	$[N.m]$
k_m	0.00767	$[V/(rad/s)]$
η_g, η_m, k_g	1, 1, 3.71	-

Fonte: Autor Próprio

Com os parâmetros do sistema selecionados a seguir é definido a função de custo utilizada pelo algoritmo para avaliar o desempenho dos vetores de ganhos.

6.2 Função de Custo

Com os parâmetros do sistema pêndulo invertido definidos, em seguida é definido a função que irá avaliar o desempenho do controlador. Então, como função de custo é selecionada a função quadrática (53), deste modo por meio das matrizes $\mathbf{Q}$ e $\mathbf{R}$ é possível obter diferentes direções para o algoritmo seguir, considerando $\mathbf{Ref} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$.

$$\sum_{t=0}^{t=t_f} J(\mathbf{x}, u) = \int_{t_1}^{t_2} ((\mathbf{Ref}^T - \mathbf{x}^T(t))\mathbf{Q}(\mathbf{Ref} - \mathbf{x}(t)) + \mathbf{R}u^2) dt \quad (53)$$

Para definir as matrizes $\mathbf{Q}$ e $\mathbf{R}$ de (53) foram realizados simulações teste utilizando o algoritmo PSO para avaliar quais os melhores valores para as matrizes. Foram realizados 4 simulações teste com 4 combinações de valores.

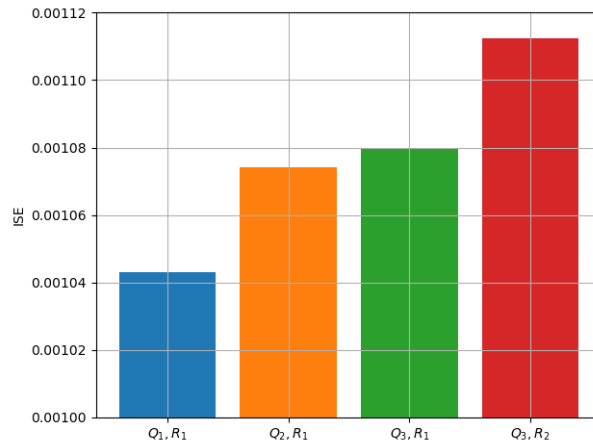
$$\begin{aligned} \mathbf{Q}_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{R}_1 = 0.5, & \mathbf{Q}_2 &= \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{R}_1 = 0.5 \\ \mathbf{Q}_3 &= \begin{bmatrix} 0.75 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{R}_1 = 0.5, & \mathbf{Q}_3 &= \begin{bmatrix} 0.75 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{R}_2 = 0.75 \end{aligned} \quad (54)$$

Tais valores foram selecionados buscando obter uma resposta com baixo erro e baixo consumo energético. Para avaliar o desempenho de cada combinação de matrizes foi utilizado os índices de consumo energético (55) e o erro quadrado integral (ISE) (56), pois a função de custo (53) é uma combinação desses dois índices de performance de controladores.

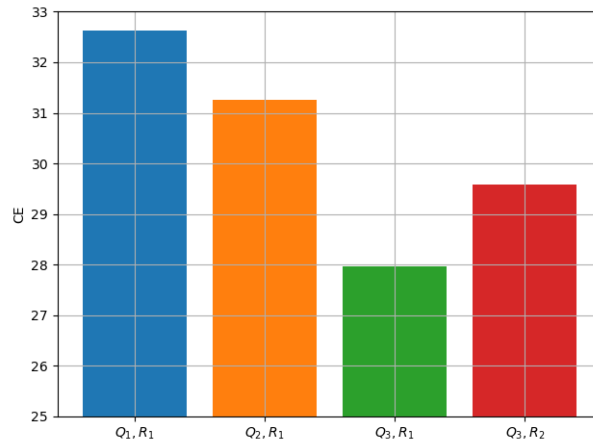
$$CE = \sum_{t_i}^{t_f} |u(t)| \quad (55)$$

$$ISE = \int_{t_i}^{t_f} (\mathbf{Ref}^T - \mathbf{x}^T(t))(\mathbf{Ref} - \mathbf{x}(t))dt \quad (56)$$

Ao final das simulações foi possível obter as seguintes resposta nas Figuras 12 e 13.

Figura 12- Índice ISE para $\mathbf{Q}_1, \mathbf{R}_1$, $\mathbf{Q}_2, \mathbf{R}_1$, $\mathbf{Q}_3, \mathbf{R}_1$ e $\mathbf{Q}_3, \mathbf{R}_2$ 

Fonte: Autor Próprio

Figura 13- Índice CE para $\mathbf{Q}_1, \mathbf{R}_1$, $\mathbf{Q}_2, \mathbf{R}_1$, $\mathbf{Q}_3, \mathbf{R}_1$ e $\mathbf{Q}_3, \mathbf{R}_2$ 

Fonte: Autor Próprio

Por meio dos resultados é possível observar nos índices ISE que os valores ficaram muito próximos com uma diferença decimal de 10^{-4} , contudo no consumo energético a combinação com menor valor foi $\mathbf{Q}_3$ e $\mathbf{R}_1$, sendo assim, foram selecionadas as matrizes $\mathbf{Q}_3$ e $\mathbf{R}_1$ para (53). Os códigos utilizados para esses testes podem ser encontrados no Apêndice A.

A seguir é apresentando os hiperparâmetros utilizados no algoritmo PSO.

6.3 Parâmetros do Algoritmo

Para os parâmetros do algoritmo c_1 , c_2 , w_k e condições de contorno, seus valores foram selecionados com objetivo de obter um caráter exploratório no algoritmo. pois

em Latha et al. (2013) é visto que o algoritmo PSO com caráter exploratório apresenta melhores resultados, os valores dos hiperparâmetros são apresentados na tabela 1. Para os sinais das condições de contorno, foi analisado os sinais de cada elemento de um vetor de controle para o sistema pêndulo invertido calculado pela técnica de controle LQR, como é observado na matriz (57),

$$\mathbf{K}_{LQRgenerico} = \begin{bmatrix} -k_1 & k_2 & -k_3 & k_4 \end{bmatrix} \quad (57)$$

assim, é possível observar que para o sistema pêndulo invertido com os estados

$$\mathbf{x} = \begin{bmatrix} x_c & \theta & \dot{x}_c & \dot{\theta} \end{bmatrix} \quad (58)$$

os elementos da matriz de controle possuem sinais diferentes quando são multiplicados por elementos do carro e do pêndulo, pois, as referências de movimento do pêndulo e do carro são contrárias, sendo que o movimento para direita é considerado positivo para pêndulo, mas negativo para o carro. Com esses argumentos estruturados os sinais das condições de contorno são determinados e são selecionados valores grandes para permitir um movimento livre das partículas no espaço de busca.

Tabela 3- Dados do Algoritmo PSO

Variável	Valor
w_k	0.9
c_1, c_2	2, 1
$x_{1,min}, x_{1,max}$	-1000, -1
$x_{2,min}, x_{2,max}$	1, 1000
$x_{3,min}, x_{3,max}$	-1000, -1
$x_{4,min}, x_{4,max}$	1, 1000
Fonte: Autor Próprio	

Para as condições de contorno da velocidade foi utilizado (59), pois de acordo com Parsopoulos e Vrahatis (2010) a velocidade máxima de uma partícula usualmente utiliza uma fração das condições de contorno do espaço de busca, sendo n definida como um limitador do passo máximo que a velocidade pode obter, neste trabalho foi considerado $n = 100$ para que os valores das novas posições não sejam muito extrapolados.

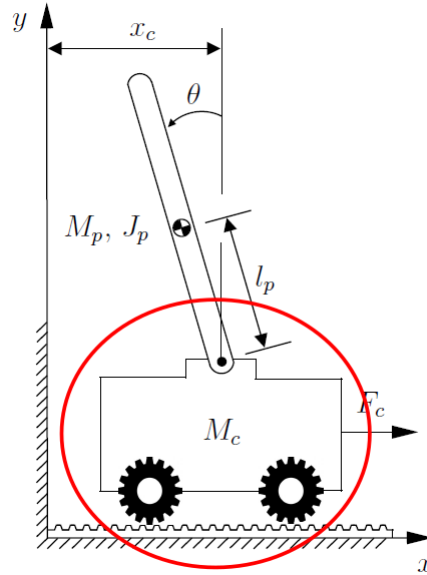
$$v_{k+1,N} = \begin{cases} v_{N,max} & \text{se } v_{k+1,N} > v_{max} \\ -v_{N,max} & \text{se } v_{k+1,N} < -v_{N,max} \end{cases} \quad (59)$$

Outro parâmetro a ser definido é o número total de iterações do algoritmo. Como o objetivo deste trabalho é gerar um controle que ao longo das iterações seja inserido mudanças no sistema, para o algoritmo se adaptar é necessário um grande número de iterações. Na simulação para encontrar os valores de $\mathbf{Q}$ e $\mathbf{R}$ para a função de custo é observado que apartir de 50 iterações o algoritmo estabiliza em uma melhor partícula, mas não converge as demais partículas. Logo é definido 300 iterações como o máximo de iterações, para que haja um intervalo razoável entre cada mudança nos parâmetros do sistema, de tal modo que haja uma semi estabilização da melhor posição e que as partículas não converjam para a mesma.

As mudanças nos parâmetros do sistema serão realizados com o intuito de simular uma perda, dano ou uma dinâmica inerente ao próprio sistema. Foram selecionados os seguintes parâmetros para serem alterados, a massa do carro, M_c , o comprimento do pêndulo, l_p , e os coeficientes de atrito B_{eq} e B_q .

Para o parâmetro de massa do carro é apenas considerado a massa do carro, como circulado na Figura 14, desconsiderando a massa do pêndulo em seu todo.

Figura 14- Sistema Pêndulo invertido focando a massa do carro

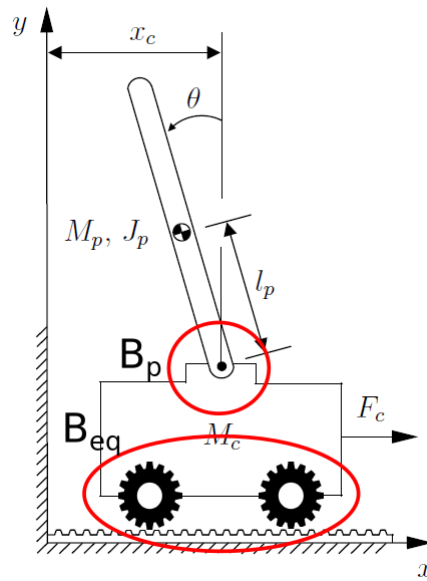


Fonte: Adaptado de Netto et al. (2018)

A mudança neste parâmetro será realizada por meio de uma perda periódica da porcentagem da massa podendo representar uma perda de combustível ou uma literal perda da carcaça do carro na qual é verificada de tempos em tempos. Especificamente foi programado para o sistema perder 20% da massa atual do carro nas iterações 50, 100, 150 e 200.

Para o parâmetro de comprimento do pêndulo foi definido a perda de um terço de seu comprimento podendo representar a perda do comprimento de um foguete ao ser lançado para órbita nas iterações 100 e 200. Os coeficientes de atrito serão aumentados em 10% nas iterações 50, 100, 150 e 200 para representar uma dificuldade na movimentação rotacional do pêndulo e na movimentação horizontal do carro, na Figura 15 é observado a localização onde dos coeficientes de atrito no sistema.

Figura 15- Sistema Pêndulo invertido focando nos coeficientes de atrito



Fonte: Adaptado de Netto et al. (2018)

E por fim, o último parâmetro a ser definido para o algoritmo é a quantidade de partículas em cada população, pois, em Parsopoulos e Vrahatis (2010) é descrito que o tamanho da população influencia diretamente na qualidade dos resultados e no nível de exploração no espaço de busca. Neste trabalho será utilizado para todas populações o tamanho de 100 partículas, pois, de acordo com Piotrowski et al. (2020) é recomendado que o tamanho da população do *swarm* seja entre 70 a 100 partículas quando se desconhece os efeitos do tamanho da população sobre a função de custo utilizadas. Por esse motivo e considerando a quantidade de iterações totais que o algoritmo terá e a quantidade de iterações para cada mudança, 100 é número não tão grande onde os resultados não serão repetitivos, não tão pequeno que o algoritmo não conseguirá explorar o espaço de busca e possivelmente explorar a capacidade de exploração da função de custo.

6.4 Populações

Para este trabalho foram geradas 5 populações diferentes, uma por meio de uma distribuição normal aleatória, duas populações foram criadas utilizando as técnicas LQR e LMI, mas para um sistema com parâmetros diferentes do sistema que será controlado e por fim duas populações também criadas com as técnicas LQR e LMI, mas com os parâmetros do sistema pêndulo invertido a ser controlado.

Para as populações geradas com a técnica LQR, variaram-se os ponto de operação para a linearização, mantendo se constante as matrizes $\mathbf{Q}$ e $\mathbf{R}$, que foram selecionados a partir das matrizes sugeridas pela empresa *Quanser* respectivamente:

$$\mathbf{Q} = \begin{bmatrix} 0.75 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{R} = 0.0012 \quad (60)$$

Por meio dessas matrizes, (60), o controle gerado considerará apenas os estados de posição do carro, x_c , posição angular do pêndulo, θ , e a tensão de controle, V_m , pois, são os estados menor erro de mensuração caso fosse implementado ao sistema real, na qual iria requerer um sensor para cada estado mencionado anteriormente e por meio desses estados realizar derivadas ou a construção de um observador para obter as respectivas velocidades, ocasionando, em uma acumulação de erros ou/e um aumento na dificuldade do controle.

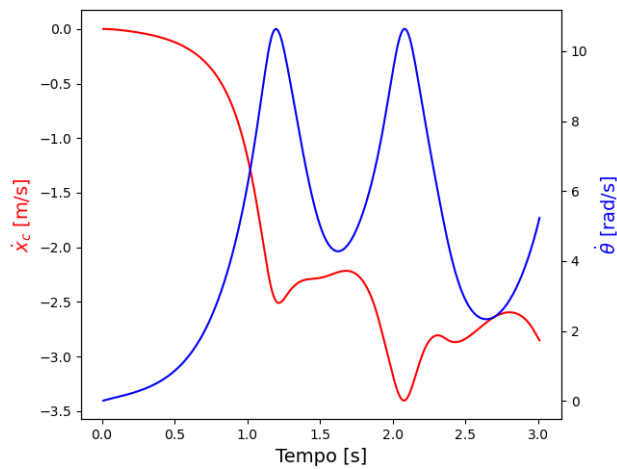
Os estados para o sistema pêndulo invertido são x_c , θ , $\dot{x}_c$, $\dot{\theta}$ e V_m e ao realizar as derivadas da matriz Jacobiana, (37), é possível observar que as matrizes $\mathbf{A}$ e $\mathbf{B}$ estão em função dos estados θ , $\dot{x}_c$, $\dot{\theta}$ e da tensão V_m . Logo, para gerar uma população de 100 partículas, serão combinados os pontos de operação, e variados entre seus máximos e mínimos, onde seus valores podem ser observados na tabela 4.

Tabela 4- Tabela de Máximos e Mínimos dos Pontos de Operação

Parâmetros	Mínimo	Máximo	Unidade
θ	0	5	[°]
$\dot{x}_c$	0	5	[m/s]
$\dot{\theta}$	0	10	[rad/s]
V_m	0	10	[V]

Tais valores da tabela 4 foram selecionados de acordo algumas limitações, como determinar que do pêndulo invertido não ultrapasse um ângulo máximo de 5° , o motor simulado da empresa *Quanser* possuir uma saturação de ± 10 volts, assim, quando simulado o sistema pêndulo invertido sem controle, Figura 16, pode ser observado que as velocidades do carro e do pêndulo não ultrapassam respectivamente 5 m/s e 10 rad/s, por isso esses valores são selecionados como os máximos dos pontos de operação da tabela 4. Os códigos utilizados para esses testes podem ser encontrados no Apêndice A.

Figura 16- Simulação das velocidades do Sistema pêndulo invertido sem controle



Fonte: Autor Próprio

Para as populações geradas por meio da formulação em LMI do controle LQR será variado a porcentagem de incerteza do parâmetro selecionado no modelo politópico. Neste trabalho, são utilizados os parâmetros de coeficientes de atrito do sistema, B_p e B_{eq} . Tais parâmetros são difíceis de serem determinados, pois, mesmo fornecido pelo fabricante eles podem sofrer variações devido a temperatura ambiente, o desgaste do material e entre outros fatores (PANTALEÓN et al., 2012).

Neste trabalho as populações geradas serão denominadas respectivamente população A, B, C, D e E. A seguir é apresentado a criação de cada população.

6.4.1 População A

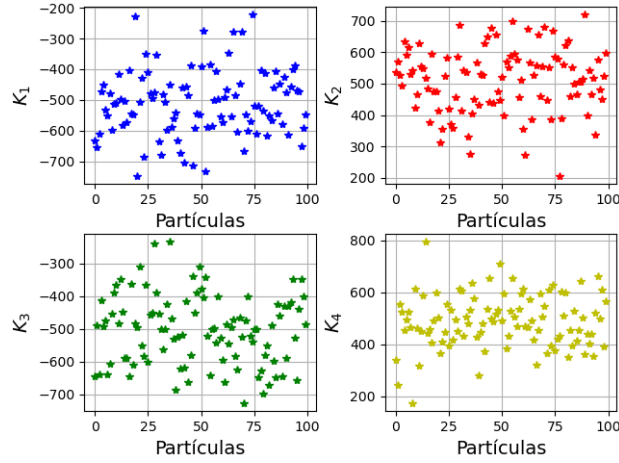
A população A foi gerada de modo aleatório por meio de uma distribuição normal (Gaussiana), definindo médias para cada espaço de busca e um desvio padrão com referência dos sinais os sinais das condições de contorno do algoritmo PSO. Neste trabalho foi definido que a distribuição possua uma média no centro do respectivo espaço de busca

limitado pelas condições de contorno, para isso, as médias das distribuições normais aleatórias foram determinadas deste modo

$$\mu = \begin{bmatrix} \frac{x_{1,min}-x_{1,max}}{2} \\ \frac{x_{2,max}-x_{2,min}}{2} \\ \frac{x_{3,min}-x_{3,max}}{2} \\ \frac{x_{4,max}-x_{4,min}}{2} \end{bmatrix} \quad (61)$$

, assim, os valores das médias foram 499.5 e -449.5, para o desvio padrão foi definido 100, para que as partículas ficassem espalhadas no espaço de busca sem extrapolar o mesmo. As posições iniciais da população A podem ser observadas a seguir na Figura 17.

Figura 17- População A



Fonte: Autor Próprio

É visto na população apresentada na Figura 17 que as partículas estão bem distribuídas dentro dos respectivos espaços de busca.

6.4.2 Populações B e C

Na geração das populações B e C será utilizada a técnica LQR, contudo, para população B os parâmetros do sistema utilizados no cálculo dos vetores de controle serão os mesmo do sistema a ser controlado, apresentado na tabela 2. Para a população C os parâmetros do sistema utilizado na elaboração dos vetores de controle serão diferentes do sistema a ser controlado, como pode ser observado na tabela 5.

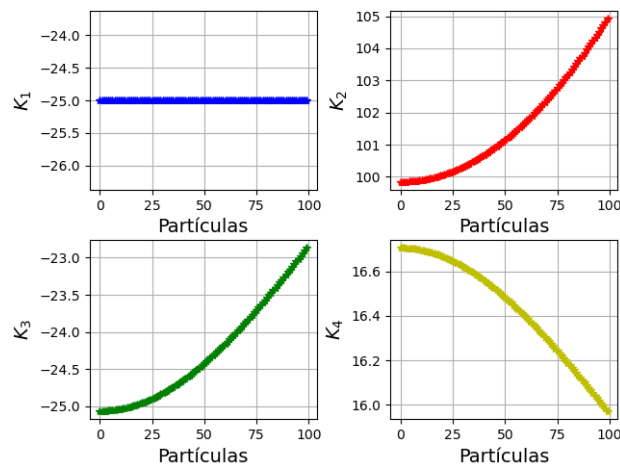
Tabela 5- Dados do Sistema para População C

Variável	Valor	Unidade
M_c, M_p	3.3113, 0.915	$[kg]$
l_p, r_{pm}	0.7474, $9.98 \cdot 10^{-3}$	$[m]$
J_p	0.02908	$[kg.m^2]$
J_m	$1.3819 \cdot 10^{-6}$	$[kg.m^2]$
B_{eq}, B_p	6.422, 0.00252	$[N.m/m^2]$
r	1.352	$[ohm]$
k_t	0.00735	$[N.m]$
k_m	0.0127	$[V/(rad/s)]$
η_g, η_m, k_g	0.88, 0.593, 2.82	-

Fonte: Autor Próprio

Os valores da tabela 5 foram selecionados aleatoriamente com uma variação entre um quarto do parâmetro real e o quádruplo do parâmetro real, utilizando a uma distribuição uniforme em um intervalo máximo e mínimo.

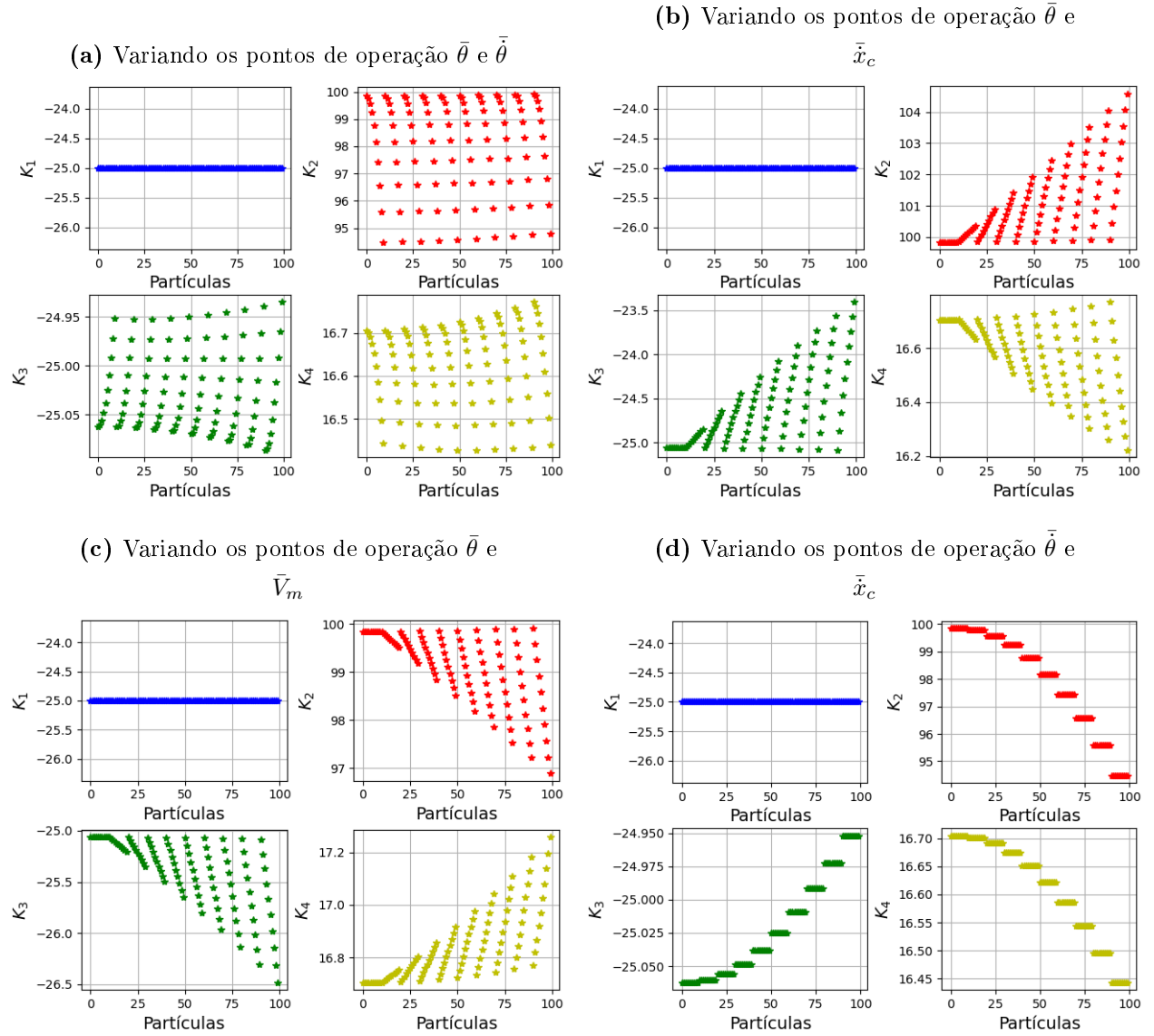
Para gerar diferentes membros nas populações B e C serão variados os pontos de operação no qual o sistema será linearizado ao redor e calcular os vetores de controle. Para isso é necessário combinar os estados e os valores da tabela 4 e avaliar qual combinação oferecerá maior diversidade entre os membros das populações. Inicialmente são combinados de modo crescente todos os valores da tabela 4 e gerado a primeira opção da população B, como apresentado na Figura 18.

Figura 18- Combinação de Todos os Pontos de Operação

Fonte: Autor Próprio

Em seguida são combinados dois estados e variando seus valores de acordo com a tabela 4 em 10 valores igualmente espaçados para cada estado, essas combinações podem ser observadas na Figura 19.

Figura 19- Possíveis populações B



Fonte: Autor Próprio

Para avaliar a diversidade das possíveis populações foi analisado o desvio padrão ao quadrado dos membros de cada candidato a população B no seu respectivo espaço de busca, obtendo quatro desvios padrões ao quadrado para cada possível população B, como é apresentado na tabela 6.

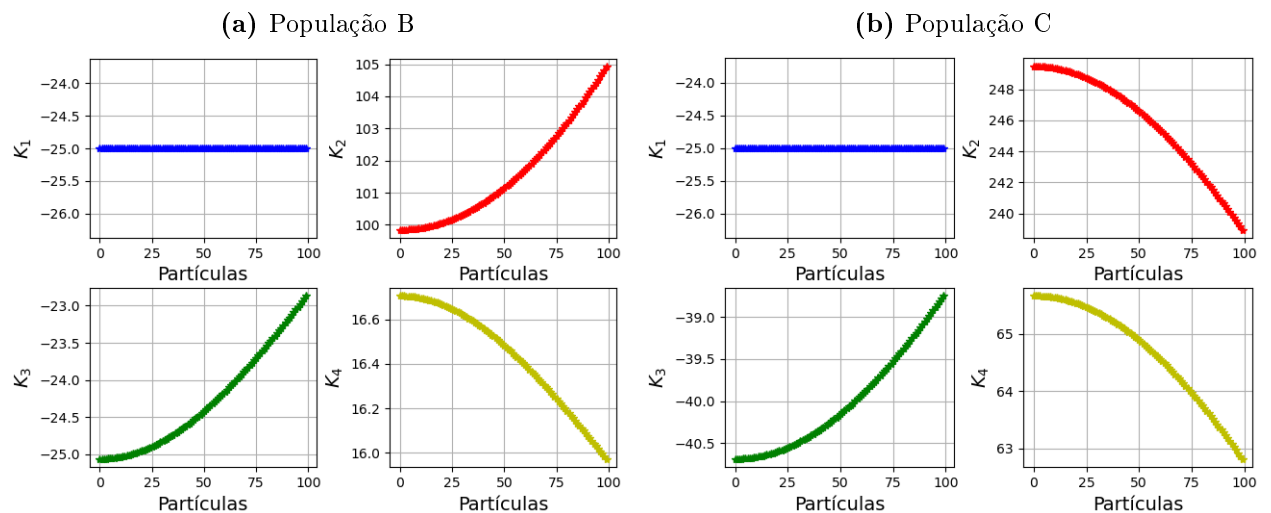
Tabela 6- Tabela do Desvio Padrão ao quadrado das Possíveis Populações B

	σ^2			
	K_1	K_2	K_3	K_4
Combinação de Todos os Pontos de Operação	0	2.3686	0.4624	0.0523
Variando os pontos de operação $\bar{\theta}$ e $\bar{\theta}$	0	3.057	0.0017	0.0096
Variando os pontos de operação $\bar{\theta}$ e $\bar{x}_c$	0	1.3729	0.1817	0.0167
Variando os ponto de operação $\bar{\theta}$ e $\bar{V}_m$	0	0.5362	0.1163	0.0171
Variando os pontos de operação $\bar{\theta}$ e $\bar{x}_c$	0	3.1952	0.0013	0.0076

Fonte: Autor Próprio

De acordo com a tabela 6 é possível observar que a combinação de todos os pontos de operação oferece um maior desvio padrão ao quadrado nos espaços de busca K_3 e K_4 e é um pouco menor quando comparado no espaço K_2 , por esses motivos esse candidato foi selecionado para a população B. De tal modo, para a população C onde os parâmetros se diferem do sistema a ser controla sua população será gerada combinando todos os pontos de operação de modo crescente, assim, as populações B e C são apresentadas na Figura 20. Todos os códigos e funções são encontrados no Apêndice A

Figura 20- Populações B e C



Fonte: Autor Próprio

6.4.3 População D e E

Semelhante as populações B e C, as populações D e E serão geradas utilizando os dados do sistema a ser controlado, apresentado na tabela 2, e outra utilizando dados diferentes do sistema a ser controlado, como apresentado na tabela 5, respectivamente. Entretanto, para essas populações será utilizado a técnica LQR formulada em LMI para a inserção de incertezas nos parâmetros de coeficientes de atrito B_{eq} e B_p . Sendo assim, inicialmente é contruído a matriz

$$\mathbf{p} = \begin{bmatrix} B_p & B_{eq} \end{bmatrix} \quad (62)$$

onde $B_p \in [B_{p,min}, B_{p,max}]$ e $B_{eq} \in [B_{eq,min}, B_{eq,max}]$. Os máximos e mínimos das incertezas foram calculadas por meio de porcentagens, para cada coeficiente de atrito (63) e (64) foram utilizadas para obter os máximos e mínimos.

$$\begin{aligned} B_{p,min} &= \left(1 - \frac{incerteza_{B_p}}{100}\right) B_p \\ B_{p,max} &= \left(1 + \frac{incerteza_{B_p}}{100}\right) B_p \end{aligned} \quad (63)$$

$$\begin{aligned} B_{eq,min} &= \left(1 - \frac{incerteza_{B_{eq}}}{100}\right) B_{eq} \\ B_{eq,max} &= \left(1 + \frac{incerteza_{B_{eq}}}{100}\right) B_{eq} \end{aligned} \quad (64)$$

O politopo é então definido com $N = 2^2$ vértices contendo os possíveis valores de incertezas. Em seguida, o sistema a ser controlado foi linearizado ao redor dos pontos $\bar{\mathbf{x}}(t) = [\bar{x}_1 \quad \bar{x}_2 \quad \bar{x}_3 \quad \bar{x}_4]' = [0 \quad 0 \quad 0 \quad 0]'$ e $\bar{u} = 0$, resultando nas matrizes $\mathbf{A}$ e $\mathbf{B}$ em (65) e (66).

$$\mathbf{A} = \frac{1}{J_T} \begin{bmatrix} 0 & 0 & J_T & 0 \\ 0 & 0 & 0 & J_T \\ 0 & M_p^2 l_p^2 g & -B_{eq}(M_p l_p^2 + J_p) & -B_p M_p l_p \\ 0 & M_p l_p g (M_{eq} + M_p) & -B_{eq} M_p l_p & -B_p (M_{eq} + M_p) \end{bmatrix} \quad (65)$$

$$\mathbf{B} = \frac{A_{eq}}{J_T} \begin{bmatrix} 0 \\ 0 \\ M_p l_p^2 + J_p \\ M_p l_p \end{bmatrix} \quad (66)$$

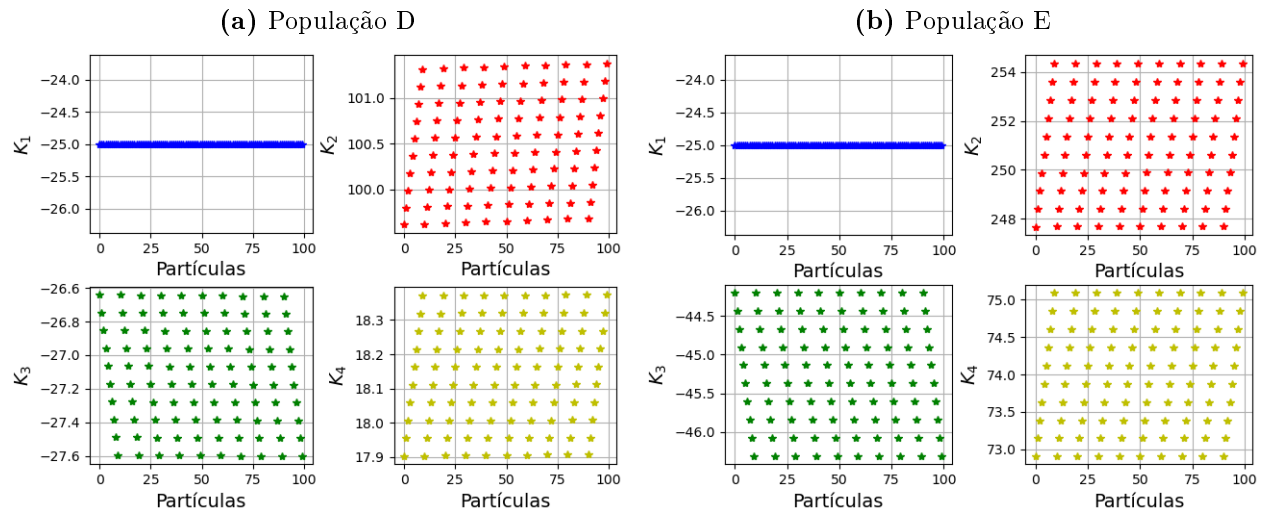
Sendo $J_T = (M_p + M_{eq})(M_p l_p^2 + J_p)$. Foi verificado que em (65) e (66), há uma

dependência linear dos parâmetros B_p e B_{eq} , logo, os vértices do modelo politópico são representados em (67) e (68).

$$\begin{aligned}
 A_1 &= \frac{1}{J_T} \begin{bmatrix} 0 & 0 & J_T & 0 \\ 0 & 0 & 0 & J_T \\ 0 & M_p^2 l_p^2 g & -B_{eq,min} (M_p l_p^2 + J_p) & -B_{p,min} M_p l_p \\ 0 & M_p l_p g (M_{eq} + M_p) & -B_{eq,min} M_p l_p & -B_{p,min} (M_{eq} + M_p) \end{bmatrix} \\
 A_2 &= \frac{1}{J_T} \begin{bmatrix} 0 & 0 & J_T & 0 \\ 0 & 0 & 0 & J_T \\ 0 & M_p^2 l_p^2 g & -B_{eq,max} (M_p l_p^2 + J_p) & -B_{p,max} M_p l_p \\ 0 & M_p l_p g (M_{eq} + M_p) & -B_{eq,max} M_p l_p & -B_{p,max} (M_{eq} + M_p) \end{bmatrix} \\
 A_3 &= \frac{1}{J_T} \begin{bmatrix} 0 & 0 & J_T & 0 \\ 0 & 0 & 0 & J_T \\ 0 & M_p^2 l_p^2 g & -B_{eq,max} (M_p l_p^2 + J_p) & -B_{p,min} M_p l_p \\ 0 & M_p l_p g (M_{eq} + M_p) & -B_{eq,max} M_p l_p & -B_{p,min} (M_{eq} + M_p) \end{bmatrix} \\
 A_4 &= \frac{1}{J_T} \begin{bmatrix} 0 & 0 & J_T & 0 \\ 0 & 0 & 0 & J_T \\ 0 & M_p^2 l_p^2 g & -B_{eq,min} (M_p l_p^2 + J_p) & -B_{p,max} M_p l_p \\ 0 & M_p l_p g (M_{eq} + M_p) & -B_{eq,min} M_p l_p & -B_{p,max} (M_{eq} + M_p) \end{bmatrix}
 \end{aligned} \tag{67}$$

$$B_{1,2,3,4} = \frac{A_{eq}}{J_T} \begin{bmatrix} 0 \\ 0 \\ M_p l_p^2 + J_p \\ M_p l_p \end{bmatrix} \tag{68}$$

Com o modelo politópico definido, as incertezas para os coeficientes de atrito B_{eq} e B_p são variadas de 0% a 10% e as populações D e E são geradas e representadas na Figura 21.

Figura 21- Populações D e E

Fonte: Autor Próprio

7 RESULTADOS

Nesta seção será apresentado os resultados do controle realizado pelo algoritmo PSO para cada população diante as três situações onde os parâmetros de massa, atrito e comprimento foram alterados. Também será analisado o desempenho dos melhores ganhos durante as iterações por meio dos índices de custo em (53), o consumo energético (55) e também a função de erro quadrado integral (56). Por fim, as melhores partículas serão simuladas e avaliadas na situação onde o sistema não sofreu nenhuma alteração, para obter uma melhor visualização da otimização ocorrida no algoritmo, e será apresentado a progressão da populações nos espaços de busca em relação ao custo para cada situação, com intuito de melhor visualizar a direção que cada população. Os códigos utilizados para as simulações podem ser encontrados no Apêndice A

7.1 População A

Nas tabelas 7, 8 e 9 é visto os melhores ganhos nas iterações inicial e final e nas iterações 50, 100, 150 e 200 onde ocorreram as situações de mudança impostas ao algoritmo, como mudança na massa, comprimento e nos atritos. Observando essas tabelas é visto que dependendo da mudança nos parâmetros a variação do vetor de controle possui um comportamento diferente, pois o algoritmo PSO buscou encontrar o vetor de controle que melhor se encaixa nas necessidades do sistema atual, ou seja, reduz o custo do melhor vetor de controle. Nos espaços de busca K_1 e K_3 as posições encontradas por todas as mudanças se encontram em regiões diferentes uma das outras, já no espaço de busca K_2 as posições encontradas pela mudança de massa e comprimento estão em regiões próximas, já no espaço de busca K_4 as posições encontradas pela mudança de comprimento e atrito estão em regiões próximas.

Tabela 7- Tabela das Melhores Posições da População A - Mudança na Massa

Iteração	K_1	K_2	K_3	K_4
0	-426.6905	719.0405	-430.569	362.9441
50	-290.3496	875.8394	-379.5744	331.0451
100	-259.3638	892.6250	-373.1282	309.9967
150	-217.3047	925.6347	-383.8545	292.9424
200	-156.4284	951.0764	-379.9921	304.1034
300	-147.8237	943.7808	-359.5983	260.5135

Fonte: Autor Próprio

Tabela 8- Tabela das Melhores Posições da População A - Mudança na Comprimento

Iteração	K_1	K_2	K_3	K_4
0	-426.6905	719.0405	-430.569	362.9441
50	-203.7508	868.6854	-322.3858	303.4742
100	-123.4844	879.0079	-314.9286	302.3338
150	-96.6084	944.3706	-296.6768	277.5118
200	-74.2878	971.2025	-299.1883	281.8791
300	-68.9455	975.6349	-268.9234	304.3631

Fonte: Autor Próprio

Tabela 9- Tabela das Melhores Posições da População A - Mudança na Atrito

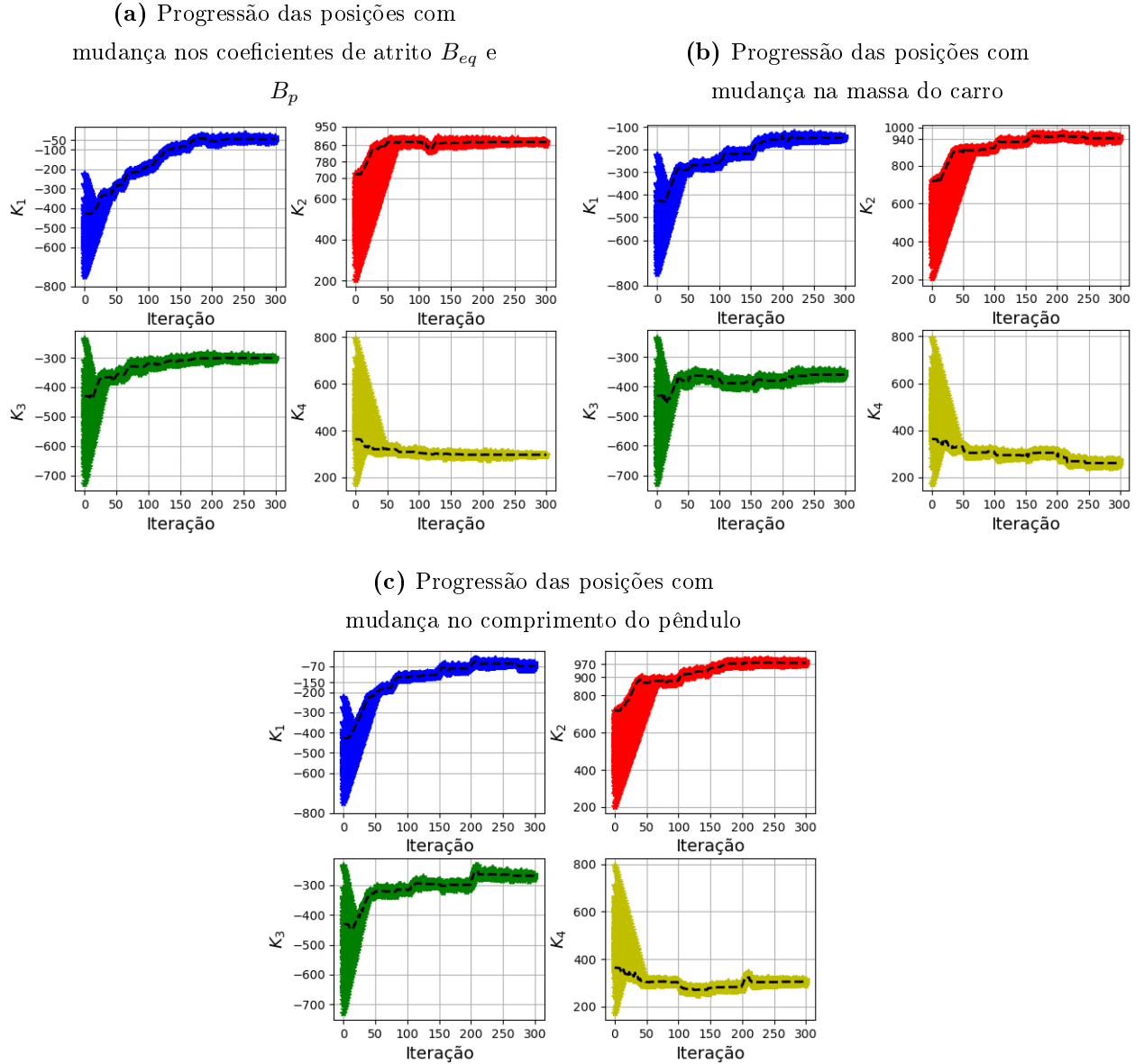
Iteração	K_1	K_2	K_3	K_4
0	-426.6905	719.0405	-430.569	362.9441
50	-290.3504	861.9132	-362.5685	320.2296
100	-183.6627	875.9956	-320.8498	304.7830
150	-87.7941	873.6713	-310.1429	299.6632
200	-53.8918	873.8554	-301.7481	296.2875
300	-45.1829	876.1521	-300.6437	296.7932

Fonte: Autor Próprio

Na Figura 22 é apresetando a progressão das posições durante o funcionamento do algoritmo. Devido a população A ser de origem aleatória com uma distribuição normal gaussiana, é visto em todos os casos que o algoritmo dedica algumas iterações iniciais para

estabilizar as posições das partículas ao redor da melhor posição ou uma posição estável. Analisando os gráficos da Figura 22a, é perceptível que a progressão das posições não foi muito influenciada pelas mudanças no atrito do sistema, já nos gráficos das Figuras 22b e 22c as posições das partículas possuíram uma resposta mais expressiva as mudanças de massa e comprimento do sistema.

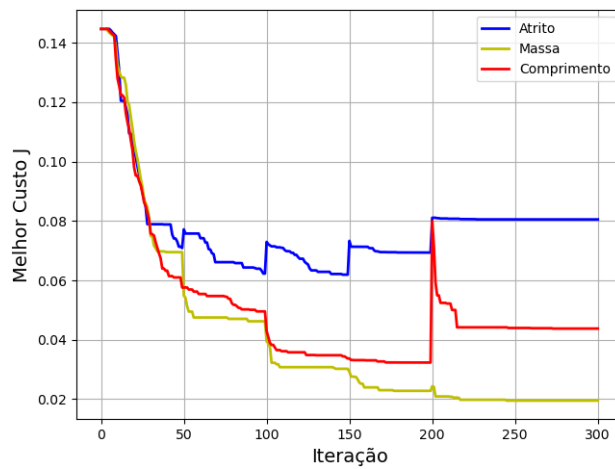
Figura 22- Progressão das posições da população A



Na progressão das posições desta população é difícil observar a influência das mudanças dos parâmetros e como isso afetou o desempenho do vetor de controle apenas analisado os gráficos da Figura 22, deste modo, é apresentado na Figura 23 a progressão do custo das melhores partículas para os três casos. Assim, fica mais claro a influência

das mudanças nos parâmetros, para a mudança do atrito é visto que na primeira mudança o custo da população aumentou, mas logo foi reduzido, este comportamento é observado anterior a última mudança na qual estabilizou em custo maior que o mínimo obtido neste caso. Para o caso da mudança de massa, é visto que o custo foi melhorado conforme o parâmetro foi alterado devido a redução do peso. Por fim, para a mudança do comprimento inicial do comprimento o custo foi reduzido e no final quando o comprimento foi reduzido novamente o custo foi elevado e logo reduzido.

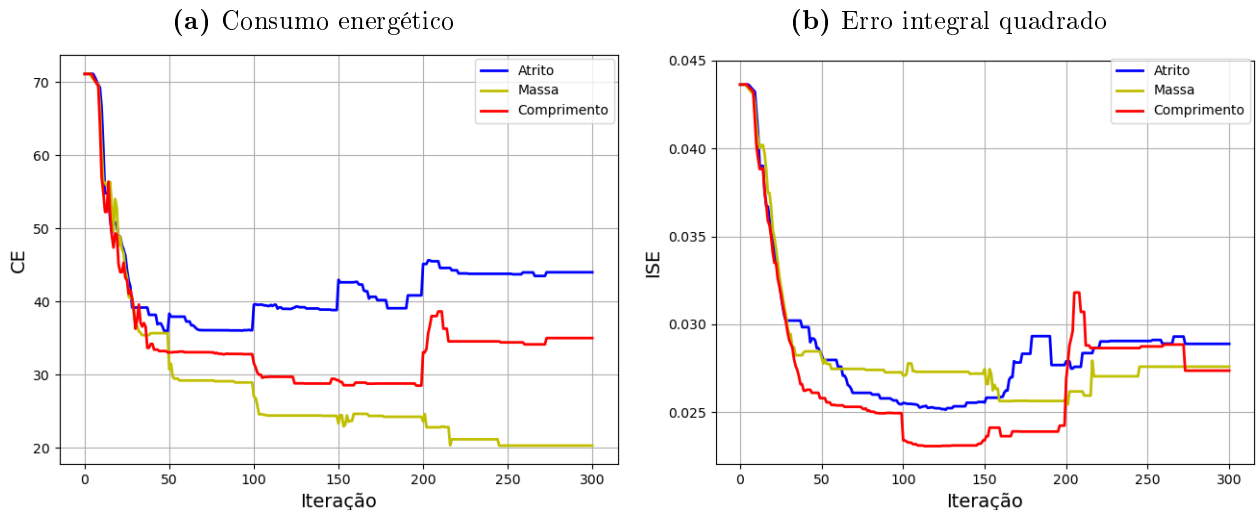
Figura 23- Custo das melhores posições da população A para todos os casos de mudanças de parâmetros



Fonte: Autor Próprio

Na Figura 24 são apresentados o consumo energético, calculado por meio de (55), e o erro integral quadrado, calculado por meio de (56), pois são as duas parcelas utilizadas em (53).

Figura 24- CE e ISE das melhores posições da população A

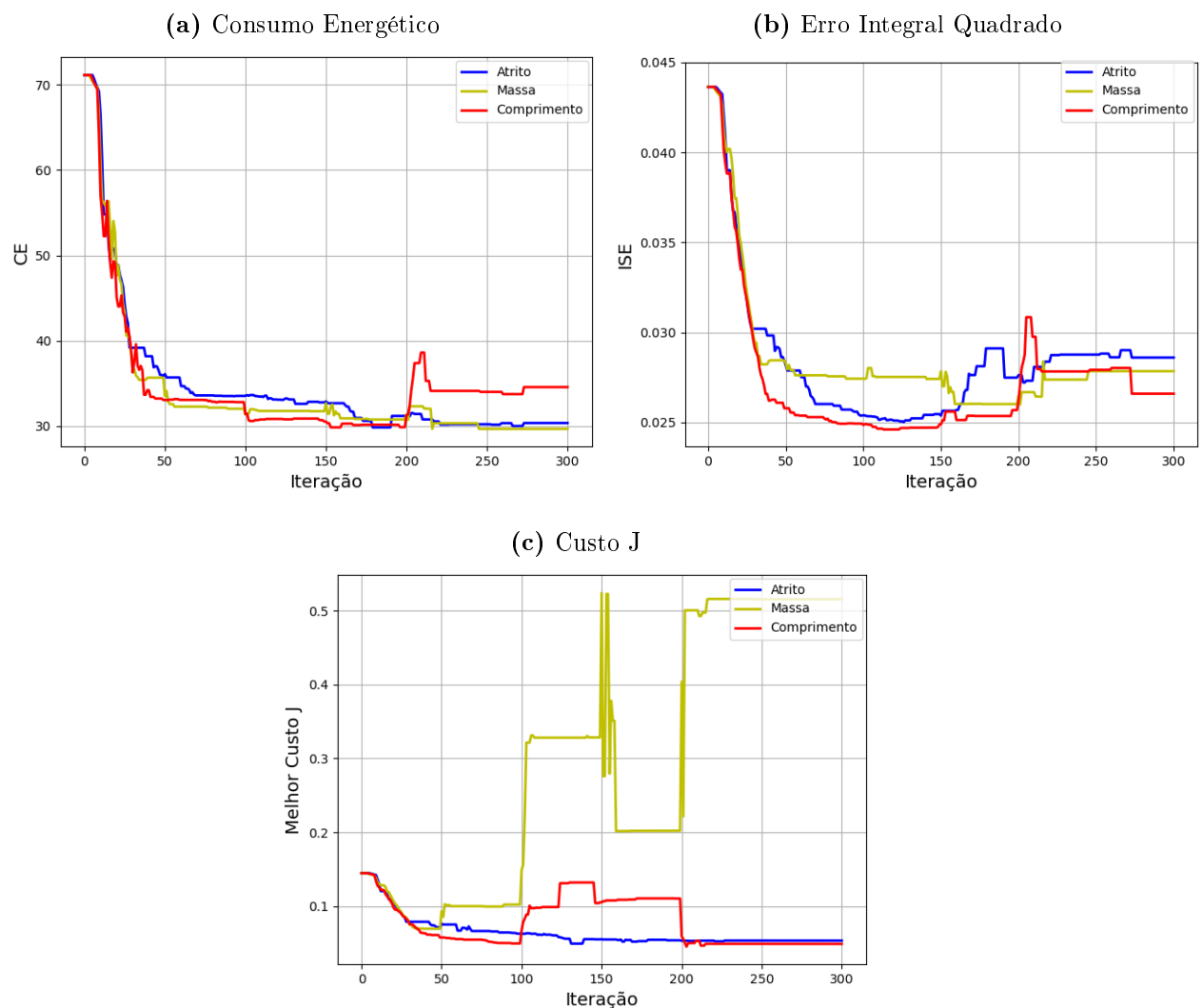


Fonte: Autor Próprio

Deste modo, é possível observar que o consumo energético foi um fator determinante em relação ao valor de erro, visto que, quanto menor a massa do sistema, menos força do atuador é requerida, diferente das outras mudanças. No caso do atrito, o sistema sofreu dificuldades tanto na movimentação do carro quanto na rotação do pêndulo. No caso da redução do comprimento do pêndulo, quando o comprimento foi reduzido muito, esta mudança se tornou um fator dificultador no controle. Assim, em ambos os casos é requerido mais energia do atuador para estabilizar o pêndulo invertido. É visto nos gráficos das Figuras 24a e 24b que ao final da simulação o algoritmo encontrou uma resposta melhor do que a resposta da população inicial.

Em uma última análise para esta simulação, os vetores de ganhos encontrados por cada simulação serão colocados em uma situação na qual nenhum parâmetro do pêndulo invertido foi alterado.

Figura 25- Situação sem mudanças das melhores posições da população A



Fonte: Autor Próprio

O algoritmo adaptou a população A para cada situação onde algum parâmetro foi alterado e quando comparado entre si, a adaptação na mudança de massa se destacou entre as outras. Entretanto, nos gráficos na Figura 25 é visto que as posições das partículas encontradas por meio das mudanças de atrito mantiveram sua consistência de otimização, já na mudança de comprimento o algoritmo precisou de mais algumas iterações. Mas as posições das partículas encontradas pela mudança de massa obtiveram um custo maior para o sistema saudável, mesmo tendo um consumo energético igual ou menor as demais posições das outras mudanças, assim, é visto que devido as matrizes de ponderação $\mathbf{Q}$ e $\mathbf{R}$ em (53), o custo J das posições na situação onde a massa foi alterada apresenta um valor elevado. Deste modo é possível afirmar que as posições encontradas pela mudança de massa possuem um erro de referência da posição do carro e do pêndulo maior que as outras posições encontradas pelas mudanças de comprimento e atrito.

7.2 População B

Diferente da população A, que possui origem aleatória, a população B possui origem ótima gerada pela técnica LQR.

Nas tabelas 10, 11 e 12 é visto os melhores ganhos nas iterações inicial e final e nas iterações 50, 100, 150 e 200 onde ocorreram as situações de mudança impostas ao algoritmo, como mudança na massa, comprimento e nos atritos. Nessas tabelas é visto que nos espaços de busca K_1 e K_2 as melhores posições encontradas pela mudança de massa se concentram em regiões diferentes das demais posições encontradas pelas mudanças de comprimento e atrito, já nos espaços K_3 e K_4 as posições encontradas por todas as mudanças se concentram em regiões próximas.

Tabela 10- Tabela das Melhores Posições da População B - Mudança na Massa

Iteração	K_1	K_2	K_3	K_4
0	-25	99.8443	-25.0613	16.7046
50	-23.1760	349.5524	-105.7304	126.3833
100	-29.9417	418.5012	-118.0092	128.1646
150	-46.1553	668.7442	-171.9724	173.1355
200	-59.2225	694.7088	-182.9658	164.7884
300	-53.2573	751.1722	-186.0923	156.5013

Fonte: Autor Próprio

Tabela 11- Tabela das Melhores Posições da População B - Mudança na Comprimento

Iteração	K_1	K_2	K_3	K_4
0	-25	99.8443	-25.0613	16.7046
50	-29.0752	421.6553	-123.2743	144.5365
100	-19.5004	425.8958	-127.4567	160.1404
150	-29.3991	410.5965	-120.8612	127.8863
200	-28.6631	410.7527	-122.2682	128.4824
300	-28.3007	389.4558	-122.8298	146.1307

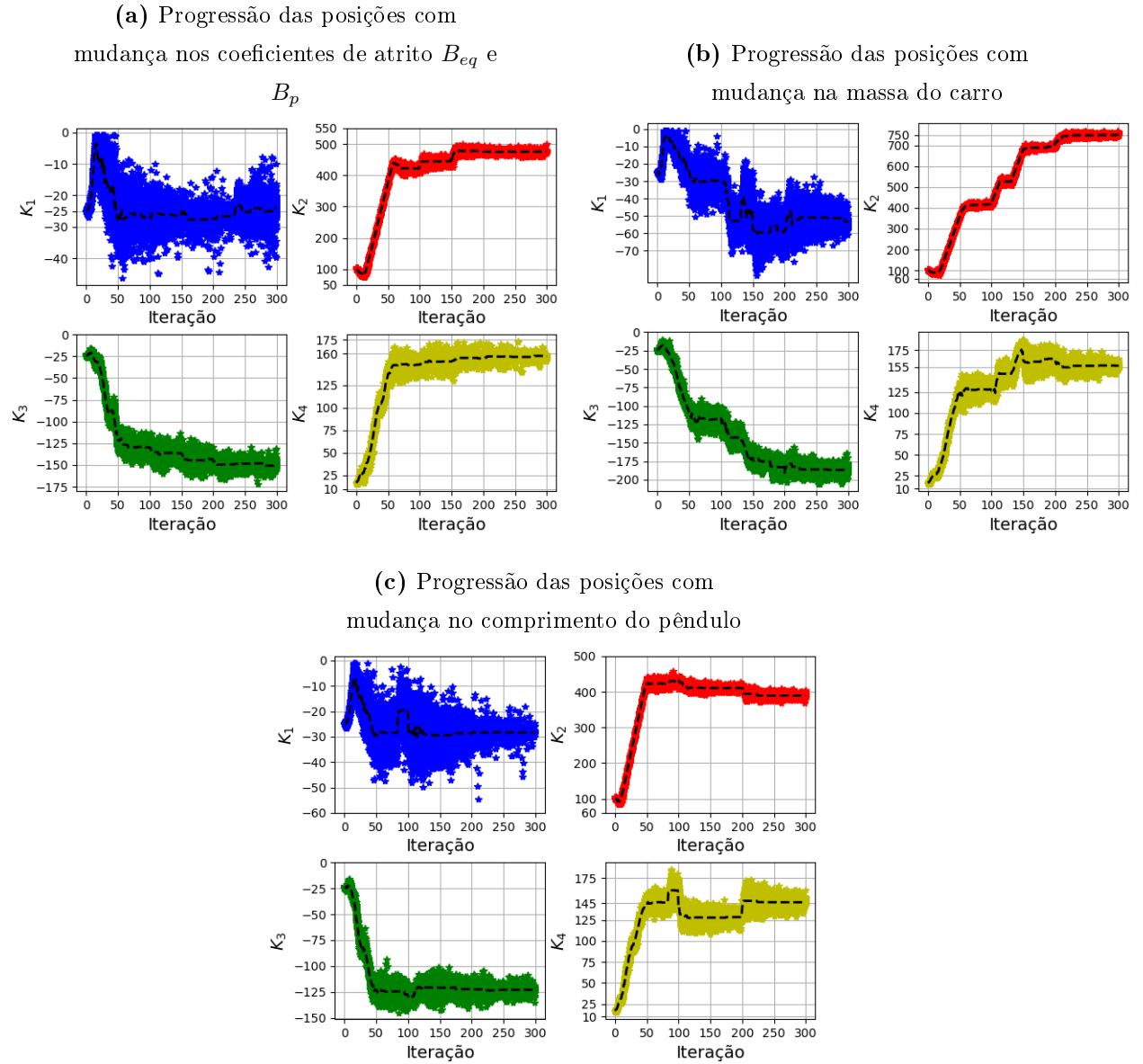
Fonte: Autor Próprio

Tabela 12- Tabela das Melhores Posições da População B - Mudança na Atrito

Iteração	K_1	K_2	K_3	K_4
0	-25	99.8443	-25.0613	16.7046
50	-27.6352	368.0884	-118.7799	133.1838
100	-26.6401	421.8108	-129.5063	147.6634
150	-26.3767	444.7602	-136.4339	151.1304
200	-27.2103	477.8366	-143.1962	154.4712
300	-24.0595	475.4941	-149.5569	156.7479

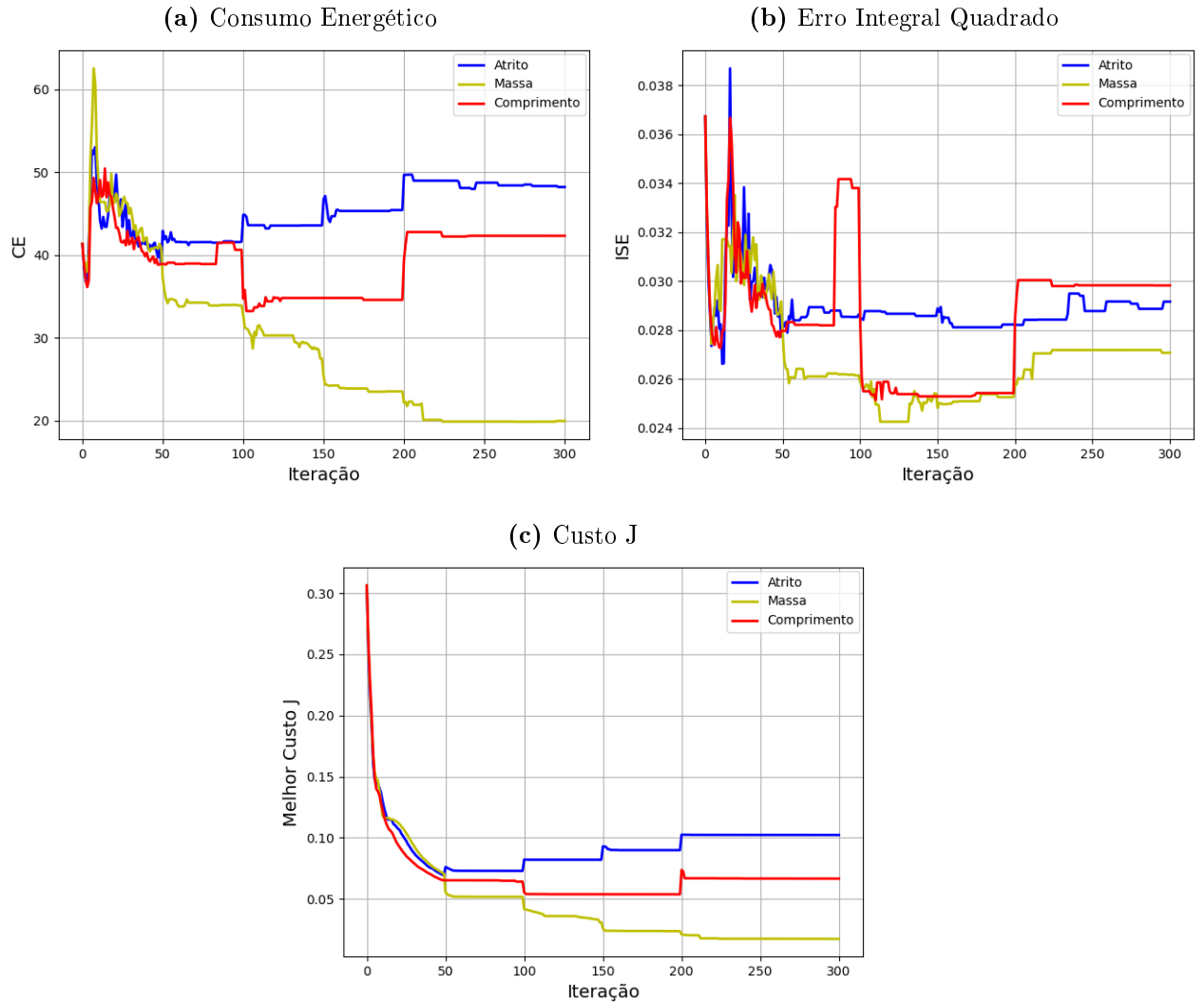
Fonte: Autor Próprio

Na Figura 26 é apresentado a progressão das posições das partículas da população B durante o funcionamento do algoritmo para as situações de mudanças de parâmetros. Nesses gráficos é visto um comportamento diferente em cada espaço de busca para cada mudança, no espaço de busca K_1 as posições das partículas foram mais sensíveis a mudança de massa, no espaço de busca K_2 as posições foram bem sensíveis as mudanças, no espaço K_3 apenas para a mudança no comprimento as posições das partículas não se alteraram muito e por fim no espaço K_4 apenas na mudança de comprimento as posições foram sensíveis.

Figura 26- Progressão das posições da população B

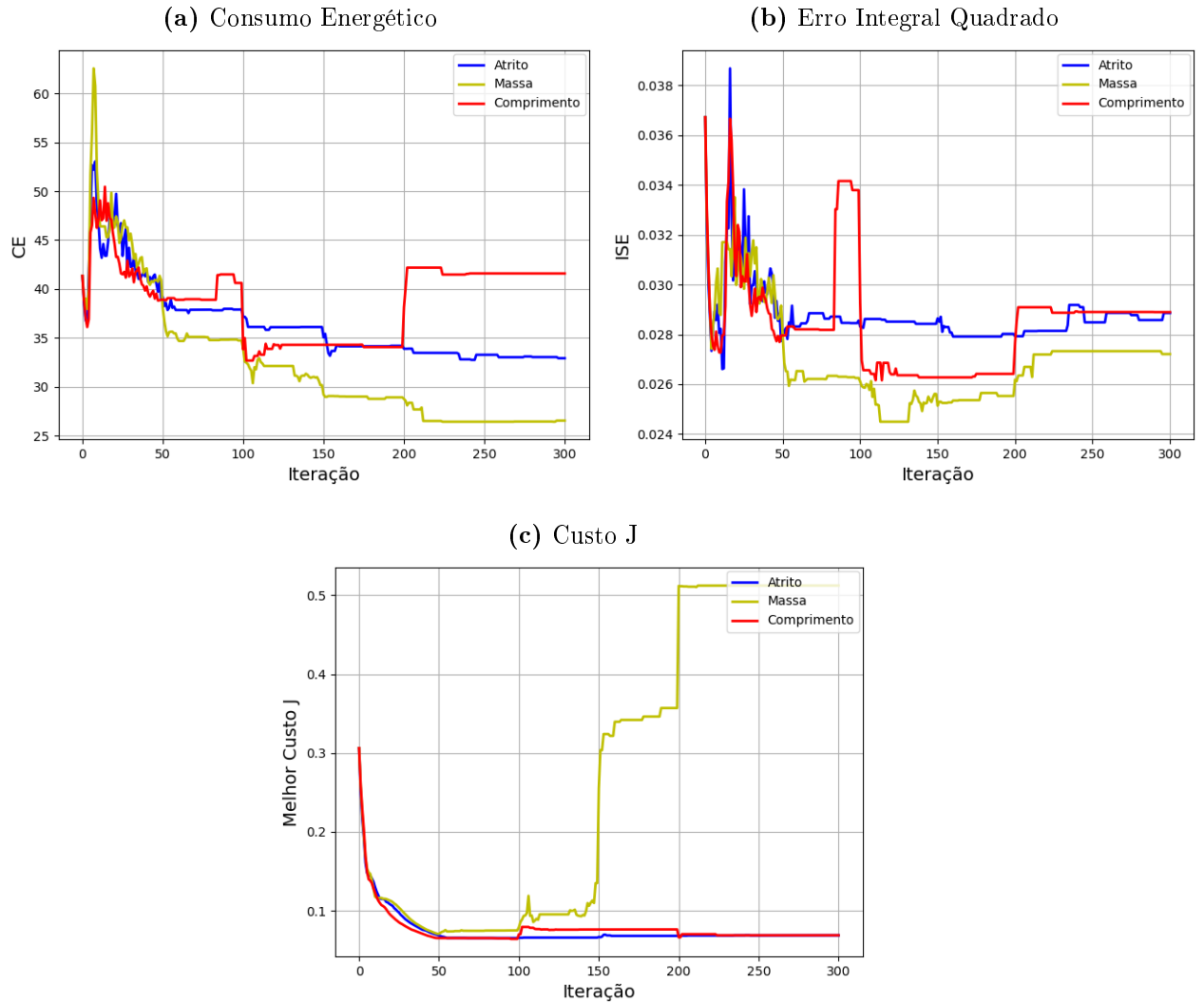
Fonte: Autor Próprio

Melhor que na população A, é possível observar melhor a influência de cada mudança no vetor de controle, contudo, para uma análise mais sensível são analisados a função de custo (53) e seus componentes consumo energético (55) e erro quadrado integral (56) das melhores posições das partículas.

Figura 27- Análise das melhores posições da população B

Fonte: Autor Próprio

Na Figura 30 é observado três gráficos, no gráfico de consumo energético é visto uma semelhança no comportamento quando comparado ao gráfico da Figura 24a, as mudanças de atrito exigem do sistema um consumo maior do atuador. No gráfico de erro quadrado, Figura 27b, é visto que os valores possuem uma pequena variação entre eles e que quando estes valores são combinados com os valores de consumo energético é obtido o custo J . Na Figura 27c, esse gráfico mostra bem em todos os casos de mudança como o algoritmo enxergou em qual direção as partículas deveriam se movimentar, pois, diferente do gráfico da Figura 23, as partículas necessitaram um tempo para compreender a situação na qual precisavam seguir. A seguir, as melhores posições encontradas para cada mudança são simuladas para uma situação do sistema sem mudanças para avaliar o desempenho de cada ganho encontrado por meio da função de custo (53) e seus componentes consumo enegético (55) e erro quadrado integral (56).

Figura 28- Situação sem mudanças das melhores posições da população B

Fonte: Autor Próprio

Semelhante a situação na população A, as posições das partículas na população B adaptadas a mudança de massa apresentam um custo J mais elevado devido as matrizes de ponderação. O comportamento do consumo energético, no caso das posições adaptadas para mudança de comprimento obtiveram um resultado maior do que o valor inicial e quando comparado com os resultados da população A, Figura 25a, a população B apresentou uma divergência final maior quando comparado aos resultados e para o erro quadrado integral seus valores se diferem pouco.

7.3 População C

Diferente da população B, que possui origem ótima, a população C possui também origem ótima, mas foi gerada para um sistema diferente do sistema na qual o algoritmo

será aplicado.

Nas tabelas 13, 14 e 15 é visto os melhores ganhos nas iterações inicial e final e nas iterações 50, 100, 150 e 200 onde ocorreram as situações de mudança impostas ao algoritmo, como mudança na massa, comprimento e nos atritos. Nessas tabelas é visto que as melhores posições das partículas encontradas pela mudança de massa e comprimento se encontram, sua maioria, na mesma ou em regiões próximas, diferente das posições encontradas pela mudança de atrito.

Tabela 13- Tabela das Melhores Posições da População C - Mudança na Massa

Iteração	K_1	K_2	K_3	K_4
0	-25	249.4618	-40.686	65.6601
50	-29.6525	422.8984	-123.0399	145.1744
100	-56.8161	698.2851	-188.0218	194.7016
150	-54.4402	764.9466	-176.1310	184.7698
200	-63.3384	814.5532	-181.0431	173.1677
300	-69.6198	841.4713	-226.1091	202.8267

Fonte: Autor Próprio

Tabela 14- Tabela das Melhores Posições da População C - Mudança na Comprimento

Iteração	K_1	K_2	K_3	K_4
0	-25	249.4618	-40.686	65.6601
50	-28.8247	422.6005	-125.9094	146.3009
100	-51.0535	704.8712	-202.8614	202.9621
150	-62.1805	782.5580	-233.0660	211.5219
200	-71.0812	825.8695	-234.6671	197.9307
300	-66.0536	840.5673	-234.2443	206.2580

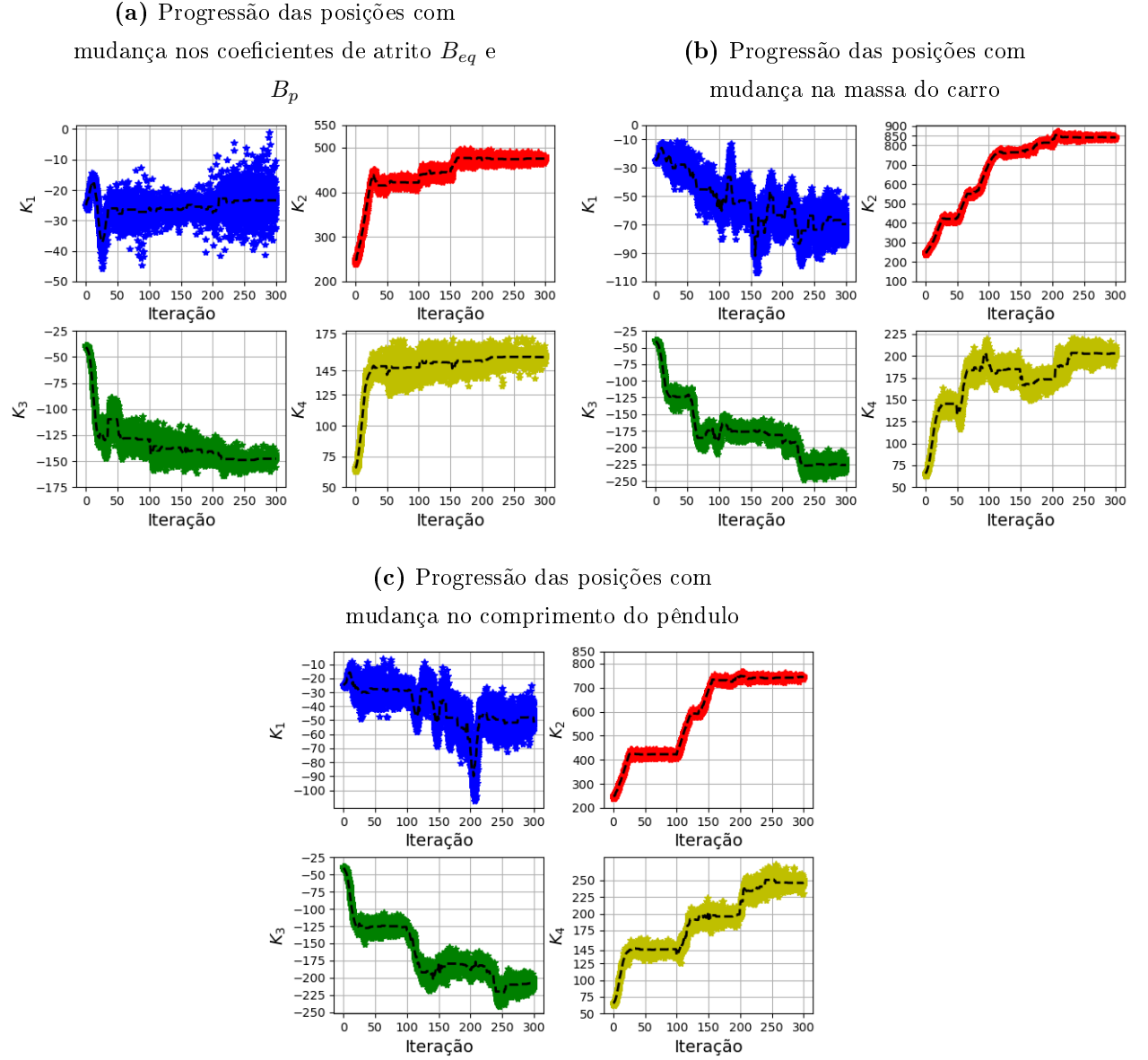
Fonte: Autor Próprio

Tabela 15- Tabela das Melhores Posições da População C - Mudança no Atrito

Iteração	K_1	K_2	K_3	K_4
0	-25	249.4618	-40.686	65.6601
50	-26.0211	415.1181	-109.7962	148.3902
100	-27.1825	421.6298	-129.2242	147.4629
150	-26.4607	443.9999	-137.0107	151.3688
200	-25.5147	477.0517	-140.6830	153.0674
300	-23.3953	475.1960	-147.7588	155.8475

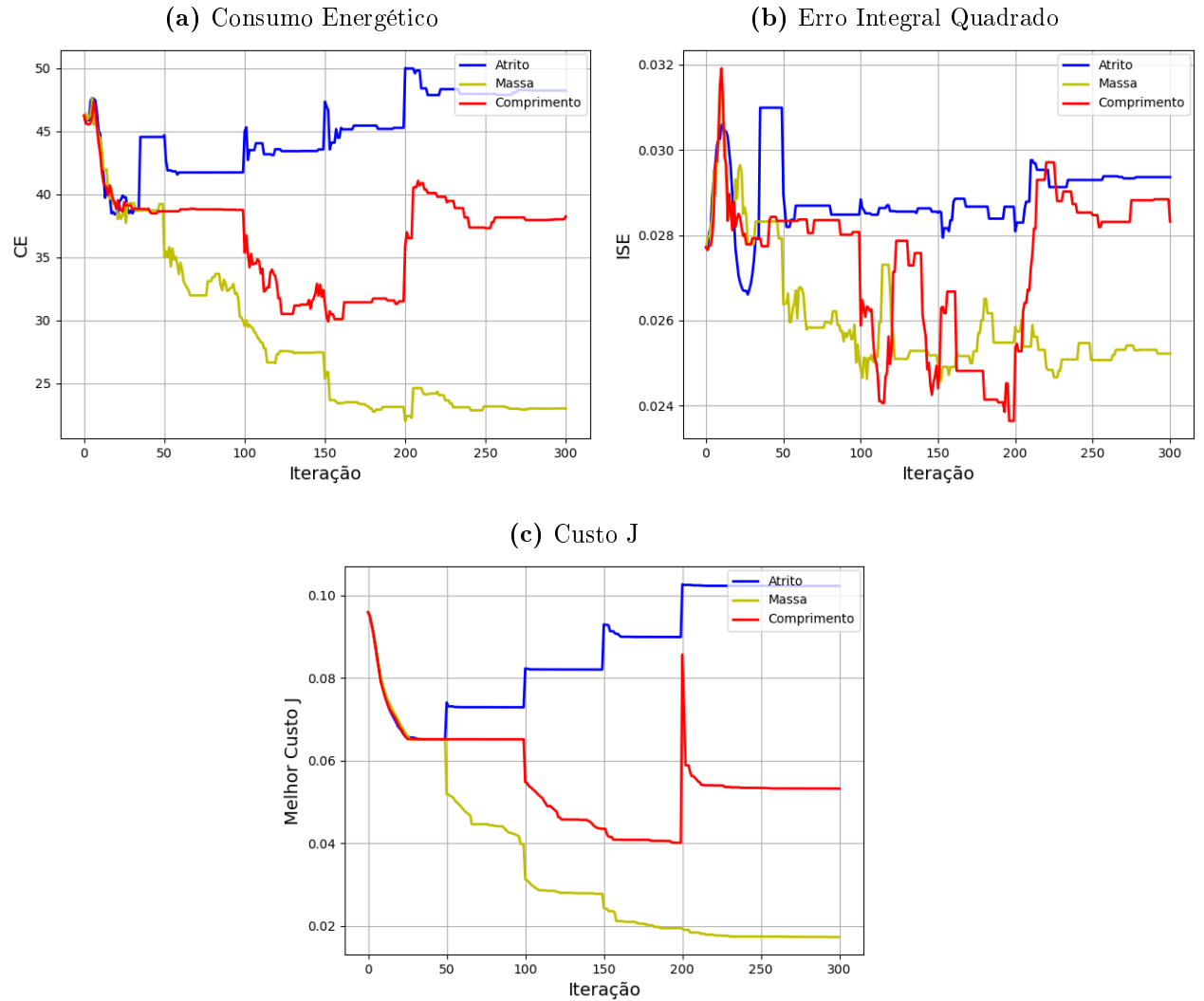
Fonte: Autor Próprio

Na Figura 29 é apresentado a progressão das posições das partículas da população C durante o funcionamento do algoritmo para as situações de mudanças de parâmetros.

Figura 29- Progressão das posições da população C

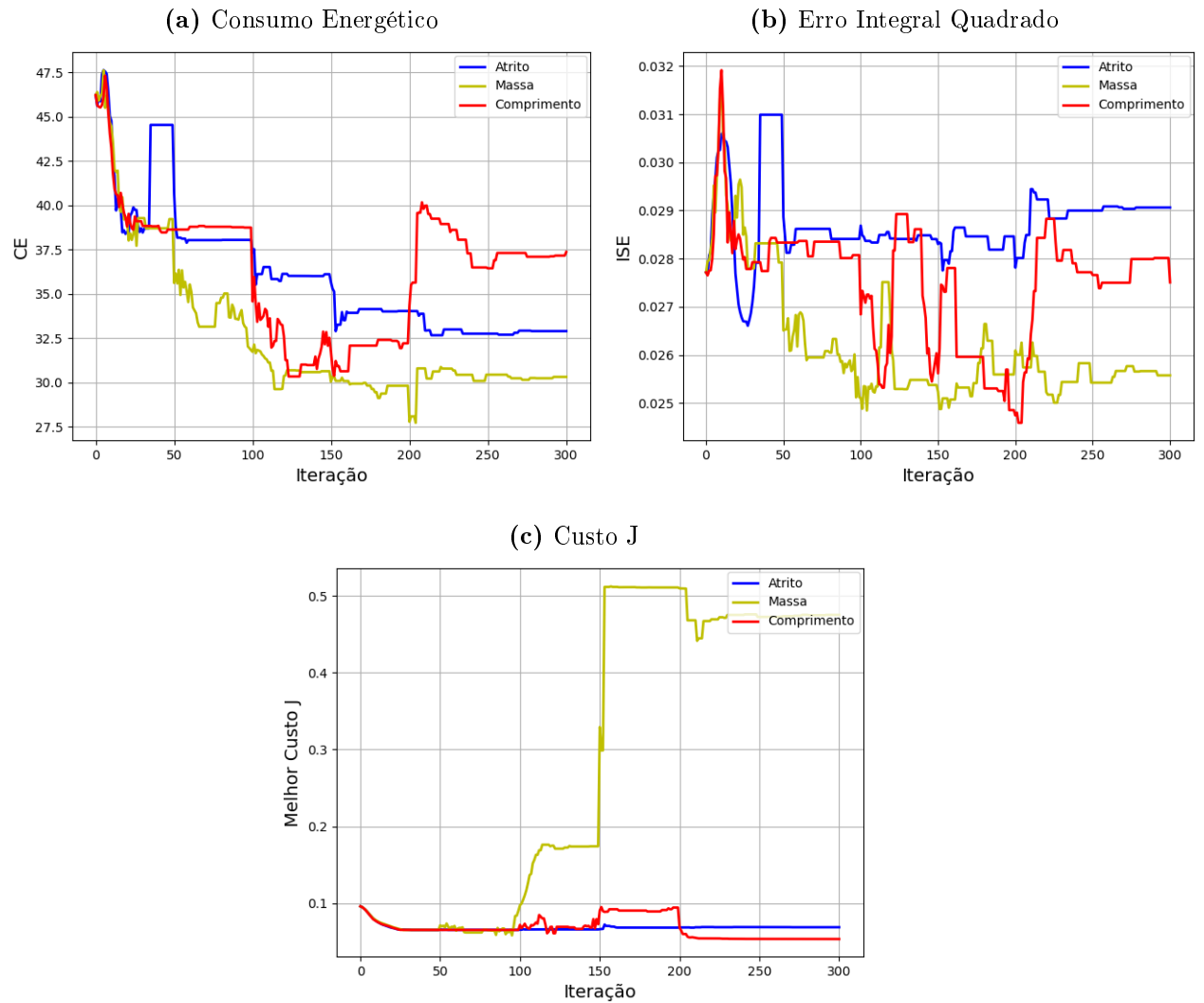
Fonte: Autor Próprio

Nesses gráficos, em apenas alguns espaços de busca apresentaram uma mudança significativa devido a mudança dos parâmetros. Por este motivo e para uma análise mais aprofundada das melhores posições das partículas será avaliado a função de custo (53) e seus componentes consumo energético (55) e erro quadrado integral (56) serão analisados para cada situação de mudança.

Figura 30- Análise das melhores posições da população C

Fonte: Autor Próprio

No gráfico da Figura 30c é possível observar um comportamento similar aos gráficos das Figuras 23 e 27c, onde o custo da população afetada pelo atrito foi o maior devido ao aumento da dificuldade de movimentação do sistema e a população afetada pela mudança de massa foi o menor. No caso do consumo energético, também segue o comportamento dos gráficos das Figuras 24a e 27a pelos mesmos motivos, o algoritmo adapta o comportamento da mudança de comprimento, mas a mudança no atrito exige mais força do atuador. Os resultados apresentados no gráfico da Figura 30b apresentam resultados com pequena diferença entre si. Assim, é simulado as posições partículas encontradas nas três situações para a situação do sistema sem alterações para avaliar seu comportamento.

Figura 31- Situação sem mudanças das melhores posições da população C

Fonte: Autor Próprio

No gráfico da Figura 31a é visto que o consumo energético das posições foram reduzidas em comparação ao seu valor inicial. Ademais o erro quadrado integral na Figura 31b apresenta uma diferença pequena entre as demais posições encontradas para cada mudança. No caso do gráfico da Figura 31c o custo das posições das partículas encontradas pela mudança de massa segue o mesmo padrão das demais populações, para as posições encontradas devido a mudança de atrito e comprimento seu custo possui uma diferença pequena de 0.05, pois, no quesito de utilizar menos energia do atuador é destacado as posições das partículas encontradas pela mudança no atrito, mas em questão de possuir o menor erro em relação a referência, as posições encontradas pela mudança no comprimento se sobressaem de tal modo que ao final da simulação do algoritmo obteve um custo menor, por mais pequena diferença que seja em relação as posições encontradas pela mudança no atrito.

7.4 População D

Nesta população foi utilizado o a técnica LQR formulada em LMI inserindo incertezas nos parâmetros de coeficiente de atrito.

Nas tabelas 16, 17 e 18 é visto os melhores ganhos nas iterações inicial e final e nas iterações 50, 100, 150 e 200 onde ocorreram as situações de mudança impostas ao algoritmo, como mudança na massa, comprimento e nos atritos. Ao analisar as tabelas apresentadas é visto que as melhores posições das partículas encontradas pela mudança de massa no espaço de busca K_1 e K_2 sofrem grandes alterações, diferente das demais posições encontradas pelas mudanças no comprimento e no atrito.

Tabela 16- Tabela das Melhores Posições da População D - Mudança na Massa

Iteração	K_1	K_2	K_3	K_4
0	-25	99.6136	-26.6411	17.9031
50	-22.2140	337.8131	-105.4808	123.7201
100	-30.1189	417.6811	-118.1652	128.2611
150	-38.9986	526.0438	-153.7531	151.5721
200	-50.8975	627.9189	-163.1996	152.2179
300	-56.9287	740.1532	-164.5276	146.1975

Fonte: Autor Próprio

Tabela 17- Tabela das Melhores Posições da População D - Mudança na Comprimento

Iteração	K_1	K_2	K_3	K_4
0	-25	99.6136	-26.6411	17.9031
50	-24.2636	349.4308	-112.6974	128.8806
100	-29.3078	423.3575	-126.4566	146.8277
150	-29.0626	410.8939	-124.3335	129.3205
200	-29.1208	410.6846	-120.6723	127.8442
300	-28.9318	417.2998	-131.0434	159.8303

Fonte: Autor Próprio

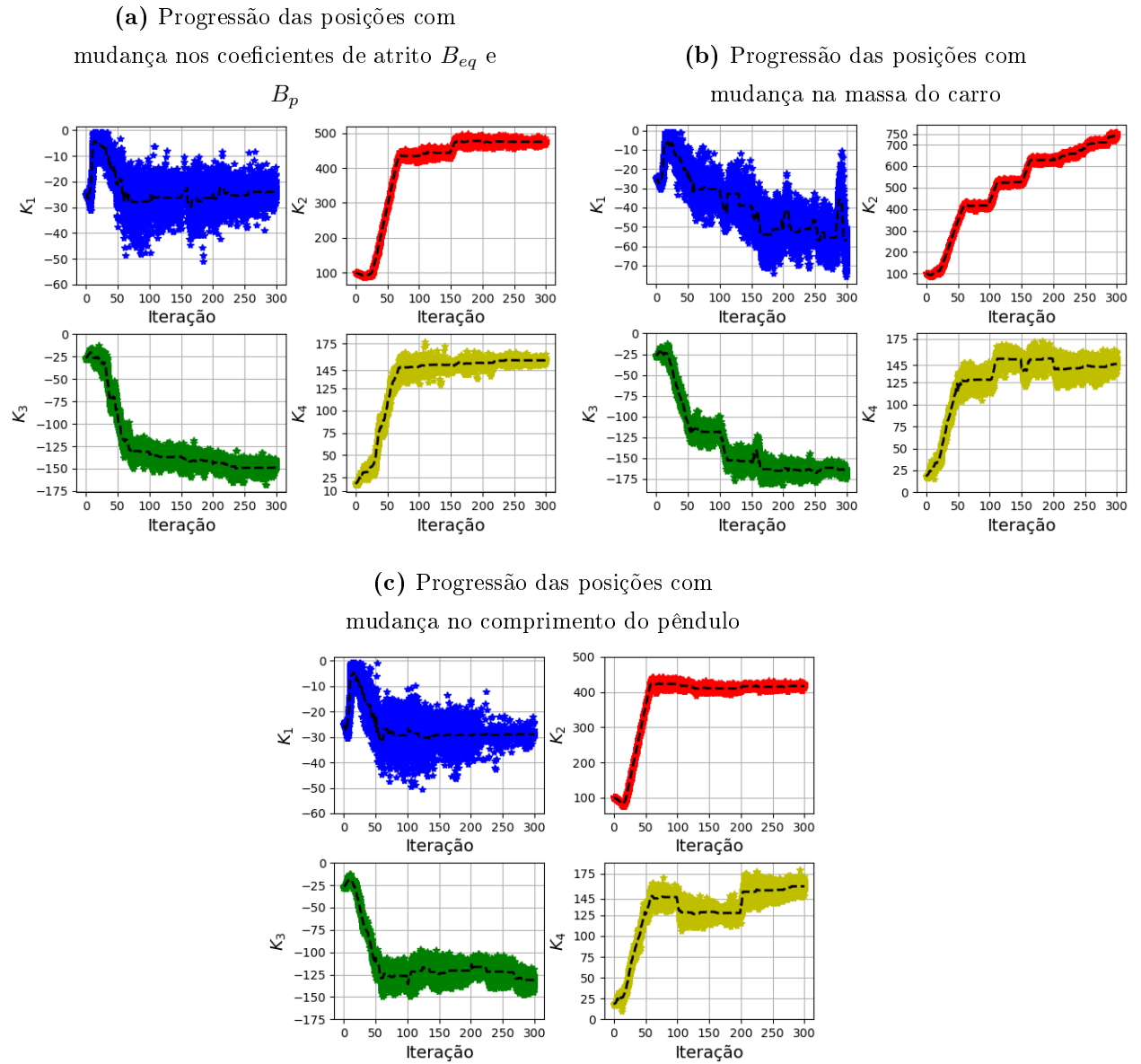
Tabela 18- Tabela das Melhores Posições da População D - Mudança no Atrito

Iteração	K_1	K_2	K_3	K_4
0	-25	99.6136	-26.6411	17.9031
50	-19.9668	280.9076	-83.8521	102.2277
100	-27.6705	434.1849	-131.0766	148.8737
150	-26.1893	443.5601	-136.6661	151.2219
200	-26.3156	477.4561	-141.3766	153.5294
300	-23.9870	475.0157	-148.9385	156.3218

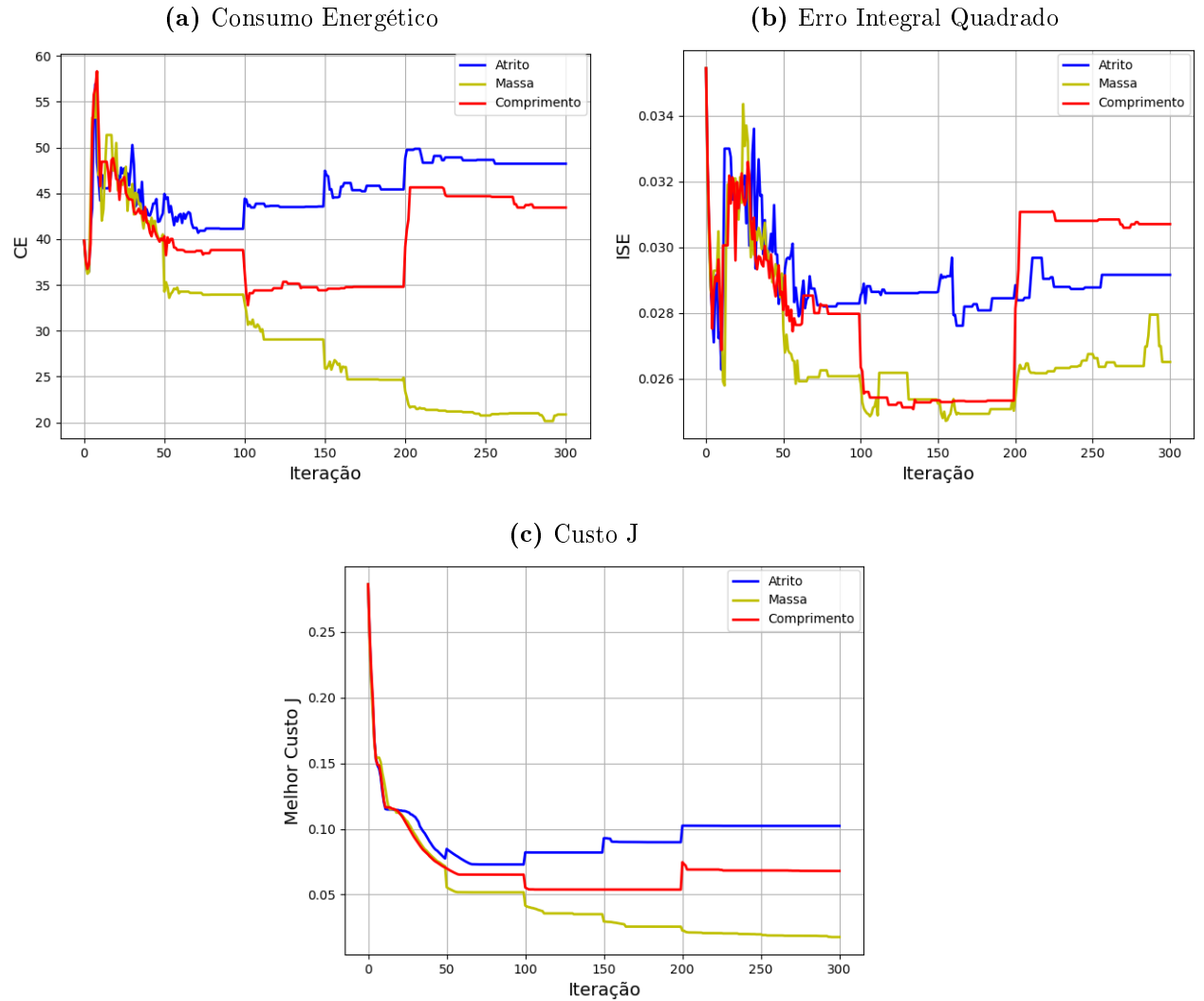
Fonte: Autor Próprio

Na Figura 32 é apresentado a progressão das posições das partículas da população D durante o funcionamento do algoritmo para as situações de mudanças de parâmetros, deste modo, é visto que as demais partículas no espaço de busca K_1 durante as iterações não mudaram muito a região onde estavam procurando a melhor posição, diferente dos demais espaços de busca, onde em alguns momentos das mudanças dos parâmetros podem ser relacionados aos pontos de mudanças nos parâmetros.

Figura 32- Progressão das posições da população D

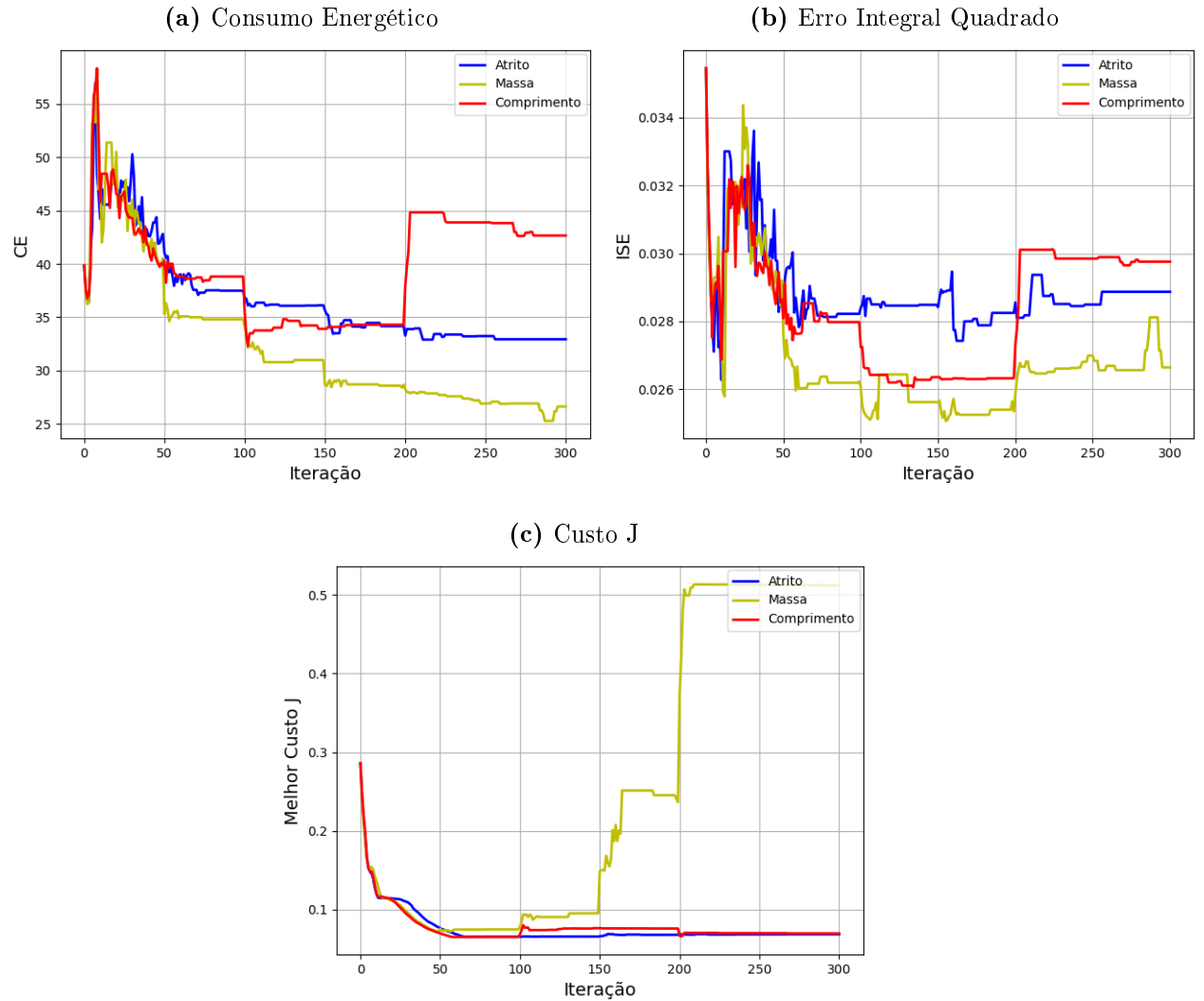


Para uma análise mais sensível das melhores posições das partículas apresentadas na Figura 32 é analisado os índices de custo (53) e, seus componentes consumo energético (55) e erro quadrado integral (56) serão analisados para cada situação de mudança.

Figura 33- Análise das melhores posições da população D

Fonte: Autor Próprio

No gráfico da Figura 33c é visto que o custo foi otimizado em relação ao custo inicial, contudo para as mudanças no atrito e de comprimento seus resultados são maiores em relação ao menor custo encontrado por eles. Em questão do consumo energético, as mudanças no atrito e no comprimento exigem um maior consumo de força do atuador. Entretanto, ao analisar o erro quadrado integral, Figura 33b, as posições das partículas encontradas pela mudança no atrito se aproximaram do erro quadrado integral das posições encontradas pela mudança de massa. Em seguida, as melhores posições foram simuladas em um sistema sem mudanças nos seus parâmetros.

Figura 34- Situação sem mudanças das melhores posições da população D

Fonte: Autor Próprio

Seguindo o padrão das demais populações simuladas, as posições das partículas de massa apresentam um consumo energético menor e um erro integral quadrado menor em relação as demais, mas devido as matrizes de ponderação é visto que seu erro em relação ao referencial de posição do carro e posição do pêndulo aumentam seu consumo J . As posições das partículas encontradas pela mudança de atrito apresentam um resultado melhor que o inicial e comparado as posições encontradas pelo comprimento, entretanto, similar as posições encontradas pela massa, seu erro em relação ao referencial de posição do carro e do pêndulo foram maiores para as últimas posições das partículas encontradas criando uma certa diferença em relação as posições encontradas pela mudança no comprimento.

7.5 População E

Nesta população foi utilizado o a técnica LQR formulada em LMI inserindo incertezas nos parâmetros de coeficiente de atrito, mas foi gerada para um sistema diferente do sistema na qual o algoritmo será aplicado.

Nas tabelas 19, 20 e 21 é visto os melhores ganhos nas iterações inicial e final e nas iterações 50, 100, 150 e 200 onde ocorreram as situações de mudança impostas ao algoritmo, como mudança na massa, comprimento e nos atritos. Por meio dessas tabelas é visto que os valores do espaço de busca K_1 sofreu uma alteração grande na mudança de massa, mas para as demais mudanças a região entre 20 e 30 ofereceram melhor desempenho de acordo com algoritmo. As posições das partículas encontradas pela mudança de massa oferecem vetores de controle longe das regiões onde as posições encontradas pela mudança de atrito e comprimento estão.

Tabela 19- Tabela das Melhores Posições da População E - Mudança na Massa

Iteração	K_1	K_2	K_3	K_4
0	-25	322.7206	-66.5379	97.2458
50	-28.0208	423.3725	-125.5299	146.4044
100	-51.4891	687.7407	-212.4636	206.0311
150	-57.4857	772.4419	-186.3405	189.9143
200	-56.6936	820.3027	-204.8388	184.2061
300	-56.1412	900.1587	-209.9379	178.4718

Fonte: Autor Próprio

Tabela 20- Tabela das Melhores Posições da População E - Mudança na Comprimento

Iteração	K_1	K_2	K_3	K_4
0	-25	322.7206	-66.5379	97.2458
50	-23.9458	325.7808	-74.0927	104.5019
100	-24.6966	325.8385	-73.7910	104.3994
150	-28.0227	411.3397	-119.6974	127.7207
200	-28.7141	410.6295	-120.6964	127.8352
300	-29.3711	423.3904	-141.8784	166.7927

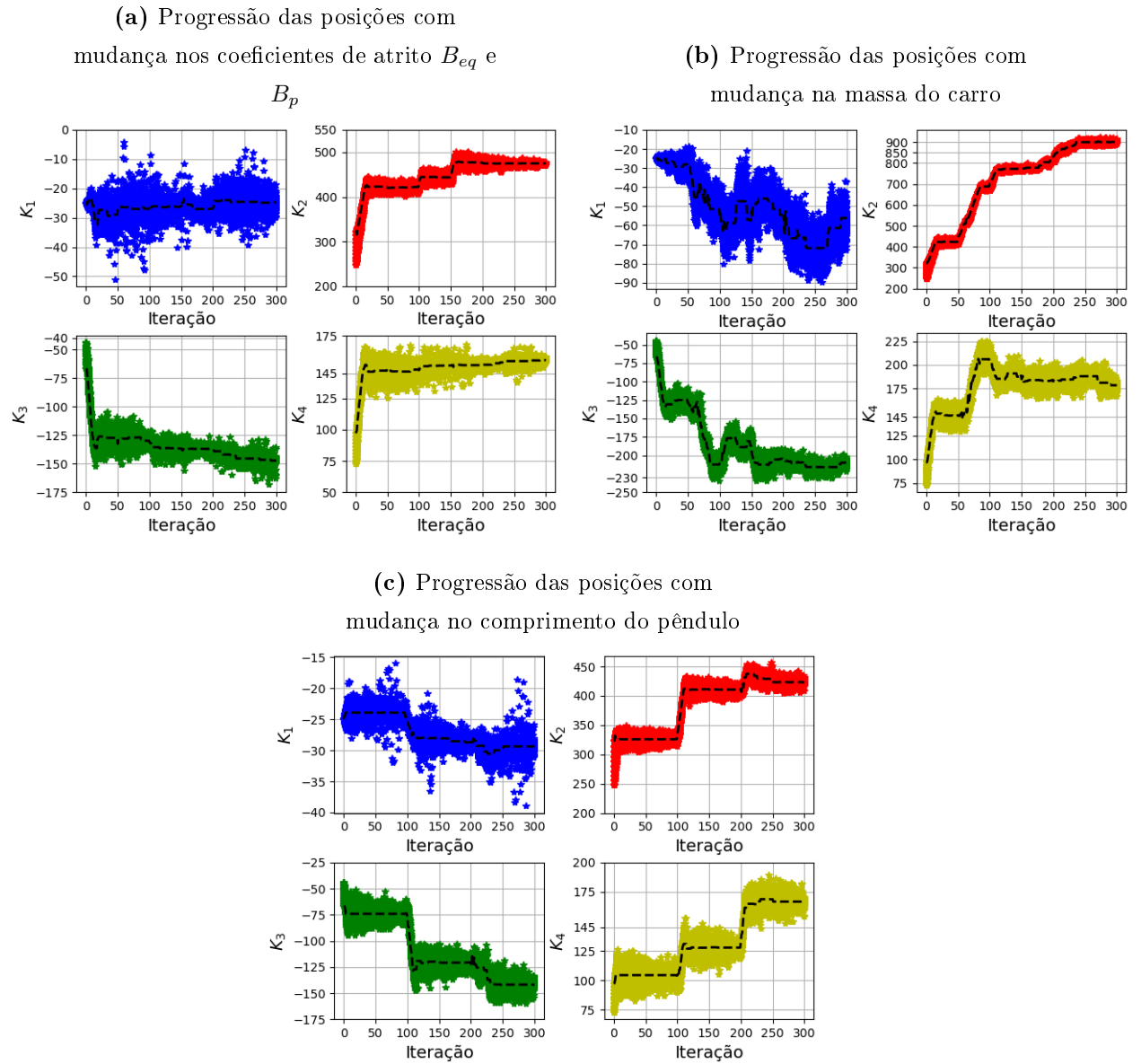
Fonte: Autor Próprio

Tabela 21- Tabela das Melhores Posições da População E - Mudança no Atrito

Iteração	K_1	K_2	K_3	K_4
0	-25	322.7206	-66.5379	97.2458
50	-28.9632	422.9378	-127.3889	147.1293
100	-27.0091	422.2094	-129.9789	148.0539
150	-26.1958	443.7545	-136.7804	151.2702
200	-26.9311	476.9175	-137.6588	151.6854
300	-24.7519	474.9497	-147.3283	155.5752

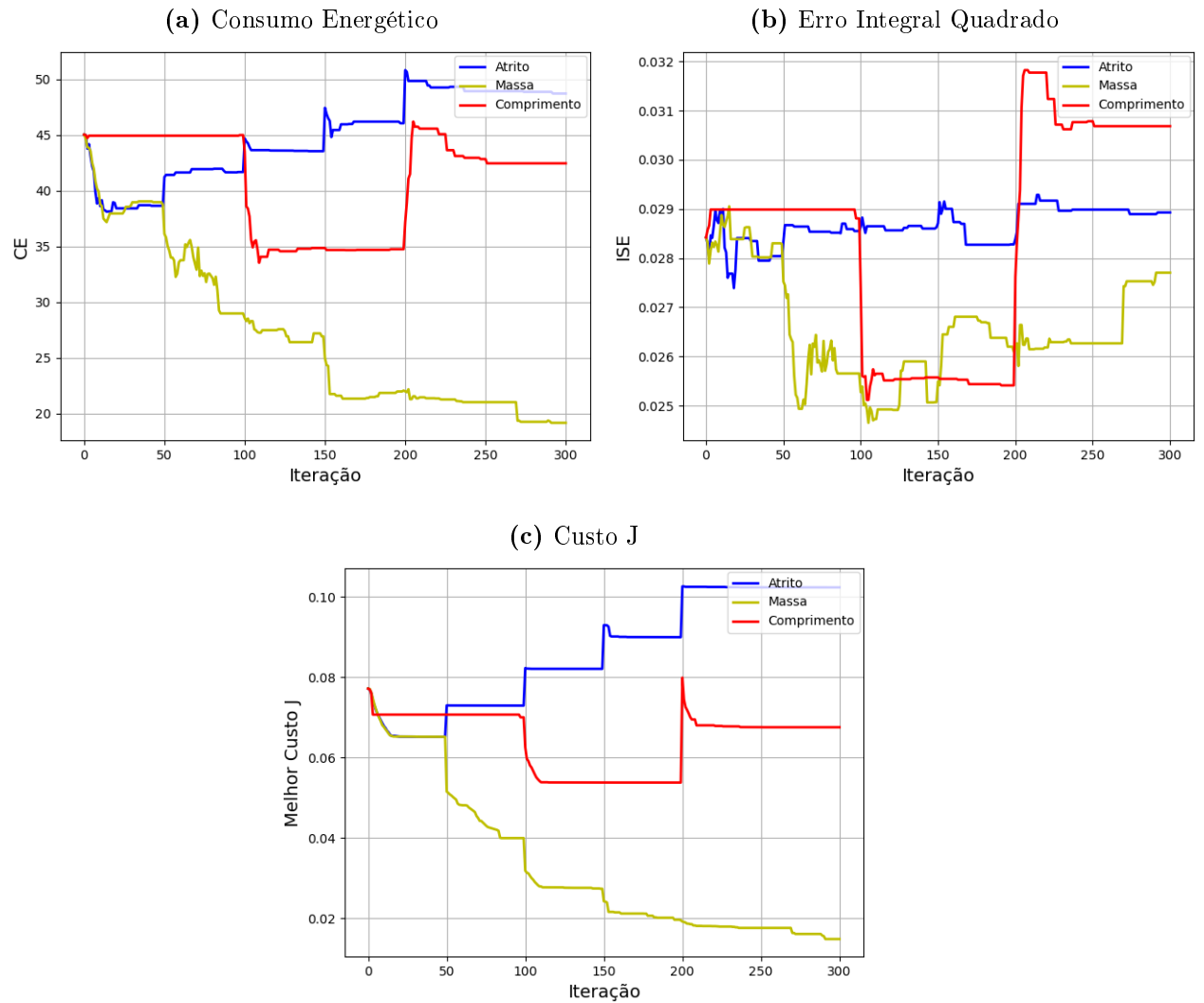
Fonte: Autor Próprio

Na Figura 35 é apresentado a progressão das posições das partículas e das melhores posições da população E durante o funcionamento do algoritmo para as situações de mudanças de parâmetros. Ao analisar os gráficos da Figura 35 é visto que no espaço de busca K_2 houve uma maior sensibilidade as mudanças nos parâmetros do sistema, ademais, apenas nos espaços de busca K_3 e K_4 da mudança no comprimento as posições foram sensíveis a tais mudanças.

Figura 35- Progressão das posições da população E

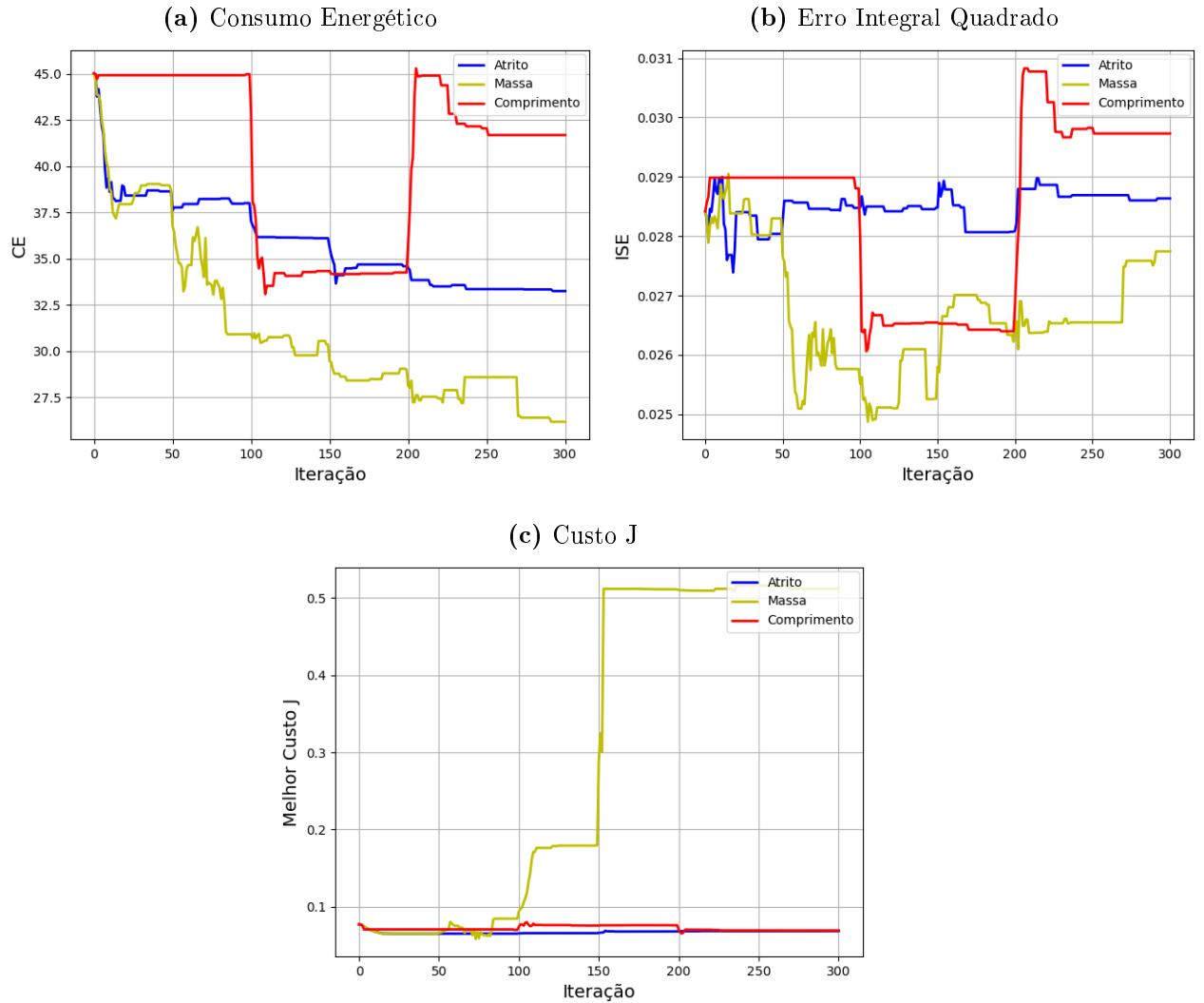
Fonte: Autor Próprio

Em seguida, foi analisado os índices de desempenho de custo (53) e, seus componentes consumo energético (55) e erro quadrado integral (56) das melhores posições das partículas para cada mudança de parâmetro.

Figura 36- Análise das melhores posições da população E

Fonte: Autor Próprio

Nos gráficos da Figura 36 é visto que o custo das melhores posições das partículas segue o mesmo padrão das demais populações. O comportamento do custo apresentando na Figura 36c da população C, Figura 30c, visto que essas populações foram geradas para um sistema diferente do sistema no qual seria aplicado o controle/algorithm. Em seguida é analisado as melhores posições das partículas para o sistema sem mudanças em seus parâmetros e avaliado sua resposta pelos índices de desempenho de custo, consumo energético e ISE.

Figura 37- Situação sem mudanças das melhores posições da população E

Fonte: Autor Próprio

Seguindo o padrão das demais populações, as melhores posições das partículas encontradas pela mudança de massa apresentam um custo elevado para um sistema sem alterações, mesmo possuindo um consumo energético e ISE menor, tal comportamento é dado devido as matrizes de ponderação de (53). Deste modo, é possível entender que as posições encontradas pela massa apresentam um erro de referência em relação a posição do carro e do pêndulo altos em comparação com as demais. As posições encontradas pela mudança no atrito nesta população obtiveram melhor consumo energético e ISE do que as encontradas pela mudança de comprimento. Contudo, pelo mesmo motivo das posições encontradas pela massa, o erro de referência da posição do carro e do pêndulo reduzem o custo J das partículas encontradas pela mudança de comprimento aproximando bastante do custo das posições encontradas pela mudança de atrito ao final da iteração.

7.6 Comparação Populacional

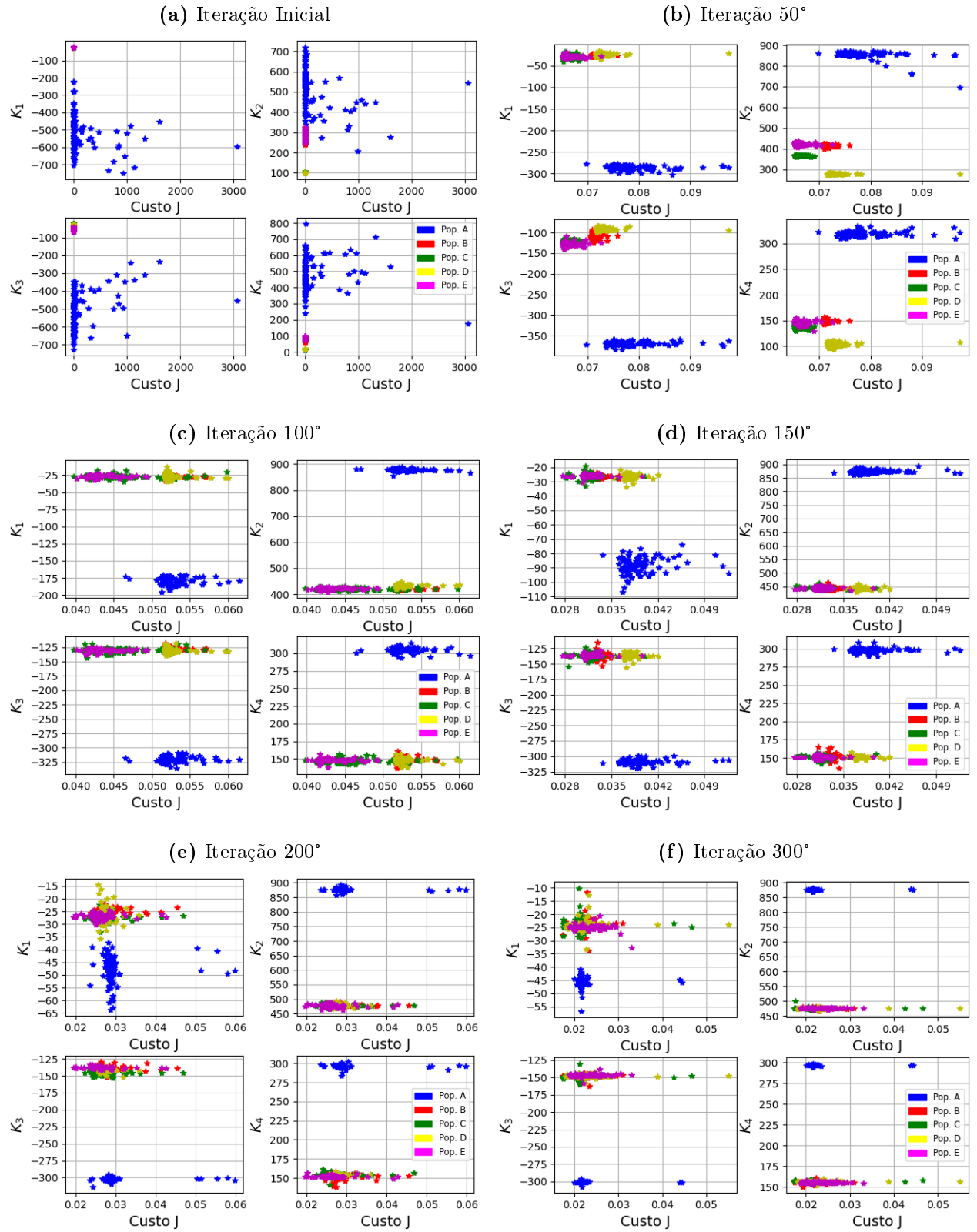
Após apresentado o desempenho individual de cada população para cada situação de mudança de parâmetro, nesta seção todas as populações serão comparadas em relação a posição dos vetores no espaço de busca e seu custo J .

Na Figura 38 é apresentado o percurso das posições das partículas durante as iterações para a mudança de massa para todas as populações geradas. Devido a população A ter sido gerada de modo aleatório, algumas posições das partículas geram valores de custos altos, podendo representar que alguns vetores de controle gerados pela população não controlam o sistema.

Nas iterações seguintes as posições das partículas encontram valores de custo próximos, mas, apenas as populações B,C,D e E se concentram na mesma região, já a população A se encontra numa região no espaço acima das demais.

Nas últimas iterações visto que algumas posições das populações se concentraram numa mesma região, mas mantendo poucas posições com um custo maior longe em uma tentativa de encontrar um custo melhor. Também é visto que nas iterações finais todas as populações chegaram aproximadamente no mesmo custo, devido a redução de massa facilitar o controle de acordo com a simulação

Figura 38- Progressão das populações na mudança de massa

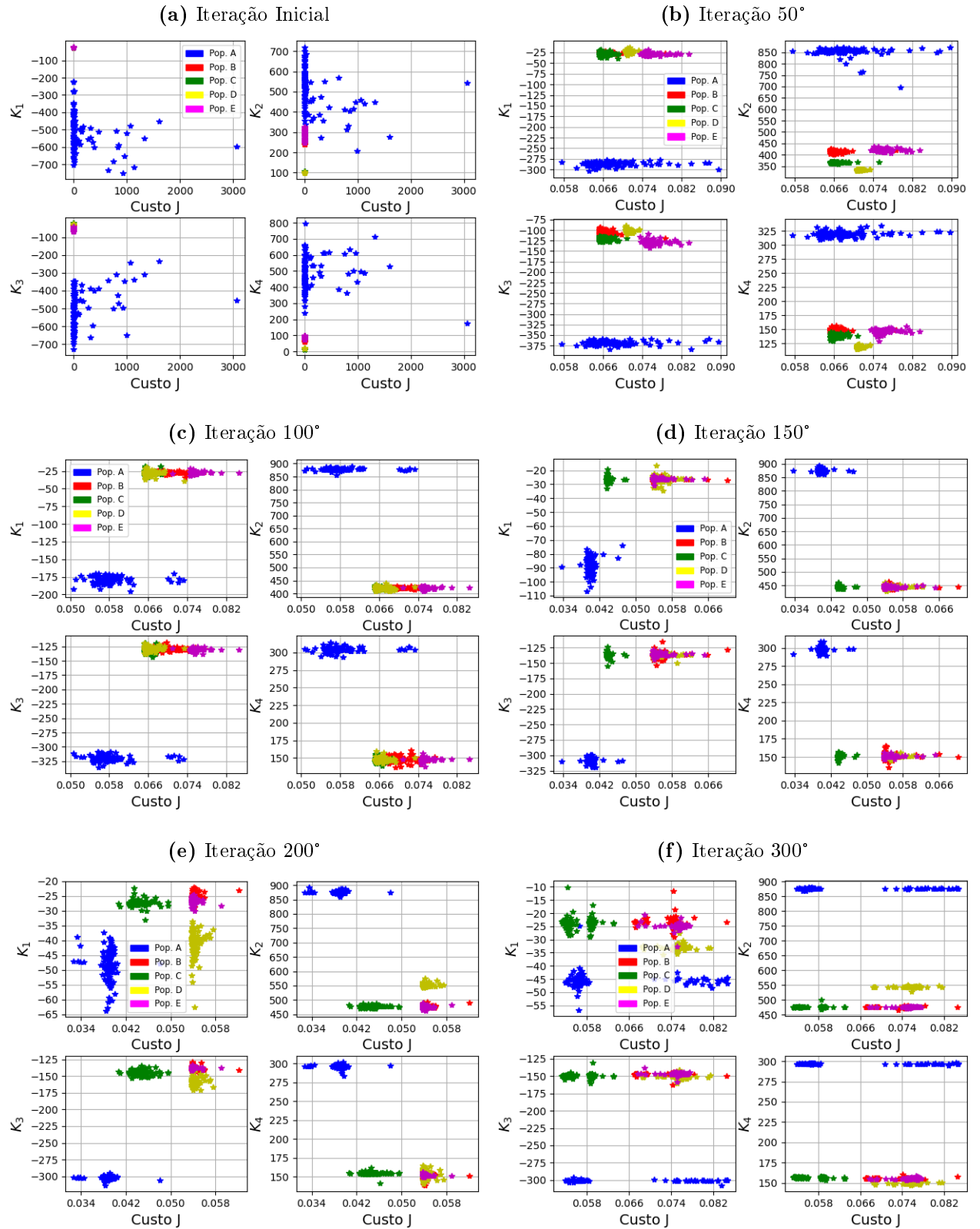


Na Figura 39 é apresentado o percurso das posições das partículas durante as iterações

para a mudança de comprimento para todas as populações geradas. Nas seguintes iterações é visto que as populações B, C, D e D se encontram em regiões no espaço de busca próximos e com custos também aproximados, já a população A se encontra em uma região dos espaços de busca longe das demais, mas abrangendo valores de custo próximos e com partículas exploratórias que encontram valores de custo menores direcionando a população.

Na primeira redução do valor do comprimento da haste do pêndulo na iteração 100° é visto que das as posições partículas em todas as populações se encontravam em regiões robustas que continuaram a otimizar o vetor de controle sem aumento no custo. Na segunda redução do valor do comprimento da haste do pêndulo na iteração 200 é visto que as posições das partículas ainda se encontram em uma região robusta, pois, há um pico no custo, mas logo é ajeitado. Tais dinâmicas podem ser lembradas nas Figuras 23, 27c, 30c, 33c e 36c. Ao final da simulação é visto que as posições das partículas encontram regiões próximas, mas com diferentes custos mostrando que, no caso dos gráficos da Figura 39f, a posição no espaço de busca K_1 está afetando o vetor de controle fortemente.

Figura 39- Progressão das populações na mudança de comprimento

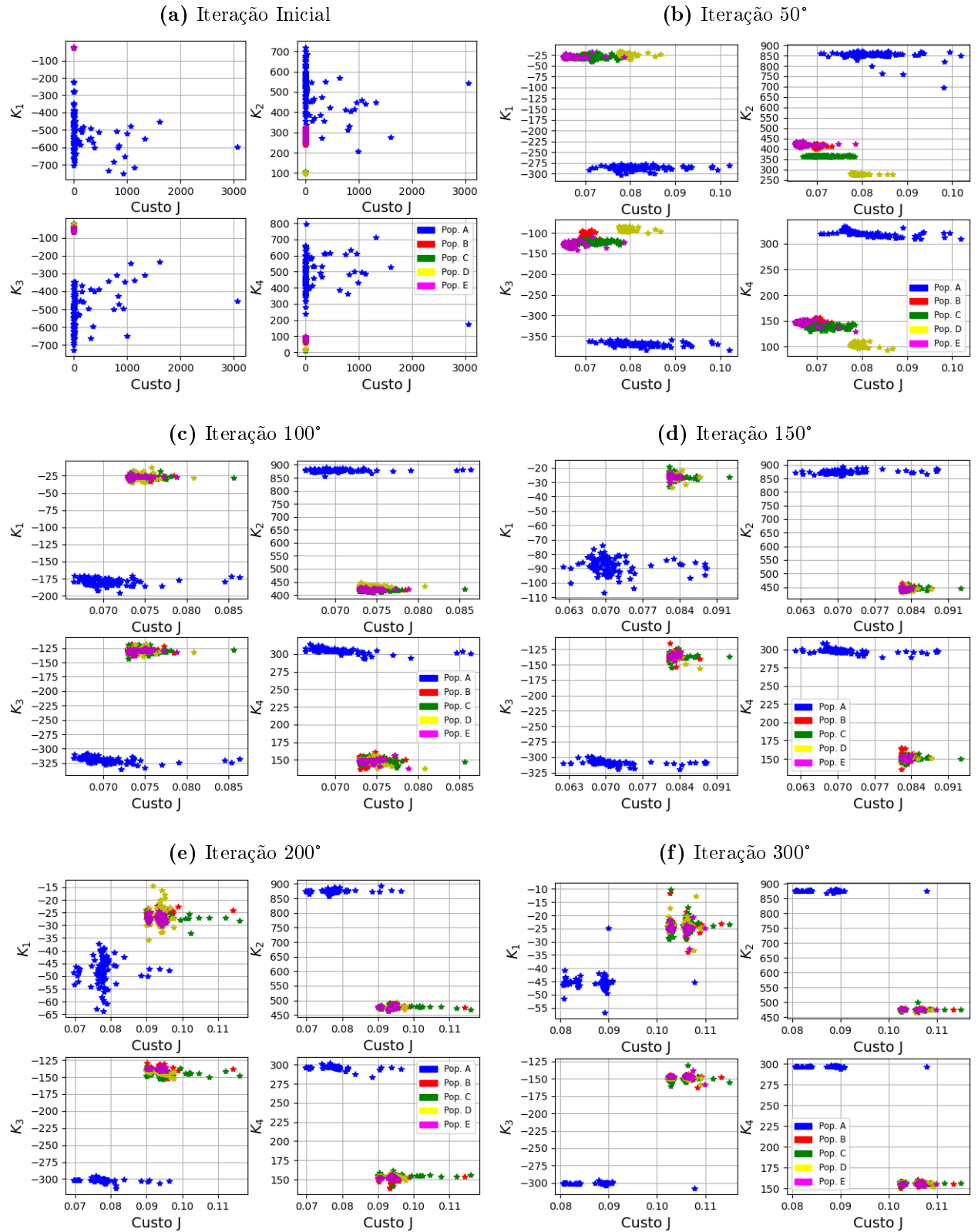


Na Figura 40 é apresentado o percurso das posições das partículas durante as iterações

para a mudança de atrito para todas as populações geradas. Na iteração 50°, onde ocorreu a primeira mudança nos coeficientes de atrito, é visto que as populações B, C, D e E se encontram em regiões próximas nos espaços de busca, mas com diferentes valores de custo, ademais, a população A se localiza em uma região longe das demais e abrangendo uma quantidade maior de valores de custo desde altos até menores que as demais populações nos espaços de busca. Na iteração 100, iteração da segunda mudança nos coeficientes de atrito, é visto que as populações B, C, D, E convergiram parcialmente em uma região com valor de custo semelhante nos espaços de busca, possuindo uma elevação do custo em relação ao menor custo encontrado entre elas, mas a população A, devido ao seu caráter aleatório e bem distribuído as posições das partículas desta população abrangeram posições com custo melhor diante a mudança nos coeficientes de atrito quando comparado as demais populações.

Nas seguintes iterações é visto que a população A se encontra em regiões onde o custo é melhor comparado as demais populações, mas é possível perceber que algumas posições das partículas desta população possuem o mesmo custo das demais populações. Deste modo, é visto que todas as populações ao final da simulação estão com suas posições aproximadamente concentradas em algumas regiões, no espaço de busca K_1 é visto que as populações estão concentradas na região entre -10 e -55, no espaço de busca K_2 a população A se alojou longe das demais populações, no espaço de busca K_3 a grande maioria das populações se alojaram na região perto do valor -150 diferente da população A que ficou na região -300, e por fim no espaço K_4 a população B, C, D e E escolherem regiões próximas e a população A uma região distante.

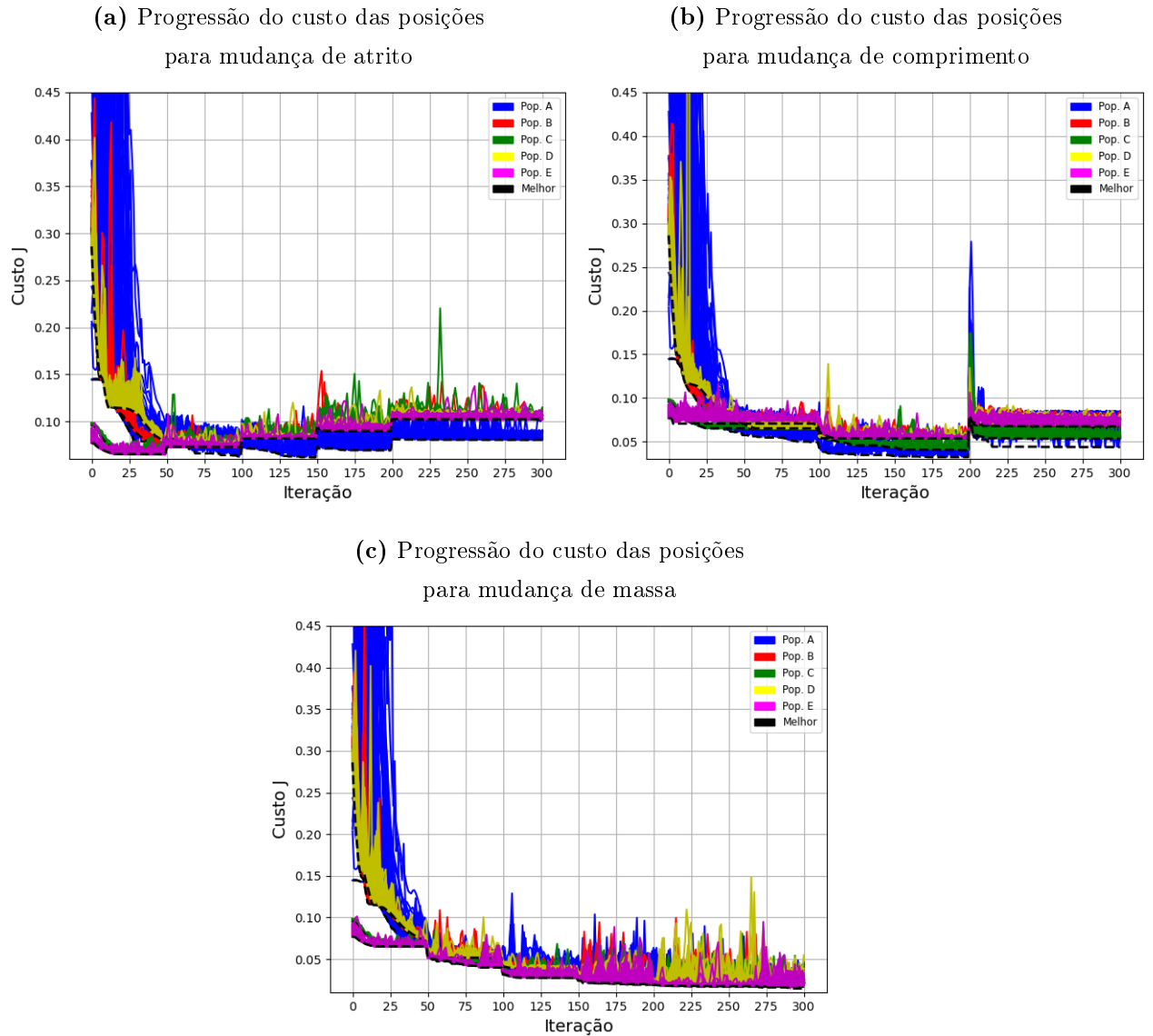
Figura 40- Progressão das populações na mudança de atrito



As Figuras 38, 39 e 40 apresentam a progressão das posições das partículas nos espaços

de busca do vetor de controle, mostrando como cada posição influência no custo e as regiões onde cada população se alojou. A população A nas situações das mudanças no atrito e no comprimento obteve o menor custo em relação as demais populações, contudo, ao observar os gráficos da Figura 41, é possível observar a quantidade de iterações nas quais a população A, inicialmente, precisou para obter um custo igual as demais populações, apenas após 50 iterações a população A conseguiu igual o custo as demais e superar. Pois, por se tratar de uma população aleatória alguns vetores produzem respostas instáveis, demandando do algoritmo dedicar iterações para apenas encontrar posições de partículas com resposta estáveis.

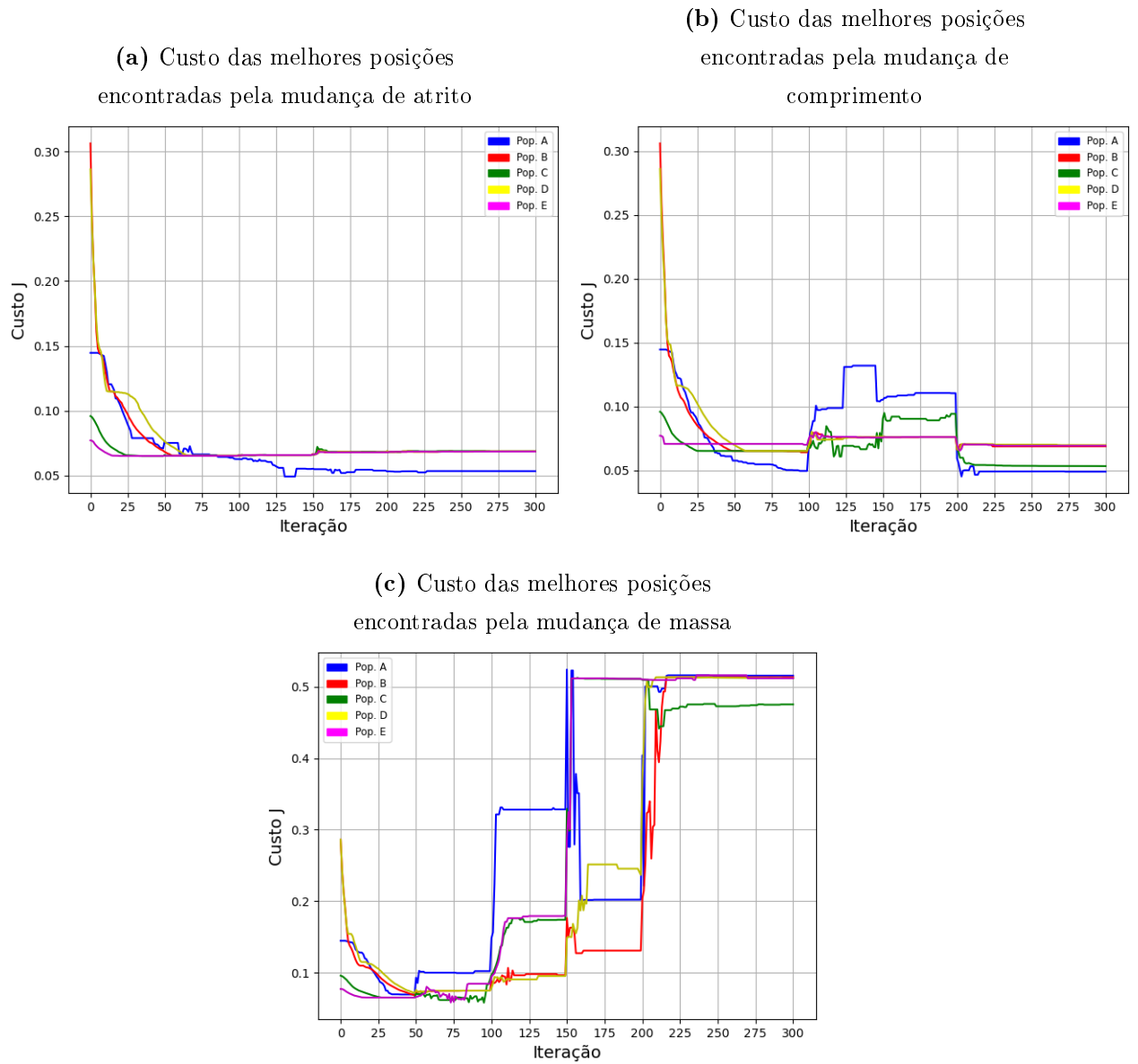
Além da demanda de iterações foi visto que a população A utiliza uma maior quantidade de tempo durante as iterações iniciais, em média foi contado 6 minutos por iteração até a iteração 60, após essa iteração a média foi reduzida para 3 minutos por iteração, para as demais populações foi quantificado o tempo por iteração sendo em média 3 minutos por iteração.

Figura 41- Progressão do custo das partículas

Fonte: Autor Próprio

Por fim, além do número de iterações e a quantidade de tempo, ao analisar o custo das melhores posições das partículas para a situação sem mudança e comparar entre as populações, Figura 42, é visto que as posições das partículas da população A encontradas pela mudança de massa sofreram maiores variações durante as iterações resultando em um custo maior que as demais populações, por menor que seja a diferença. Para as posições encontradas pela mudança no comprimento é visto que todas as populações sofreram um salto em seu custo na primeira mudança na iteração 100°, também é visto que a população E, foi a população na qual o custo das posições das partículas encontradas variou menos. A população E possui um comportamento semelhante para suas posições encontradas pela mudança de atrito também tendo menor variação de seu custo.

Figura 42- Custo das melhores posições para situação sem mudança



Fonte: Autor Próprio

8 CONSIDERAÇÕES FINAIS

Neste trabalho, pôde-se verificar como o *machine learning*, utilizando o algoritmo *particle swarm optimization*, gerou vetores de controle otimizados para diferentes situações, onde os parâmetros do sistema a ser controlado foram alterados e como diferentes populações de vetores de ganhos gerados de modos diferentes afetam o desempenho do algoritmo para otimizar novos vetores de controle.

O sistema utilizado foi o modelo pêndulo invertido e os parâmetros alterados foram a massa do carro, sendo essa reduzida em 20% a cada 50 iterações, o comprimento da haste do pêndulo, sendo essa reduzida em um terço a cada 100 iterações, e por fim os coeficientes de atrito B_p e B_{eq} , sendo esses aumentados em 10% a cada 50 iterações.

Para o algoritmo, foi definido um caráter exploratório e uma função de custo que utiliza apenas três grandezas para avaliar o desempenho do vetor de controle, tais grandezas foram a posição do pêndulo, a posição do carro e a força do atuador. Deste modo, as posições das partículas seriam avaliadas e ao final da avaliação a melhor posição seria escolhida e reavaliada, para que o algoritmo saiba compreender se algo ocorreu com o ambiente ou o sistema sem necessariamente saber qual mudança ocorreu.

No caso da mudança de massa, esta mudança facilitou o controle do algoritmo e consequentemente o algoritmo encontrou uma resposta local ótima para esta mudança, pois, em todas as populações quando os vetores de controle encontrados foram simulados para uma situação onde o sistema não sofreu nenhuma alteração, seu custo aumentou radicalmente, mesmo em relação ao custo das populações iniciais.

No caso da mudança de comprimento haste, esta mudança é algo que dificulta o controle, mas que não perdura ao decorrer das iterações, pois, o algoritmo consegue se adaptar e reduzir o custo dos vetores que são encontrados. Diferente da mudança no atrito, que também dificulta o controle por dificultar o movimento do atuador, necessitando de mais força e energia para realizar o controle. Contudo, estas mudanças, em todas as populações geradas, foram a que forçaram o algoritmo a encontrar vetores de controle mais robustos que quando simulados para uma situação sem mudança nos parâmetros do sistema, apresentaram um custo menor do que a população inicial e sendo otimizados ao decorrer das iterações.

Ao comparar as populações é visto que a população A, gerada de modo aleatório, foi a população que obteve melhor custo quando comparado as demais, entretanto, foi a população que mais demandou tempo e que quando foi exposta a mudanças na massa do carro, foi a que se direcionou primeiro a uma região do espaço de busca onde quando exposta a situação sem mudança, as posições das partículas alcançaram os piores custos. Logo, uma população gerada de maneira aleatória pode obter custos bons, mas demandará maior tempo e poderá ser facilmente influenciada e direcionada a uma região relativamente ruim.

Por fim, é visto que cada alteração possui um impacto diferente no controle do sistema, em sua performance e como a máquina irá entender o que precisa ser realizado, e também que cada população influencia a adaptabilidade e robustez das futuras respostas. O controle via *machine learning* se mostrou eficiente para as diferentes situações impostas, mesmo que o algoritmo não tenha sido programado para englobar ou foi treinado para tal mudança. Neste controle a quantidade de iterações de treinamento para se adaptar a cada mudança não foi estritamente fixa, e foi possível concluir que as populações mesmo possuindo diferentes origens ou sendo criadas para diferentes sistemas, o algoritmo chegou em custos próximos otimizados, mesmo em regiões diferentes nos espaços de busca para cada população. E também ao simular os vetores de controle encontrados para as diferentes situações de mudanças em um sistema sem mudança nenhuma o algoritmo, foi obtido respostas melhores em relação ao custo inicial em sua grande maioria.

Como proposta de continuação de trabalho é sugerido contruir mais populações utilizando técnicas de controle que não requerem linearizar o sistema e também personalizar as partículas não mantendo a estrutura da lei de controle fixa, para obter além diferentes valores para o vetor de controle, obter diferentes estruturas do vetor de controle.

REFERÊNCIAS

- ABRAHAM, A.; GUO, H.; LIU, H. Swarm intelligence: foundations, perspectives and applications. In: **Swarm intelligent systems**. [S.l.]: Springer, 2006. p. 3–25.
- AKHTARUZZAMAN, M.; SHAFIE, A. A. Modeling and control of a rotary inverted pendulum using various methods, comparative assessment and result analysis. In: IEEE. **2010 IEEE International Conference on Mechatronics and Automation**. [S.l.], 2010. p. 1342–1347.
- BLUM, C.; MERKLE, D. **Swarm intelligence: introduction and applications**. [S.l.]: Springer Science & Business Media, 2008.
- BONACCORSO, G. **Machine learning algorithms**. [S.l.]: Packt Publishing Ltd, 2017.
- BOYD, S.; GHAOUI, L. E.; FERON, E.; BALAKRISHNAN, V. **Linear matrix inequalities in system and control theory**. [S.l.]: SIAM, 1994.
- BUŞONIU, L.; BRUIN, T. de; TOLIĆ, D.; KOBER, J.; PALUNKO, I. Reinforcement learning for control: Performance, stability, and deep approximators. **Annual Reviews in Control**, Oxford, v. 46, p. 8–28, 2018.
- CHAKRABORTY, A.; KAR, A. K. Swarm intelligence: A review of algorithms. In: . [S.l.]: Springer, 2017. p. 475–494.
- DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. **Theoretical computer science**, Amsterdam, v. 344, n. 2-3, p. 243–278, 2005.
- DURIEZ, T.; BRUNTON, S. L.; NOACK, B. R. **Machine learning control-taming nonlinear dynamics and turbulence**. [S.l.]: Springer, 2017.
- FERON, E.; BALAKRISHNAN, V.; BOYD, S.; GHAOUI, L. E. Numerical methods for h 2 related problems. In: IEEE. **1992 American Control Conference**. [S.l.], 1992. p. 2921–2922.
- HE, D.; CHANG, H.; CHANG, Q.; LIU, Y. Particle swarm optimization based on the initial population of clustering. In: IEEE. **2010 Sixth International Conference on Natural Computation**. [S.l.], 2010. v. 5, p. 2664–2667.
- JANPROM, K.; PERMPOONSINSUP, W.; WANGNIPPARNTO, S. Intelligent tuning of pid using metaheuristic optimization for temperature and relative humidity control of comfortable rooms. In: . [S.l.]: Hindawi, 2020. v. 2020.
- JHA, S. et al. Comparative study of different classical and modern control techniques for the position control of sophisticated mechatronic system. In: . [S.l.]: Elsevier, 2016. v. 93, p. 1038–1045.

KARMOUCHE, D. C. J. **DiogoKarmouche/PSO-PenduloInvetido**. Acessado em 11/03/2022. Disponível em: <<https://github.com/DiogoKarmouche/PSO-PenduloInvetido>>.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. **Proceedings of ICNN'95-international conference on neural networks**. [S.l.], 1995. v. 4, p. 1942–1948.

KIRANYAZ, S.; INCE, T.; GABBOUJ, M. **Multidimensional particle swarm optimization for machine learning and pattern recognition**. [S.l.]: Springer, 2014.

LANGLEY, P.; LAIRD, J. E. Artificial intelligence and intelligent systems. In: . [S.l.: s.n.], 2006. v. 1001, p. 48109–2121.

LATHA, K.; RAJINIKANTH, V.; SUREKHA, P. Pso-based pid controller design for a class of stable and unstable systems. In: . [S.l.]: Hindawi, 2013. v. 2013.

LI, E.; ZHOU, Z.; CHEN, X. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In: **Proceedings of the 2018 Workshop on Mobile Edge Communications**. [S.l.: s.n.], 2018. p. 31–36.

LIN, J.-H. Particle swarm optimization for control of nonlinear dynamics. In: IEEE. **First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)**. [S.l.], 2006. v. 1, p. 542–545.

MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey wolf optimizer. In: . [S.l.]: Elsevier, 2014. v. 69, p. 46–61.

MUELLER, J. P.; MASSARON, L. **Machine learning for dummies**. [S.l.]: John Wiley & Sons, 2021.

NETTO, L.; LIMA, J. et al. Controle cooperativo h2 e hinf via rede de comunicação: teoria e implementação prática em pêndulos invertidos. [sn], 2018.

OGATA, K. **Modern control engineering**. [S.l.]: Prentice hall, 2010.

OLALLA, C.; LEYVA, R.; AROUDI, A. E.; QUEINNEC, I. Robust lqr control for pwm converters: An lmi approach. In: . [S.l.]: IEEE, 2009. v. 56, n. 7, p. 2548–2558.

OLIVEIRA, P. de M.; HEDENGREN, J. D.; PIRES, E. Swarm-based design of proportional integral and derivative controllers using a compromise cost function: An arduino temperature laboratory case study. In: . [S.l.]: Multidisciplinary Digital Publishing Institute, 2020. v. 13, n. 12, p. 315.

PACKARD, A.; POOLLA, K.; HOROWITZ, R. **ME 132, Dynamic Systems and Feedback Class Notes**. [S.l.]: Berkeley, US, 2005.

PANTALEÓN, E.; TANAKA, D.; BERNARDES, F. Análise das variações do coeficiente de atrito e as correlações com os mecanismos de desgaste. In: . [S.l.]: Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, 2012. v. 1, p. 62–72.

PARSOPOULOS, K. E.; VRAHATIS, M. N. **Particle swarm optimization and intelligence: advances and applications: advances and applications**. [S.l.]: IGI global, 2010.

PHIL, K. **Matlab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence**. [S.l.]: Apress, 2017.

PIOTROWSKI, A. P.; NAPIORKOWSKI, J. J.; PIOTROWSKA, A. E. Population size in particle swarm optimization. In: . [S.l.]: Elsevier, 2020. v. 58, p. 100718.

POLI, R. Analysis of the publications on the applications of particle swarm optimisation. **Journal of Artificial Evolution and Applications**, Hindawi, v. 2008, 2008.

RINI, D. P.; SHAMSUDDIN, S. M.; YUHANIZ, S. S. Particle swarm optimization: technique, system and challenges. In: . [S.l.]: International Journal of Computer Applications, 244 5 th Avenue, # 1526, New York, 2011. v. 14, n. 1, p. 19–26.

ROY, S.; BISWAS, S.; CHAUDHURI, S. S. Nature-inspired swarm intelligence and its applications. **International Journal of Modern Education & Computer Science**, v. 6, n. 12, 2014.

ROZALI, S. M.; RAHMAT, F.; HUSAIN, A. R. et al. Performance comparison of particle swarm optimization and gravitational search algorithm to the designed of controller for nonlinear system. In: . [S.l.]: Hindawi, 2014. v. 2014.

SARKAWI, H.; OHTA, Y. Uncertain dc-dc zeta converter control in convex polytope model based on lmi approach. In: . [S.l.]: IAES Institute of Advanced Engineering and Science, 2018. v. 9, n. 2, p. 829.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A.; DOYLE, F. J. **Process dynamics and control**. [S.l.]: John Wiley & Sons, 2017.

SERAPIÃO, A. B. d. S. Fundamentos de otimização por inteligência de enxames: uma visão geral. In: . [S.l.]: SciELO Brasil, 2009. v. 20, p. 271–304.

SIDEY-GIBBONS, J. A.; SIDEY-GIBBONS, C. J. Machine learning in medicine: a practical introduction. **BMC medical research methodology**, BioMed Central, v. 19, n. 1, p. 1–18, 2019.

THOBANI, S. **Improving e-Commerce sales using machine learning**. Tese (Doutorado) — Massachusetts Institute of Technology, 2018.

WAHAB, M. N. A.; NEFTI-MEZIANI, S.; ATYABI, A. A comprehensive review of swarm optimization algorithms. **PloS one**, Public Library of Science San Francisco, CA USA, v. 10, n. 5, p. e0122827, 2015.

WU, Z.; FAN, D.; ZHOU, Y.; LI, R.; NOACK, B. R. Jet mixing optimization using machine learning control. **Experiments in Fluids**, Berlin, v. 59, n. 8, p. 1–17, 2018.

YU, T.; WANG, L.; HAN, X.; LIU, Y.; ZHANG, L. Swarm intelligence optimization algorithms and their application. In: . [S.l.: s.n.], 2015. v. 3.

ZHAO, J.; LI, T.; QIAN, J. Application of particle swarm optimization algorithm on robust pid controller tuning. In: SPRINGER. **International Conference on Natural Computation**. [S.l.], 2005. p. 948–957.

APÊNDICE A – PROGRAMAÇÃO

Os códigos do testes e das simulações do algoritmo PSO e do pêndulo invertido podem ser encontrado no repertório do *GitHub* PSO-PenduloInvertido (<https://github.com/DiogoKarmouche/PSO-PenduloInvetido>) (KARMOUCHE,).