

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DANIEL SIMÃO HOREVICZ

ESTUDO DA TÉCNICA ANT COLONY LEARNING NO  
CONTROLE DE SISTEMAS NÃO-LINEARES

DISSERTAÇÃO

CORNÉLIO PROCÓPIO

2017

DANIEL SIMÃO HOREVICZ

**ESTUDO DA TÉCNICA ANT COLONY LEARNING NO  
CONTROLE DE SISTEMAS NÃO-LINEARES**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Engenharia Elétrica” – Área de Concentração: Controle e Automação de Sistemas.

Orientador: Dr. Fábio Renan Durand

Co-orientador: Dr. Cristiano Marcos Agulhari

**CORNÉLIO PROCÓPIO**

**2017**

---

### Dados Internacionais de Catalogação na Publicação

---

H811 Horevicz, Daniel Simão

Estudo da técnica Ant Colony Learning no controle de sistemas não-lineares / Daniel Simão Horevicz. – 2017.  
57 f. : il. color. ; 31 cm.

Orientador: Fábio Renan Durand.

Coorientador: Cristiano Marcos Agulhari.

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. Cornélio Procópio, 2017.

Bibliografia: p. 56-57.

1. Otimização matemática. 2. Sistemas não-lineares. 3. Algoritmos heurísticos. 4. Engenharia Elétrica – Dissertações. I. Durand, Fábio Renan, orient. II. Agulhari, Cristiano Marcos, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

CDD (22. ed.) 621.3

---

### Biblioteca da UTFPR - Câmpus Cornélio Procópio

Bibliotecários/Documentalistas responsáveis:  
Simone Fidêncio de Oliveira Guerra – CRB-9/1276  
Romeu Righetti de Araujo – CRB-9/1676

## TERMO DE APROVAÇÃO

Título da Dissertação Nº 042:

# “Estudo da Técnica Ant Colony Learning no controle de Sistemas Não-Lineares”.

por

**Daniel Simao Horevicz**

Orientador: **Prof. Dr. Fábio Renan Durand**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM ENGENHARIA ELÉTRICA – Área de Concentração: Sistemas Eletrônicos Industriais, pelo Programa de Pós-Graduação em Engenharia Elétrica – PPGEE – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 14:30h do dia **25 de agosto de 2017**. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:

---

Prof. Dr. Cristiano Marcos Agulhari  
Presidente

---

Prof. Dr. Leonimer Flávio de Melo  
UEL

---

Prof. Dr. Emerson Ravazzi Pires da Silva  
UTFPR-CP

Visto da coordenação:

---

Prof. Dr. Alessandro do Nascimento Vargas  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica  
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Av. Alberto Carazzai, 1640 - 86.300-000- Cornélio Procópio – PR.  
Tel. +55 (43) 3520-4007 / e-mail: [ppgee-cp@utfpr.edu.br](mailto:ppgee-cp@utfpr.edu.br) / [www.utfpr.edu.br/cornelioprocopio/ppgee](http://www.utfpr.edu.br/cornelioprocopio/ppgee)

## RESUMO

HOREVICZ, Daniel. ESTUDO DA TÉCNICA ANT COLONY LEARNING NO CONTROLE DE SISTEMAS NÃO-LINEARES. 59 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Os algoritmos bio-inspirados baseados em populações e meta-heurísticas têm se tornado uma opção atraente na resolução de problemas de busca e otimização, para os quais abordagens tradicionais apresentam dificuldades ou até mesmo impossibilidade de aplicação. O presente trabalho tem como objetivo verificar o desempenho de um destes algoritmos no campo de controle de sistemas não-lineares, cujo métodos analíticos não garantem soluções. Para tal, é feito o estudo de um método recentemente desenvolvido, chamado *Ant Colony Learning* (ACL), o qual é baseado no algoritmo de otimização por colônia de formigas (ACO, do inglês *Ant Colony Optimization*). O ACL é uma abordagem multiagente para aprendizado de políticas de controle. Para validar a análise da técnica, é feito o controle de dois sistemas: um pêndulo invertido e um sistema do tipo quadricóptero. Ao final deste trabalho, apresentam-se os resultados dos sistemas controlados e as observações obtidas com o uso desta técnica.

**Palavras-chave:** ACO, ACL, controle, sistemas não-lineares.

## LISTA DE FIGURAS

FIGURA 1	– Colônia de formigas <i>I. humilis</i> selecionando a menor ramificação nos dois módulos da ponte; a) um módulo da ponte, b) e c) esquemático representando o comportamento das formigas após 4 e 8 minutos. . . . .	11
FIGURA 2	– Exemplo do comportamento de um sistema com 3 entradas constantes distintas. . . . .	16
FIGURA 3	– Exemplo de partição equidistante do plano de fase em que $\mathbf{x}_{a1}$ pertence a partição A-7. . . . .	17
FIGURA 4	– Funções de pertinência em $\mathcal{X}^1$ . . . . .	18
FIGURA 5	– Representação esquemática de um pêndulo. . . . .	22
FIGURA 6	– Representação da característica espelhada no pêndulo. . . . .	23
FIGURA 7	– Política de Controle do pêndulo adotando particionamento $21 \times 21$ . . . . .	30
FIGURA 8	– Política de Controle do pêndulo adotando particionamento $31 \times 31$ . . . . .	30
FIGURA 9	– Política de Controle do pêndulo adotando particionamento $41 \times 41$ . . . . .	31
FIGURA 10	– Política de Controle do pêndulo adotando particionamento $51 \times 51$ . . . . .	31
FIGURA 11	– Política de Controle do pêndulo adotando particionamento $61 \times 61$ . . . . .	32
FIGURA 12	– Política de Controle para $\mathcal{U} = [-0.8, 0, 0.8]$ . . . . .	33
FIGURA 13	– Resposta e entrada do sistema controlado com estados iniciais nulos. . . . .	33
FIGURA 14	– Comportamento dos estados em um plano de fase com estados iniciais nulos. . . . .	34
FIGURA 15	– Resposta e entrada do sistema controlado com estados iniciais $[\frac{\pi}{3}, -3]'$ . . . . .	35
FIGURA 16	– Comportamento dos estados em um plano de fase com estados iniciais $[\frac{\pi}{3}, -3]'$ . . . . .	35
FIGURA 17	– Forças e torques que atuam sobre o quadricóptero . . . . .	36
FIGURA 18	– Esquemático do sistema de controle do quadricóptero. . . . .	40
FIGURA 19	– Política de Controle do quadricóptero adotando configuração de partição $21 \times 21$ . . . . .	47
FIGURA 20	– Política de Controle do quadricóptero adotando configuração de partição $31 \times 31$ . . . . .	48
FIGURA 21	– Política de Controle do quadricóptero adotando configuração de partição $41 \times 41$ . . . . .	48
FIGURA 22	– Política de Controle do quadricóptero adotando configuração de partição $51 \times 51$ . . . . .	49
FIGURA 23	– Política de Controle do quadricóptero adotando configuração de partição $61 \times 61$ . . . . .	49
FIGURA 24	– Política de Controle do quadricóptero utilizando $\rho = 0,01$ , $61 \times 61$ partições e $u_{max} = 2$ . . . . .	52
FIGURA 25	– Histograma do tempo de acomodação do quadricóptero. . . . .	53
FIGURA 26	– Resposta das posições angulares do sistema controlado. . . . .	54
FIGURA 27	– Resposta das velocidades angulares do sistema controlado. . . . .	55
FIGURA 28	– Velocidade angular dos motores do quadricóptero. . . . .	55

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
<b>2</b>	<b>OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS - ACO</b>	<b>10</b>
2.1	INSPIRAÇÃO E SURGIMENTO	10
2.2	MÉTODO E ALGORITMO	12
<b>3</b>	<b>MÉTODO ANT COLONY LEARNING - ACL</b>	<b>15</b>
3.1	CARACTERIZAÇÃO DE UM PROBLEMA	15
3.2	PARTICIONAMENTO DO ESPAÇO DE ESTADOS	17
3.2.1	Fuzzy	18
3.3	UTILIZAÇÃO DO ACO NO MÉTODO ACL	19
<b>4</b>	<b>PROBLEMA I: PÊNDULO INVERTIDO</b>	<b>22</b>
4.1	DESCRIÇÃO DO PROBLEMA	22
4.1.1	Característica Espelhada	23
4.2	SIMULAÇÕES	24
4.2.1	Varição da taxa de evaporação	25
4.2.2	Varição da divisão do espaço de estados	27
4.2.3	Simulação da resposta do pêndulo	32
<b>5</b>	<b>PROBLEMA II: QUADRICÓPTERO</b>	<b>36</b>
5.1	MODELAGEM	36
5.2	REDUÇÃO DO MODELO	39
5.3	APLICAÇÃO	40
5.3.1	Controle meta-heurístico	41
5.3.2	Controle Clássico	41
5.3.3	Switch	42
5.3.4	Conversão torque para rotação dos motores	42
5.4	SIMULAÇÃO	42
5.4.1	Varição da taxa de evaporação	43
5.4.2	Varição do número de partições	45
5.4.3	Varição no universo de ações da política de controle	50
5.4.4	Simulação da reposta do quadricóptero controlado	52
<b>6</b>	<b>CONCLUSÃO</b>	<b>56</b>
	<b>REFERÊNCIAS</b>	<b>58</b>

## 1 INTRODUÇÃO

Para que seja possível controlar um sistema necessita-se, dependendo da técnica de controle empregada, de um modelo matemático que descreva de forma adequada seu comportamento físico. Entretanto, dependendo do sistema considerado e das particularidades admitidas, pode-se obter modelos com diferentes níveis de complexidade. Deste modo, pode existir mais de um modelo matemático que represente determinado sistema (OGATA et al., 2003).

A maioria dos sistemas físicos apresenta características não lineares, o que dificulta a síntese de controladores. Isso ocorre pois os métodos clássicos de controle, aplicados geralmente em sistemas lineares, não podem ser empregados em sistemas não-lineares uma vez que o princípio da superposição não é satisfeito. Nestes casos, utilizam-se outras técnicas de controle. Uma das possíveis abordagens usualmente empregada é a linearização, que possibilita o uso das ferramentas aplicadas aos sistemas lineares. Porém, a linearização pode não ser suficiente, pois esta consiste em aproximar o sistema em torno de um ponto de operação, limitando a garantia da estabilidade a uma região local. Outra limitação é a existência de fenômenos como ciclo limite, harmônicos, caos, entre outros que ocorrem apenas na presença de não-linearidades e, portanto, não podem ser descritos por modelos lineares. Assim, torna-se interessante buscar estratégias de controle que propiciem uma análise confiável à categoria de sistemas não-lineares. Dentre os controles mais utilizados em sistemas não-lineares citam-se: controle adaptativo, controle robusto, fuzzy Takagi-Sugeno, entre outros (KHALIL, 1996).

No projeto de um controlador visa-se, além do controle, que o sistema atinja o desempenho estabelecido. Assim, é interessante, em alguns casos, buscar a solução ótima considerando determinado critério de desempenho como, por exemplo, minimizar o tempo necessário para que o sistema alcance o equilíbrio. Entretanto, nem sempre o cálculo da solução ótima é garantido, porém, na maioria das vezes uma solução sub-ótima é satisfatória e esta pode ser obtida por algoritmos de heurística e meta-heurística. No caso das meta-heurísticas, tem-se uma interação entre procedimentos de melhoria local e



uma estratégia que visa criar um processo capaz de escapar de mínimos locais e efetuar uma busca robusta pelo espaço de solução (GLOVER; KOCHENBERGER, 2006). A capacidade das meta-heurísticas em resolver problemas difíceis e complexos, de maneira aceitável e com um tempo razoável, fez com que estas ganhassem popularidade nas últimas duas décadas (TALBI, 2009).

Uma das meta-heurísticas que se provou efetiva para a busca de soluções ótimas é a Otimização por Colônia de Formigas, do inglês *Ant Colony Optimization* (ACO). Na literatura podem ser encontradas diversas aplicações de algoritmos ACO, como nos exemplos a seguir: problema do caixeiro viajante (DORIGO et al., 1996), problema generalizado de atribuição (PGA) (RAMALHINHO-LOURENÇO; SERRA, 1998), problema quadrático de alocação (PQA) (MANIEZZO; COLORNI, 1999), roteamento em redes de telecomunicações (ARAÚJO et al., 2004), problema da mochila multidimensional (ALAYA et al., 2004), problema de clique máximo em grafos (SOLNON; FENET, 2006), problema de roteirização de veículos (ÇATAY, 2010), escalonamento de tripulações (DENG; LIN, 2010). Além das aplicações acadêmicas, o ACO também é utilizado em aplicações comerciais (DORIGO; BIMTTARI, 2006). Uma das primeiras empresas a explorar algoritmos baseados na metaheurística ACO foi a EuroBios. Outra empresa que teve um papel muito importante na promoção da aplicação do ACO é AntOptima. Pesquisadores da AntOptima desenvolveram um conjunto de ferramentas para a solução de problemas de roteamento de veículos cujos algoritmos de otimização são baseados em ACO. Os produtos particularmente bem-sucedidos baseados nessas ferramentas são DYVOIL, para a gestão e otimização da distribuição de óleo de aquecimento com uma frota não homogênea de caminhões, usada pela primeira vez por Pina Petroli na Suíça, e AntRoute, para o encaminhamento de centenas de veículos de empresas como Migros, a principal cadeia suíça de supermercados, ou Barilla, a principal fabricante italiana de massas alimentícias. Ainda outra aplicação de roteamento de veículos foi desenvolvida pela BiosGroup para a empresa francesa Air Liquide.

A efetividade do ACO, mencionada anteriormente, somada as diversas variações advindas deste algoritmo despertou a curiosidade do autor sobre o uso de ferramentas meta-heurísticas na área de controle. Por meio de pesquisas notou-se que a maioria das aplicações do ACO em controle se resume na sintonia dos ganhos de controladores PID, como em: (HASSANZADEH; MOBAYEN, 2011; HSIAO et al., 2004; CHIHA et al., 2012).

Neste trabalho, utiliza-se uma abordagem pouco explorada chamada *Ant Colony*

*Learning* (ACL) (AST et al., 2008). Esta técnica visa alcançar soluções sub-ótimas de problemas de controle não-linear, em que o espaço de estados é tratado como um espaço no qual as formigas artificiais encontram o menor caminho entre qualquer posição inicial e a objetivo. Foi reproduzido o problema de levantamento e controle de um pêndulo invertido, o mesmo utilizado nos trabalhos que propuseram o ACL. Foi proposta também uma nova aplicação para o método onde objetivou-se controlar de atitude de um quadricóptero a partir de qualquer posição angular e de uma ampla faixa de velocidades angulares iniciais. Em ambos os problemas abordados os controladores foram validados via simulação com o software MATLAB/SIMULINK®.

A seguir no Capítulo 2 tem-se um breve resumo da otimização por colônia de formigas, a qual é a principal ferramenta do método ACL. No Capítulo 3 é descrito o funcionamento do ACL. Nos Capítulos 4 e 5 são descritos os problemas de controle do pêndulo invertido e do quadricóptero, respectivamente. Nestes capítulos, ainda, apresenta-se os resultados da solução encontrada por meio das simulações. Por fim, no Capítulo 6 é apresentada a conclusão deste trabalho.

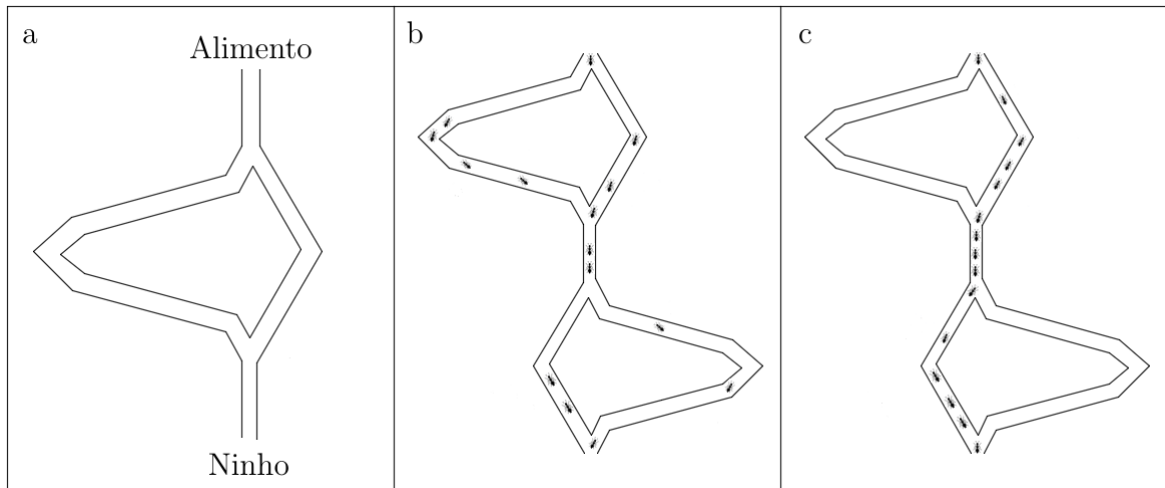
## 2 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS - ACO

### 2.1 INSPIRAÇÃO E SURGIMENTO

A Otimização por Colônia de Formigas (*Ant Colony Optimization* - ACO) é um modelo metaheurístico populacional. Como o próprio nome sugere, é inspirado pelo comportamento de colônias de formigas, mais especificamente na atuação da coleta de alimentos (ALOISE et al., 2002). O que torna este comportamento tão atrativo é a efetividade que as formigas apresentam ao encontrar a rota mais curta até o alimento, ainda que sua orientação seja limitada por sua pouca visão. Isto é possível graças à comunicação entre as formigas, que se dá por meio de uma substância química denominada feromônio, depositada pelas próprias formigas (MULATI et al., 2013). Enquanto as formigas caminham por uma trilha, inicialmente de forma aleatória, elas depositam uma certa quantidade de feromônios no solo. Assim, as próximas formigas tomam a decisão de seguir um caminho com probabilidade diretamente proporcional à quantidade de feromônio depositada anteriormente. Este comportamento é chamado de auto-catalítico por ser um processo que reforça a si mesmo (MULATI et al., 2013). O feromônio depositado tende a evaporar com o tempo, assim caminhos desvantajosos acabam sendo descartados no decorrer do tempo.

Em 1989, Deneubourg e colaboradores realizaram um experimento com as formigas argentinas *Iridomyrmex humilis*, observando o comportamento destas na aquisição de alimentos. O experimento conectava a colônia de formigas a uma fonte de alimentos por meio de uma estrutura composta por dois módulos idênticos posicionados em sentidos contrários, sendo cada módulo composto por dois caminhos com diferentes extensões, como mostrado na Figura 1. A princípio, cada formiga seguia um caminho exploratório aleatório (Figura 1b). Porém, com o passar do tempo as formigas, enquanto se deslocavam do ninho até a fonte de alimentos, formaram uma trilha química, de modo que as próximas formigas tendiam a escolher o caminho onde houvesse maior concentração química, funcionando como um *feedback* positivo. Como as formigas que escolheram o menor caminho faziam o percurso mais rapidamente, uma vez que já estavam voltando e depositando ainda mais feromônios, enquanto as outras ainda estavam completando a

primeira transição, elas depositavam uma maior quantidade de feromônio nesse caminho em relação ao mais extenso em um mesmo intervalo de tempo que as demais. Logo, em um determinado momento a intensidade do feromônio no caminho mais curto estava tão alta que quase todas as formigas seguiam por ele (Figura 1c). Mais detalhes sobre este experimento podem ser encontrados em (GOSS et al., 1989).



**Figura 1:** Colônia de formigas *I. humilis* selecionando a menor ramificação nos dois módulos da ponte; a) um módulo da ponte, b) e c) esquemático representando o comportamento das formigas após 4 e 8 minutos.

**Fonte:** Adaptado de (GOSS et al., 1989)

Inspirados pelos experimentos de coleta de alimentos por formigas M. Dorigo, V. Maniezzo e A. Colorni propuseram o *Ant System* (AS) (DORIGO et al., 1991), que foi aplicado para resolver o problema do caixeiro viajante (TSP, do inglês *Traveling Salesman Problem*). De modo geral, o TSP pode ser descrito como o problema de um vendedor que precisa sair de sua cidade e percorrer todas as cidades contidas em uma determinada área geográfica antes de retornar à sua cidade de origem, de modo que o percurso total seja mínimo e as cidades sejam visitadas apenas uma vez (MULATI et al., 2013). O algoritmo proposto para solucionar o TSP baseia-se em um grupo de formigas artificiais que liberam feromônio durante o seu trajeto e seguem trilhas de feromônio artificial para encontrar o menor caminho entre todas as cidades. Inicialmente três diferentes variantes foram propostas: *AS-density*, *AS-quantity* e *AS-cycle*, diferenciando-se pela maneira na qual as trilhas de feromônio eram atualizadas (DORIGO et al., 1991). A última abordagem mostrou-se mais eficiente e a maior parte dos algoritmos de otimização por colônia de formigas atuais derivam-se dela.

## 2.2 MÉTODO E ALGORITMO

Segundo (DORIGO; STÜTZLE, 2009) as formigas artificiais utilizadas no ACO são procedimentos de construção de soluções estocásticas, adicionando iterativamente componentes ou parcelas de soluções parciais. As soluções são construídas levando em consideração (i) informações heurísticas sobre o problema a ser resolvido e (ii) trilhas de feromônio (artificiais) que mudam conforme o tempo de execução, a fim de refletir a experiência adquirida pelas formigas durante sua busca.

As soluções parciais do problema são chamadas de estados. Cada formiga muda do seu estado atual para um próximo estado, que corresponde a uma solução parcial melhorada, com o objetivo de chegar ao estado final, que é a solução total do problema. Em cada passo da construção da solução a formiga visualiza o conjunto de expansões possíveis para a solução atual, isto é, identifica o conjunto de estados viáveis para o qual se pode direcionar a partir do estado atual (SERAPIÃO, 2009).

As formigas artificiais possuem uma estrutura de dados, chamada de lista tabu, que armazena os trechos já percorridos de modo a evitar que a formiga os repita até que o objetivo seja satisfeito (CARVALHO et al., 2007). Após a construção de todos os caminhos a intensidade de feromônio em cada aresta (trilha) é acrescida de modo proporcional à qualidade da solução gerada.

Um procedimento geral do ACO para aplicações em problemas de otimização combinatorial é mostrado no Algoritmo 1 (DORIGO; STÜTZLE, 2009). No algoritmo, é feita a inicialização dos parâmetros e das trilhas de feromônios para, em seguida, as  $M$  formigas construírem as soluções para o problema. Nesta busca por soluções consideram-se os feromônios e as informações heurísticas disponíveis. Após as soluções serem encontradas, é possível melhorar as mesmas por meio de uma busca local. Por fim, antes que a próxima iteração seja iniciada, a trilha de feromônios é atualizada, com o intuito de transmitir o aprendizado obtido pelas buscas das formigas.

---

### Algoritmo 1: PROCEDIMENTO GERAL DE UM ACO

---

- 1 Inicialização
  - 2 **enquanto** *Condição de término não alcançada* **faça**
  - 3     Construção das Soluções pelas Formigas
  - 4     Aplicação da Busca Local (opcional)
  - 5     Atualização dos Feromônios
  - 6 **fim**
-

A seguir, os principais procedimentos utilizados no ACO são expostos de forma mais detalhada, de acordo com (DORIGO; STÜTZLE, 2009):

**Inicialização:** No início do algoritmo, os parâmetros são definidos e todas as variáveis de feromônio são inicializadas no mesmo valor fixo definido por  $\tau_0$ .

**Construção das Soluções de Formigas:** Um conjunto de  $M$  formigas constrói soluções para o problema empregado partindo de posições iniciais aleatórias. Cada formiga começa com uma solução inicialmente vazia e, a cada passo de construção, esta aumenta sua solução parcial atual  $s_p$ , escolhendo um componente de solução viável  $c_i^j \in N(s_p)$  para adicionar à sua solução parcial, sendo  $c_i^j$  um componente de solução que leva a formiga do vértice  $i$  ao  $j$  e  $N(s_p)$  o conjunto que contém os componentes da solução que podem ser adicionados à solução parcial.

A escolha do componente de solução a ser adicionado é feita probabilisticamente por meio de uma regra em cada passo de construção. A regra mais utilizada é a do *Ant System*:

$$p_c\{c_i^j|s_p\} = \frac{\tau_{i,j}^\alpha \cdot [\eta(c_i^j)]^\beta}{\sum_{c_i^l \in N(s_p)} \tau_{i,l}^\alpha \cdot [\eta(c_i^l)]^\beta}, \quad \forall c_i^j \in N(s_p), \quad (1)$$

em que  $\tau_{i,j}$  é a quantidade de feromônio entre os vértices  $i$  e  $j$ ,  $\eta(\cdot)$  é a função que atribui a cada componente de solução viável  $c_i^j \in N(s_p)$  um valor heurístico, usualmente chamado de informação heurística. Essas são informações definidas previamente pelo sistema, tendo como intuito aumentar a velocidade de convergência ao resultado final. Um exemplo de função  $\eta(\cdot)$  para o problema do caixeiro viajante é dado por  $\eta(c_i^j) = \frac{1}{d}$ , em que  $d$  é a distância entre os vértices  $i$  e  $j$ . O valor da heurística neste caso é inversamente proporcional à distância dos vértices, dando preferência assim para a escolha de vértices mais próximos.

Os parâmetros  $\alpha$  e  $\beta$  determinam a influência da trilhas de feromônio e da informação heurística, respectivamente. Assim, se  $\alpha = 0$  a probabilidade de seleção é proporcional a  $[\eta(c_i^j)]^\beta$ , portanto, uma parcela de solução com alto valor heurístico terá uma maior chance de ser selecionada. Do mesmo modo, se  $\beta = 0$  a componente de solução com mais feromônios terá uma probabilidade maior de ser escolhida.

**Aplicação da Busca Local (opcional):** Após obter todas as soluções candidatas, pode-se efetuar uma busca local a fim de melhorar o desempenho obtido.

**Atualização dos Feromônios:** A atualização dos feromônios objetiva tornar os componentes que pertençam a boas soluções mais atrativos para a próxima iteração. Isto

é feito por meio de dois mecanismos, o depósito e a evaporação de feromônios. O primeiro visa aumentar o nível de feromônio das parcelas de solução que estejam associadas a um conjunto  $S_{upd}$  de boas soluções. O segundo mecanismo faz o decremento, com o passar do tempo, do feromônio depositado por formigas anteriores. A evaporação do feromônio faz-se necessária para evitar que o algoritmo convirja muito rápido e acabe em uma região sub-ótima. Este “esquecimento” do feromônio propicia a exploração de novas áreas do espaço de busca. Uma maneira de implementar a atualização de feromônio é por meio da regra

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \sum_{s \in S_{upd}} \Delta\tau_{i,j}(s), \quad \forall(i, j), \quad (2)$$

em que  $S_{upd}$  é o conjunto de soluções que são utilizadas para o depósito de feromônio,  $\rho \in (0, 1]$  é o parâmetro, constante em todas iterações, chamado de taxa de evaporação e  $\Delta\tau_{i,j}(s)$  é o depósito de feromônio em  $\tau_{i,j}$  referente à solução  $s$ , computado como:

$$\Delta\tau_{i,j}(s) = \begin{cases} F(s) & , \text{ se } (i, j) \in s, \\ 0 & , \text{ caso contrário,} \end{cases} \quad (3)$$

sendo  $F(s)$  a função *fitness* que define quão boa é a solução  $s$ .

### 3 MÉTODO ANT COLONY LEARNING - ACL

Nesta seção é analisado o ACL, um método que pode ser utilizado na resolução de problemas de controle, o qual se baseia na Otimização por Colônia de Formigas. Este método foi apresentado pelos artigos (AST et al., 2008, 2009) e mais tarde pela tese de doutorado (AST, 2010), sendo esta última referência a mais recente encontrada pelo autor.

#### 3.1 CARACTERIZAÇÃO DE UM PROBLEMA

Admita um sistema não-linear, cuja dinâmica é dada por

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \quad (4)$$

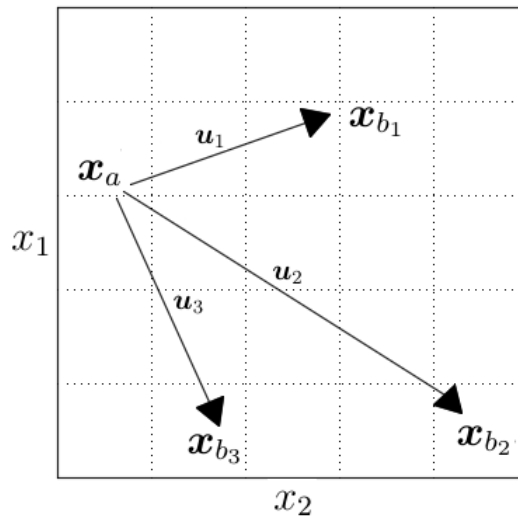
sendo  $\mathbf{x}(t) \in \mathcal{X}$  um vetor n-dimencional de estados, dado por

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad (5)$$

$\mathcal{X}$  o espaço de estados do sistema, e  $\mathbf{u}(t)$  um vetor de entradas pertencentes ao conjunto finito  $\mathcal{U}$ .

Em um dado período de tempo, o sistema com estado  $\mathbf{x}_a$  é levado para um novo estado  $\mathbf{x}_b$ , sendo aplicada a entrada externa  $\mathbf{u}$ . Na Figura 2 pode-se observar este comportamento em um sistema com duas dimensões em  $\mathcal{X}$ , considerando três valores constantes e diferentes para  $\mathbf{u}$ .





**Figura 2:** Exemplo do comportamento de um sistema com 3 entradas constantes distintas.

**Fonte:** Autoria própria

O problema de controle considerado consiste em determinar uma sequência de valores de  $\mathbf{u}(t)$  de forma a conduzir o sistema de um estado inicial qualquer para um estado final desejado, também chamado de referência (KHALIL, 1996). A abordagem apresentada neste trabalho consiste em usar o ACO para determinar a sequência de entradas, alternativamente denominada política de controle, que leva o sistema de qualquer estado inicial à referência pré-determinada, otimizando algum critério de desempenho.

De forma a aplicar o método ACL no problema proposto, é escolhido um período de tempo fixo, chamado de  $h$ , assim, representando o sistema de forma discreta como em

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)), \\ &\downarrow \\ \mathbf{x}[k+1] &= g(\mathbf{x}[k], \mathbf{u}[k]). \end{aligned} \tag{6}$$

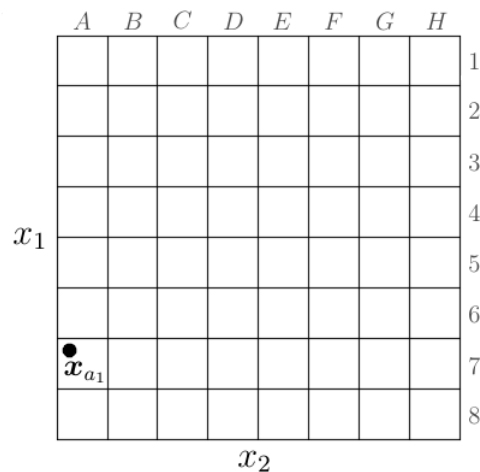
Com o período de tempo fixado, cada estado  $\mathbf{x}$  pode ser visualizado como um vértice de um grafo e cada possível combinação de  $(\mathbf{x}, \mathbf{u})$  como uma aresta, que liga os vértices  $\mathbf{x}[k]$  e  $\mathbf{x}[k+1]$ . Assim, o vértice representa a transição que ocorre de  $\mathbf{x}[k]$  para  $\mathbf{x}[k+1]$  quando uma entrada constante  $\mathbf{u}$  é aplicada. Objetiva-se então encontrar, a partir de qualquer estado inicial  $\mathbf{x}_0 \in \mathcal{X}$ , quais as arestas que fazem com que o sistema alcance o estado desejado  $\mathbf{x}_d$  em um menor número de passos. Para tanto, utiliza-se o algoritmo de otimização por colônia de formigas para encontrar os melhores caminhos, ou arestas. A partir da matriz de feromônio gerada pelo ACO é formada uma política de

controle, sendo que esta pode ser visualizada como uma tabela que dita qual entrada  $\mathbf{u}$ , pertencente a  $\mathcal{U}$ , deve ser aplicada a cada um dos possíveis estados  $\mathbf{x}$ .

### 3.2 PARTICIONAMENTO DO ESPAÇO DE ESTADOS

Deseja-se encontrar uma política de controle com a melhor entrada  $\mathbf{u} \in \mathcal{U}$  a ser aplicada ao sistema para cada estado  $\mathbf{x}$ , porém este pertence à  $\mathbb{R}^n$  e pode assumir infinitos valores. Para solucionar esse problema, particiona-se o espaço de estados em um número limitado de regiões, necessitando-se assim encontrar apenas uma política de controle para cada partição.

Um exemplo, em que o particionamento é feito de modo equidistante, pode ser verificado na Figura 3 com o estado  $\mathbf{x}_{a_1}$  pertencente à partição A-7.



**Figura 3:** Exemplo de partição equidistante do plano de fase em que  $\mathbf{x}_{a_1}$  pertence a partição A-7.

**Fonte:** Autoria própria

Cada partição é tratada como um vértice pelo ACO. Como consequência, um número elevado de partições pode acarretar em um problema de otimização com solução impraticável do ponto de vista computacional. No entanto, um número pequeno de partições pode resultar em uma representação pouco significativa do sistema. Uma maneira de aliviar esse dilema é a utilização da abordagem *fuzzy*.

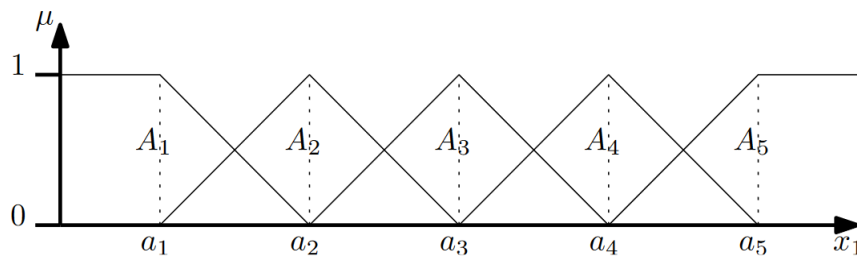
### 3.2.1 FUZZY

Os conjuntos e a lógica *fuzzy* são conceitos que podem ser empregados para tornar informações imprecisas, expressas por conjuntos de regras linguísticas, em termos matemáticos (TANSCHKEIT, 2004). Na teoria de conjuntos clássica, um elemento ou pertence a um conjunto ou não pertence. Dado um universo  $U$  e um elemento particular  $y \in U$ , o grau de pertinência  $\mu_A(y)$  com respeito a um conjunto  $A \subseteq U$  é dado por:

$$\mu_A(x) = \begin{cases} 1 & \text{se } y \in A, \\ 0 & \text{se } y \notin A. \end{cases} \quad (7)$$

Em 1965, Zadeh propôs uma caracterização mais ampla, sugerindo que alguns elementos são mais membros de um conjunto do que outros (TANSCHKEIT, 2004). Deste modo, o grau de pertinência pode então assumir qualquer valor entre 0 e 1.

Aplicando o conceito de conjuntos *fuzzy* nas partições, é possível reduzir o número de partições de forma que o sistema ainda possa ser representado significativamente. Com partições *fuzzy*, um estado  $\mathbf{x}$  deixa de pertencer necessariamente a apenas uma partição, passando a pertencer a todas com determinados graus de pertinência em cada uma delas. Estes graus de pertinência serão utilizados na escolha da entrada  $\mathbf{u}$  e na atualização dos feromônios, podendo ser vistos mais detalhadamente na Seção 3.3. A Figura 4 demonstra um exemplo de funções de pertinência em  $\mathcal{X}^1$ .



**Figura 4:** Funções de pertinência em  $\mathcal{X}^1$ .

**Fonte:** (AST, 2010)

Na Figura 4,  $A_i$  é uma função de pertinência para as variáveis de estado  $x_1$ , cujo centro é dado por  $a_i$ , sendo  $i = 1, \dots, N_1$ , sendo  $N_1$  o número de funções de pertinência para  $x_1$ . Em um momento discreto  $k$ , o grau de pertinência,  $\mu_{A_i}(x_1[k])$ , de um valor

específico do estado para  $A_i$  é calculado por:

$$\mu_{A_i}(x_1[k]) = \begin{cases} \max\left(0, \min\left(1, \frac{a_2 - x_1}{a_2 - a_1}\right)\right), & \text{se } i = 1, \\ \max\left(0, \min\left(\frac{x_1 - a_{i-1}}{a_i - a_{i-1}}, \frac{a_{i+1} - x_1}{a_{i+1} - a_i}\right)\right), & \text{se } i = 2, \dots, N_1 - 1, \\ \max\left(0, \min\left(\frac{x_1 - a_{N_1-1}}{a_{N_1} - a_{N_1-1}}, 1\right)\right), & \text{se } i = N_1. \end{cases} \quad (8)$$

Caso o sistema seja de ordem 2, calcula-se  $\mu_{B_j}(x_2[k])$  da mesma forma, sendo as funções de pertinência denominadas por  $B_j$ , com  $j = 1, \dots, N_2$ , em que  $N_2$  é o número de funções de pertinência para  $x_2$ , podendo ser expandido para ordens maiores da mesma forma.

Multiplicando-se os graus de pertinência, tem-se

$$\beta_{ij}(\mathbf{x}[k]) = \mu_{A_i}(x_1[k]) \cdot \mu_{B_j}(x_2[k]),$$

$$\boldsymbol{\beta}(\mathbf{x}[k]) = \begin{bmatrix} \beta_{11}(\mathbf{x}[k]) & \beta_{12}(\mathbf{x}[k]) & \cdots & \beta_{1N_2}(\mathbf{x}[k]) \\ \beta_{21}(\mathbf{x}[k]) & \beta_{22}(\mathbf{x}[k]) & \cdots & \beta_{2N_2}(\mathbf{x}[k]) \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{N_11}(\mathbf{x}[k]) & \beta_{N_12}(\mathbf{x}[k]) & \cdots & \beta_{N_1N_2}(\mathbf{x}[k]) \end{bmatrix}. \quad (9)$$

Caso maiores dimensões,  $d > 2$ , sejam consideradas, a matriz  $\boldsymbol{\beta}$  será multidimensional, ou seja,  $\boldsymbol{\beta} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_d}$ . Neste caso, cada elemento de tal matriz, denotado por  $\beta_{\mathbf{q}}(\mathbf{x}[k])$ , sendo  $\mathbf{q}$  cada uma das partições do conjunto considerado, consistirá no grau de pertinência do estado  $\mathbf{x}[k]$  em relação à partição  $\mathbf{q}$ .

### 3.3 UTILIZAÇÃO DO ACO NO MÉTODO ACL

Inicialmente, todas as  $M$  formigas são distribuídas pelo espaço de estados  $\mathcal{X}$  de forma que cada formiga fique atrelada a um diferente estado inicial  $\mathbf{x}_0$ . Cada formiga constrói, de forma paralela e independente, uma solução para o problema apresentado, sendo que nesta solução os valores subseqüentes de  $\mathbf{x}[k+1]$  são obtidos pela simulação de aplicação da entrada  $\mathbf{u}[k]$  ao sistema (4). A escolha da entrada  $\mathbf{u}[k]$  é sorteada, sendo as probabilidades calculadas com os valores de feromônio  $\tau_{\mathbf{q},\mathbf{u}}$ . Existe um valor  $\tau_{\mathbf{q},\mathbf{u}}$  associado a cada par de partição-ação  $(\mathbf{q}, \mathbf{u})$ , todos começam no mesmo valor inicial  $\tau_0$ , sendo este positivo, e são incrementados e decrementados conforme o treinamento é realizado.

A probabilidade da ação  $\mathbf{u} \in \mathcal{U}$  ser aplicada no estado  $\mathbf{x}$  é calculada pela equação

$$p\{\mathbf{u}|\mathbf{x}\} = \sum_{\mathbf{q} \in \mathcal{Q}} \beta_{\mathbf{q}}(\mathbf{x}) \frac{\tau_{\mathbf{q},\mathbf{u}}^{\alpha}}{\sum_{\ell \in \mathcal{U}} \tau_{\mathbf{q},\ell}^{\alpha}}, \quad (10)$$

em que  $\alpha$  é o parâmetro associado ao feromônio,  $\mathcal{U}$  é o conjunto de ações disponíveis e  $p\{\mathbf{u}|\mathbf{x}\}$  é a probabilidade da formiga escolher a ação  $\mathbf{u}$  no estado  $\mathbf{x}$ .

As ações  $\mathbf{u}$  de cada formiga  $\zeta \in \mathcal{F}$ , sendo  $\mathcal{F}$  o conjunto de todas as formigas, são armazenadas juntamente com o estado  $\mathbf{x}$ , no qual estas foram aplicadas, no multiconjunto

$$\mathcal{S}_{\zeta} = \{(\mathbf{x}_1, \mathbf{u}_1), (\mathbf{x}_2, \mathbf{u}_2) \dots (\mathbf{x}_n, \mathbf{u}_n)\}. \quad (11)$$

A cada iteração do algoritmo, cada formiga  $\zeta$  vai construindo a sua solução  $\mathcal{S}_{\zeta}$  até que: a formiga alcance um estado suficientemente próximo a  $\mathbf{x}_d$ ; ou o número máximo de passos  $T$ , pré definido, seja atingido. O conjunto das soluções  $\mathcal{S}_{\zeta}$  que alcançam um estado suficientemente próximo a  $\mathbf{x}_d$  é definido como  $\mathcal{S}_{suces}$ .

Após as soluções de todas as formigas serem completadas, é computada a qualidade da solução das formigas que obtiveram sucesso. Esta é calculada levando em consideração o número de passos  $T_k$  necessários para alcançar o objetivo, de modo que um número menor de passos resulta em um maior valor de *fitness*. A regra utilizada pode ser descrita por

$$F(\zeta) = \begin{cases} \frac{1}{\sqrt{(T_k - (1-h))h}} & \text{se } \mathcal{S}_{\zeta} \in \mathcal{S}_{suces}, \\ 0 & \text{caso contrário,} \end{cases} \quad (12)$$

em que  $h$  é o tempo de amostragem.

Em seguida, ocorre a evaporação e o incremento na matriz de feromônios como em

$$\tau_{\mathbf{q},\mathbf{u}} \leftarrow \tau_{\mathbf{q},\mathbf{u}}(1 - \rho) + \sum_{\zeta \in \mathcal{F}} \Delta\tau_{\mathbf{q},\mathbf{u}}(\zeta), \quad \forall \mathbf{q} \in \mathcal{Q}, \quad \forall \mathbf{u} \in \mathcal{U}, \quad (13)$$

sendo

$$\Delta\tau_{\mathbf{q},\mathbf{u}}(\zeta) = \sum_{\mathbf{x}: (\mathbf{x},\mathbf{u}) \in \mathcal{S}_{\zeta}} \beta_{\mathbf{q}}(\mathbf{x}) F(\zeta). \quad (14)$$

Este processo é repetido até que o número máximo de iterações pré-definido seja alcançado. Ao final, a política de controle é definida por

$$u(\mathbf{x}) = \arg \max_{\ell \in \mathcal{U}} (\tau_{\mathbf{q},\ell}), \quad (15)$$

sendo

$$\mathbf{q} = \arg \max_{\mathbf{q} \in \mathcal{Q}} (\beta_{\mathbf{q}}(\mathbf{x})). \quad (16)$$

O Algoritmo 2 demonstra um pseudo código do procedimento ACL utilizado neste trabalho.

---

**Algoritmo 2:** PROCEDIMENTO ACL UTILIZADO NESTE TRABALHO

---

```

1 Inicialização das variáveis
2 enquanto Número de iterações não atingido faça
3   para Cada formiga  $\zeta$  faça em paralelo
4     Formiga recebe  $\mathbf{x}_0$ 
5     enquanto Número de passos máximos  $T$  não atingido faça
6       Cálculo de  $\beta(\mathbf{x}[k])$  segundo equação (9)
7       Seleção de ação  $\mathbf{u}$  segundo equação (10)
8       Cálculo de  $\mathbf{x}[k + 1]$ 
9       Armazenamento de  $\mathbf{x}[k]$  e  $\mathbf{u}$  em  $\mathcal{S}_\zeta$ 
10      se  $\mathbf{x}[k + 1]$  esta suficientemente próximo de  $\mathbf{x}_d$  então
11        Quebra de loop
12      fim
13    fim
14  fim
15  para Todas as formigas  $\zeta$  faça
16    Cômputo da qualidade da solução segundo equação (12)
17    Atualização de Feromônios segundo equações (13) e (14)
18  fim
19 fim
20 Política de controle definida segundo equações (15) e (16)

```

---

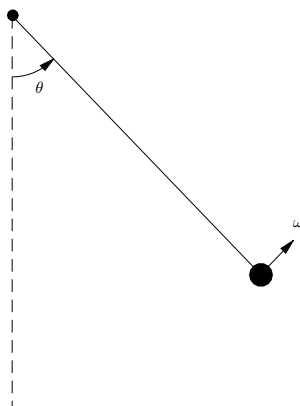
Note que a referência pré-determinada é fixada em um único estado  $\mathbf{x}_d$ , portanto o controlador resultante só será capaz de levar os estados do sistema para um único estado desejado.

## 4 PROBLEMA I: PÊNULO INVERTIDO

Com o intuito de validar a implementação do algoritmo ACL, aplica-se este ao mesmo problema proposto no artigo (AST et al., 2008), que consiste no controle de um pêndulo invertido.

### 4.1 DESCRIÇÃO DO PROBLEMA

Deseja-se realizar o levantamento e a estabilização na posição ereta de um pêndulo, sendo o mesmo modelado como uma haste ligada a um ponto pivô na qual um motor exerce um torque. Como o motor não possui torque suficiente para elevar o pêndulo em um único movimento à posição desejada, é necessário balançar o pêndulo de um lado para o outro, fazendo-o seguir uma trajetória oscilatória, e assim, eventualmente, acumular a energia necessária para alcançar o estado desejado, neste caso quando a posição angular do pêndulo esta próxima de  $\pm\pi$  e a velocidade angular próxima de 0.



**Figura 5:** Representação esquemática de um pêndulo.

**Fonte:** Autoria própria

A Figura 5 ilustra o sistema, sendo os estados a posição angular  $\theta$  e a velocidade angular  $\omega$ . O sistema representado é não-linear e de segunda ordem, e as equações

diferenciais que o regem podem ser descritas por:

$$\begin{aligned}\dot{\theta} &= \omega, \\ J\dot{\omega} &= Ku - mgR\sin(\theta) - D\omega,\end{aligned}\tag{17}$$

sendo o torque,  $u$ , a entrada do sistema que é definida positiva no sentido de rotação anti-horário. Os demais parâmetros estão descritos na Tabela 1.

#### 4.1.1 CARACTERÍSTICA ESPELHADA

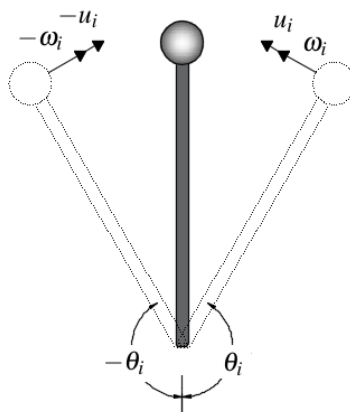
O pêndulo invertido utilizado é um sistema que possui uma característica espelhada, sendo para

$$\mathbf{x}_b = f(\mathbf{x}_a, \mathbf{u}_a),\tag{18}$$

verdade que

$$-\mathbf{x}_b = f(-\mathbf{x}_a, -\mathbf{u}_a).\tag{19}$$

O uso desta característica permite uma redução no tempo de treinamento do ACL. A Figura 6 ilustra uma ação de controle  $u_i$  aplicada nos estados  $\mathbf{x}_i = [\theta_i \ \omega_i]'$ . Com base nas equações (18) e (19) é possível intuir que uma ação de controle  $-u_i$  aplicada nos estados  $-\mathbf{x}_i = [-\theta_i \ -\omega_i]'$ , resultará em um desempenho similar ao anterior.



**Figura 6:** Representação da característica espelhada no pêndulo.

**Fonte:** Autoria própria

Desta forma, o mesmo incremento de feromônio referente ao par estado-ação  $(\mathbf{x}_i, u_i)$  pode também ser utilizado no incremento do par estado-ação  $(-\mathbf{x}_i, -u_i)$ .



## 4.2 SIMULAÇÕES

As simulações são realizadas no software MATLAB<sup>®</sup> e utiliza-se a função ODE45, baseada explicitamente em uma fórmula da família Runge-Kutta, chamada Dormand-Prince (SHAMPINE; REICHEL, 1997; DORMAND; PRINCE, 1980).

Os valores das constantes do sistema utilizados nas simulações, Tabela 1, são os mesmos que os empregados em (AST et al., 2008).

**Tabela 1:** Constantes utilizadas nas simulações. (AST et al., 2008)

Constante	Valor	Descrição
$J$	$0,005 \text{ kg.m}^2$	inércia do braço
$K$	$0,1$	ganho do motor
$D$	$0,01 \text{ kg.s}^{-1}$	amortecimento ( <i>damping</i> )
$m$	$0,1 \text{ kg}$	massa
$g$	$9,81 \text{ m.s}^{-2}$	aceleração gravitacional
$R$	$0,1 \text{ m}$	distância de meio braço

Foram realizados diversos treinamentos e testes de forma a encontrar os melhores parâmetros  $\rho$ , utilizado na equação (13), e a quantidade de divisões  $N_1$  e  $N_2$  dos estados  $x_1$  e  $x_2$ , respectivamente, que levam ao número de partições do espaço de estados para o problema do pêndulo. Para avaliar a qualidade de um treinamento verifica-se o número de passos necessários para o sistema alcançar o estado objetivo considerando diversos estados iniciais, sendo desejado um menor número de passos.

Todos os treinamentos tiveram um tempo de amostragem  $h = 100$  milissegundos, com um número máximo  $T = 300$  passos para cada formiga encontrar o objetivo, resultando em uma simulação de no máximo 30 segundos. O número de iterações foi 50 para todos os treinamentos e foi considerado como possíveis ações de controle  $u \in \mathcal{U} = \{-0,8; 0; 0,8\}$ .

Para todos os treinamentos o valor de  $\alpha$  é igual a 1 e o número de formigas é proporcional ao número de partições escolhidas, sendo que cada uma das formigas possui um estado inicial  $\mathbf{x}_0$  aleatório pertencente a uma partição. Neste caso, em que o sistema possui característica espelhada, existem partições que podem ser desconsideradas como

estado inicial, pois sua versão espelhada já foi considerada.

#### 4.2.1 VARIÇÃO DA TAXA DE EVAPORAÇÃO

Primeiramente, foi estudado o efeito da variação da taxa de evaporação  $\rho$  na qualidade do treinamento. Os valores de  $\rho$  considerados foram: 0,001; 0,01; 0,1; 0,3 e 0,7. O espaço de estados foi dividido de forma equidistante, sendo utilizado nestes treinamentos  $N_1 = N_2 = 41$  resultando em 1681 partições do espaço de estados.

Para cada opção de  $\rho$  foram feitos 5 treinamentos, resultando num total de 25 diferentes políticas de controle, e estas foram submetidas a testes com um grupo de 1517 estados iniciais, definidos por

$$\mathbf{x}_0 = \begin{bmatrix} \theta_0 \\ \omega_0 \end{bmatrix}, \quad (20)$$

tais que

$$\theta_0 \in \{0, 15k_1 : k_1 \in \mathbb{Z}, -19 < k_1 < 19\} \quad (21)$$

e

$$\omega_0 \in \{0, 45k_2 : k_2 \in \mathbb{Z}, -21 < k_2 < 21\}. \quad (22)$$

Na Tabela 2 pode-se observar o resultado da aplicação de cada uma das 25 políticas de controle a cada um dos estados iniciais. Foi computado o tempo médio e seu desvio padrão, a partir de todos os estados iniciais, necessário para que o pêndulo alcançasse a posição ereta. Considerou-se o tempo em que o sistema entrou, e se manteve, na faixa de  $|\theta| > 2,7$  radianos.

**Tabela 2:** Tempo computacional de treinamento, tempo médio de acomodação e desvio padrão obtidos com diferentes políticas de controle, analisando distintas taxas de evaporação.

$\rho$	tempo de treinamento (s)	tempo médio (s)	desvio padrão
0,001	6286	1,2106	1,6257
0,001	6321	1,2328	1,6691
0,001	6486	1,1933	1,5878
0,001	6154	1,2015	1,6295
0,001	6363	1,2139	1,6081
0,01	5931	1,2196	1,6066
0,01	6086	1,1934	1,6254
0,01	6151	1,2185	1,6835
0,01	6699	1,1907	1,6211
0,01	6036	1,2108	1,6280
0,1	5625	1,2140	1,6582
0,1	5594	1,2093	1,5984
0,1	5396	1,2681	1,7763
0,1	5306	1,2585	1,7252
0,1	5533	1,2798	1,7285
0,3	4843	1,2177	1,6052
0,3	5070	1,2754	1,6544
0,3	5110	1,2234	1,6050
0,3	5142	1,2332	1,6137
0,3	5180	1,2026	1,6263
0,7	4729	1,2127	1,5986
0,7	4810	1,2674	1,6632
0,7	4566	1,2435	1,6150
0,7	4581	1,2100	1,6285
0,7	4731	1,2286	1,6138

A Tabela 3 apresenta as médias dos tempos médios de cada opção de evaporação  $\rho$ . Nesta, pode-se observar que os treinamentos com  $\rho = 0,01$  obtiveram, com uma pequena margem, um melhor tempo médio acomodação e, portanto, este valor foi escolhido para os treinamentos seguintes.

**Tabela 3:** Média do tempo computacional de treinamento e média do tempo médio de acomodação obtidas para cada distinta taxa de evaporação.

$\rho$	média do tempo de treinamento (s)	média do tempo médio (s)
0,001	6322	1,2104
0,01	6181	1,2066
0,1	5491	1,2459
0,3	5069	1,2304
0,7	4683	1,2324

#### 4.2.2 VARIAÇÃO DA DIVISÃO DO ESPAÇO DE ESTADOS

Posteriormente, foram feitos treinamentos utilizando  $\rho = 0,01$  e variando o número de partições do espaço de estados com  $N_1 \times N_2$  assumindo as configurações:  $21 \times 21$ ,  $31 \times 31$ ,  $41 \times 41$ ,  $51 \times 51$  e  $61 \times 61$ .

Para cada configuração de  $N_1 \times N_2$ , foram treinadas 5 políticas de controle e por meio destas foi obtido, via simulação, o tempo médio necessário para alcançar a posição ereta a partir dos mesmos estados iniciais definidos em (20), (21) e (22). Os resultados obtidos são apresentados na Tabela 4.

**Tabela 4:** Tempo computacional de treinamento, tempo médio de acomodação e desvio padrão obtidos com diferentes políticas de controle, analisando distintas configurações de particionamento.

$N_1 \times N_2$	tempo de treinamento (s)	tempo médio (s)	desvio padrão
21 × 21	2829	1,2801	1,7188
21 × 21	2444	2,5660	2,8696
21 × 21	2508	2,6414	2,9835
21 × 21	2802	1,2778	1,6516
21 × 21	2481	2,5583	2,8438
31 × 31	4134	1,2047	1,6249
31 × 31	4526	1,2124	1,6008
31 × 31	4389	1,2430	1,6417
31 × 31	4255	1,2594	1,7007
31 × 31	4233	1,2188	1,6071
41 × 41	5931	1,2196	1,6066
41 × 41	6086	1,1934	1,6254
41 × 41	6151	1,2185	1,6835
41 × 41	6699	1,1907	1,6211
41 × 41	6036	1,2108	1,6280
51 × 51	8121	1,3114	1,8817
51 × 51	8541	1,2100	1,5972
51 × 51	8558	1,2235	1,7340
51 × 51	8095	1,1842	1,5865
51 × 51	8613	1,1803	1,5867
61 × 61	11383	1,2038	1,6718
61 × 61	11076	1,1628	1,5798
61 × 61	11662	1,1725	1,5823
61 × 61	11173	1,1725	1,6070
61 × 61	11083	1,1739	1,6370

Na Tabela 5, observou-se que conforme é incrementado o número de partições, implicando em um maior número de formigas, o tempo computacional necessário para o treinamento também aumenta. Pode-se verificar também que, quando  $N_1 \times N_2 = 21 \times 21$ , o tempo médio de acomodação do pêndulo na posição ereta aumenta de forma significativa, demonstrando que, para este problema, esta quantia de partições não é

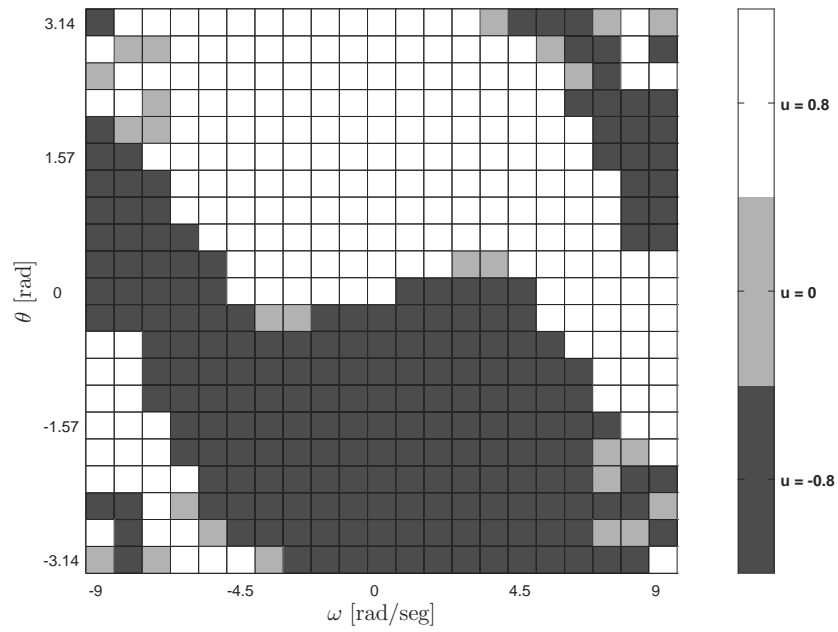
uma aproximação suficiente. Um número maior de partições trouxe uma melhor média de tempo de acomodação e um tempo maior de treinamento, o que faz sentido intuitivamente. A configuração escolhida para se utilizar foi  $61 \times 61$ , pois esta teve o menor tempo médio de acomodação dentre as configurações testadas.

**Tabela 5:** Média do tempo computacional de treinamento e média do tempo médio de acomodação obtidas para cada distinta configuração de particionamento.

$N_1 \times N_2$	média do tempo de treinamento (s)	média do tempo médio (s)
$21 \times 21$	2613	2,0647
$31 \times 31$	4307	1,2276
$41 \times 41$	6181	1,2066
$51 \times 51$	8386	1,2219
$61 \times 61$	11275	1,1771

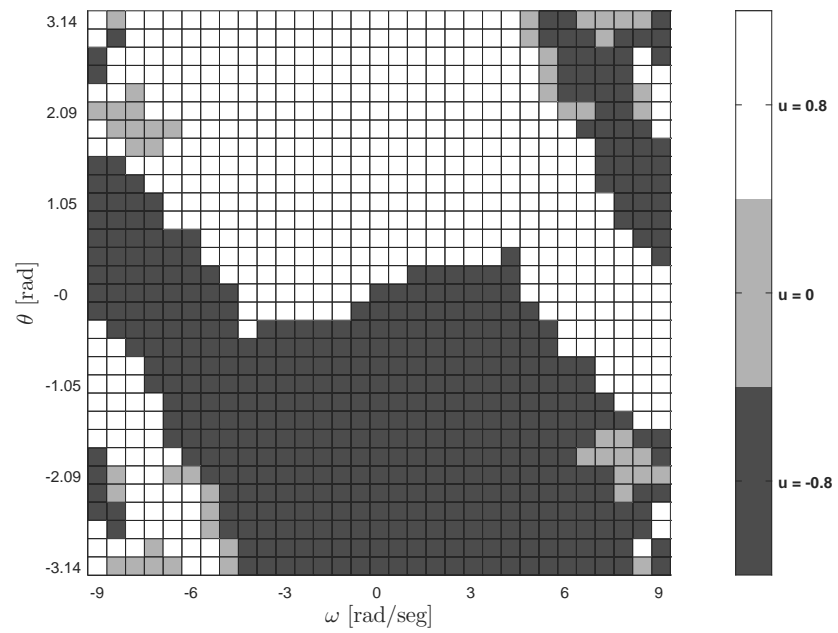
Nas Figuras 7-11, é possível verificar políticas de controle com diferentes quantias de partições. No eixo horizontal está representada a velocidade angular  $\omega$  e no eixo vertical é representada a posição angular  $\theta$ . Cada quadrado representa uma partição  $\mathbf{q}$ , sendo que sua cor define qual torque  $u$  deve ser aplicado. Por exemplo, caso os estados do sistema em determinado instante estivessem em torno de  $\theta = -1,5$  e  $\omega = 3$ , para qualquer uma das políticas de controle obtida, a ação de controle  $u$  seria de  $-0,8N.m$ , pois estas coordenadas estão na área preta.

É possível notar uma simetria causada pelo uso da característica espelhada no treinamento, sendo que caso um estado  $\mathbf{x}_i$  pertença a uma partição branca na política de controle ( $u = 0,8$ ), o estado  $-\mathbf{x}_i$  estará em uma partição preta ( $u=-0,8$ ).



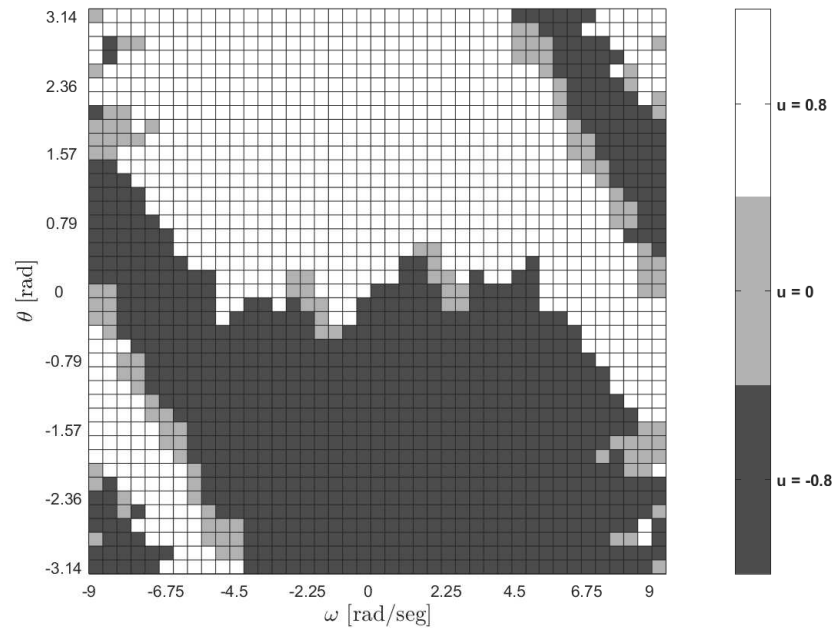
**Figura 7:** Política de Controle do pêndulo adotando particionamento  $21 \times 21$ .

**Fonte:** Autoria própria



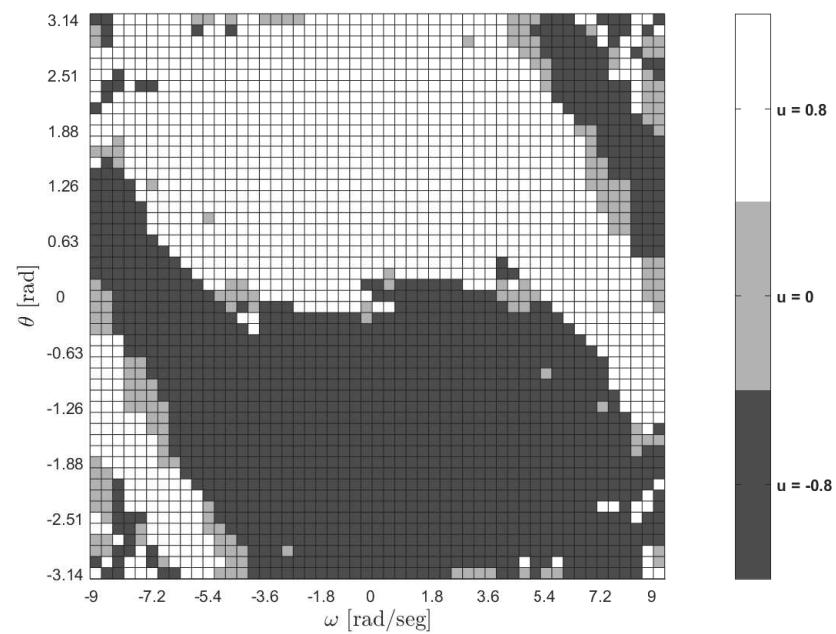
**Figura 8:** Política de Controle do pêndulo adotando particionamento  $31 \times 31$

**Fonte:** Autoria própria



**Figura 9:** Política de Controle do pêndulo adotando particionamento  $41 \times 41$

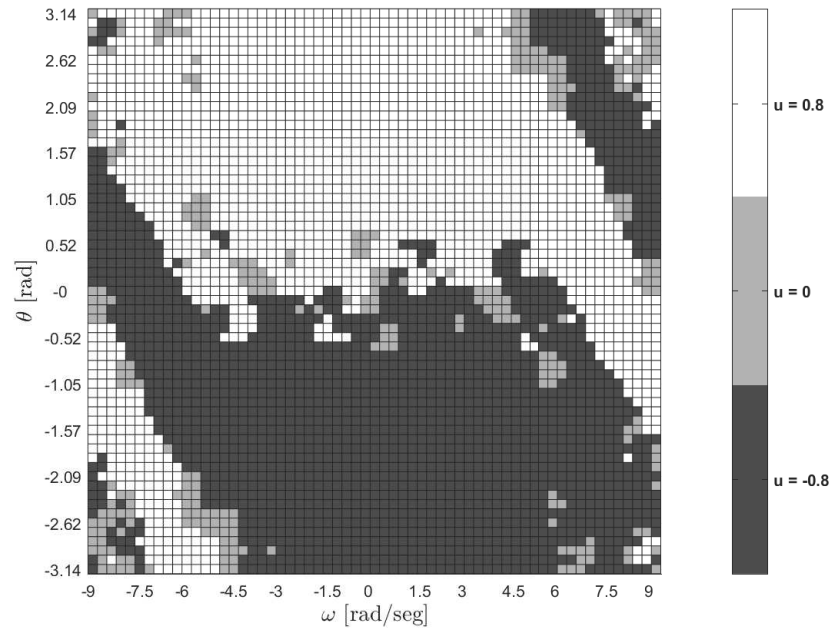
**Fonte:** Autoria própria



**Figura 10:** Política de Controle do pêndulo adotando particionamento  $51 \times 51$

**Fonte:** Autoria própria



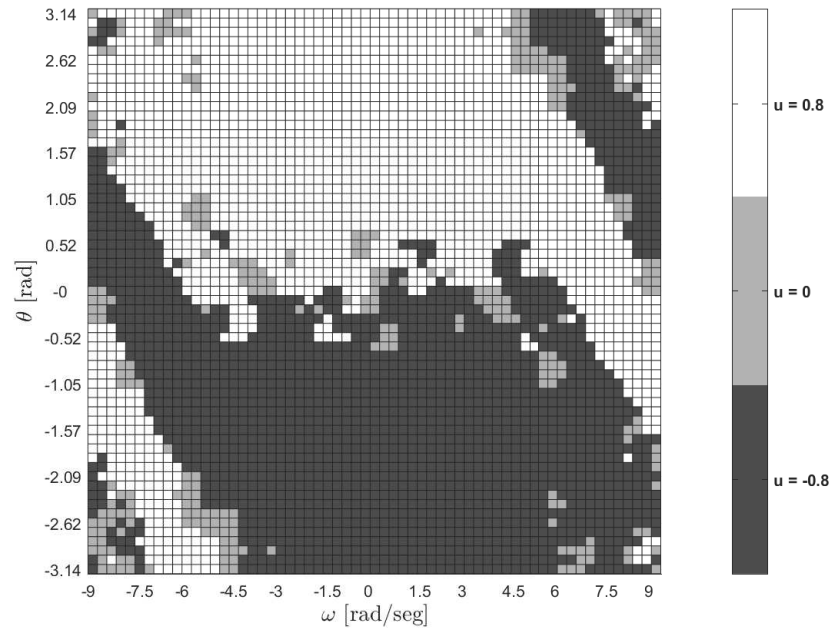


**Figura 11:** Política de Controle do pêndulo adotando particionamento  $61 \times 61$

**Fonte:** Autoria própria

#### 4.2.3 SIMULAÇÃO DA RESPOSTA DO PÊNDULO

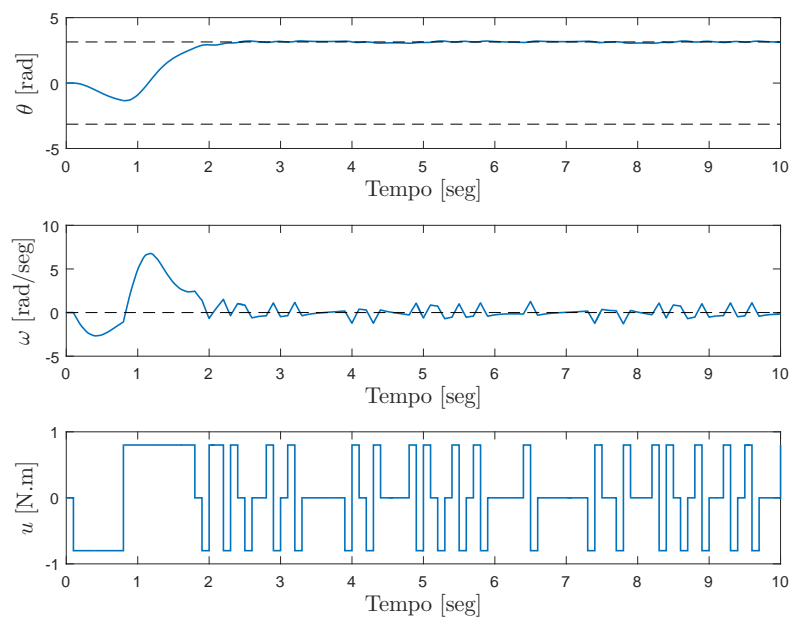
Nesta seção, é demonstrada a simulação do pêndulo invertido no SIMULINK, utilizando uma política de controle treinada com os valores da taxa de evaporação  $\rho$  e a configuração de particionamento  $N_1 \times N_2$  que apresentaram os melhores resultados anteriormente. Utiliza-se  $\rho = 0,01$ ,  $N_1 \times N_2 = 61 \times 61$  e os demais parâmetros definidos na seção 4.2. A política de controle utilizada é mostrada na Figura 12.



**Figura 12:** Política de Controle para  $\mathcal{U} = [-0.8, 0, 0.8]$ .

**Fonte:** Autoria própria

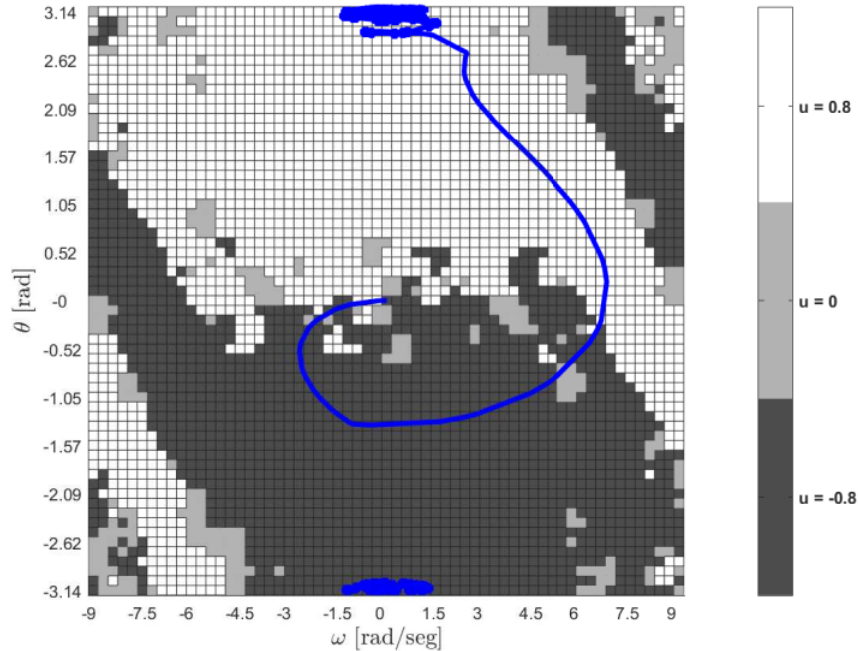
A resposta do sistema controlado com estado inicial  $\mathbf{x}_0 = [0, 0]'$  é mostrada na Figura 13, na qual pode-se verificar que o sistema alcança a posição angular almejada  $\theta = \pm\pi$  em aproximadamente 2 segundos. O comportamento da velocidade angular  $\omega$  e o torque aplicado ao motor  $u$  também são apresentados.



**Figura 13:** Resposta e entrada do sistema controlado com estados iniciais nulos.

**Fonte:** Autoria própria

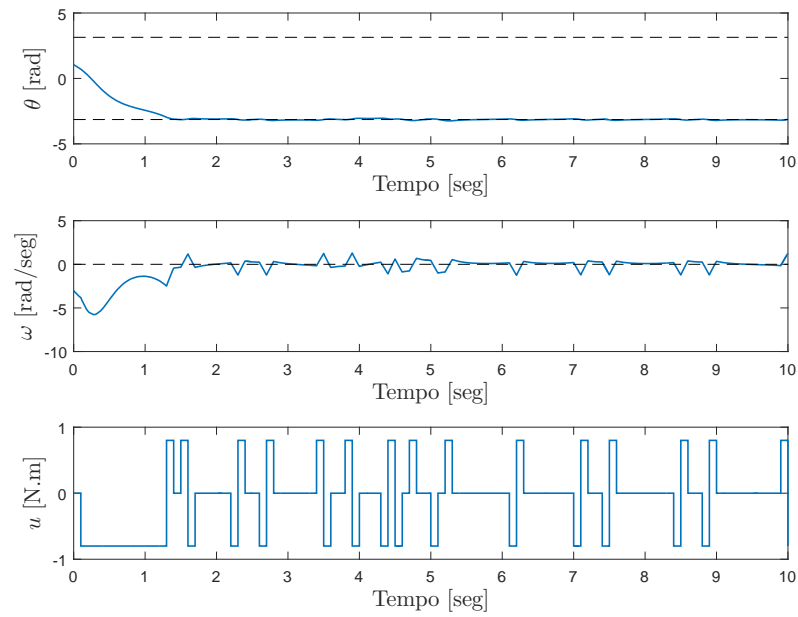
Na Figura 14 apresenta-se o comportamento dos estados do sistema em um diagrama de fase. A linha azul simboliza o trajeto dos estados do sistema, começando no estado  $[0, 0]'$  e terminando em torno de  $[\pi, 0]'$ .



**Figura 14:** Comportamento dos estados em um plano de fase com estados iniciais nulos.

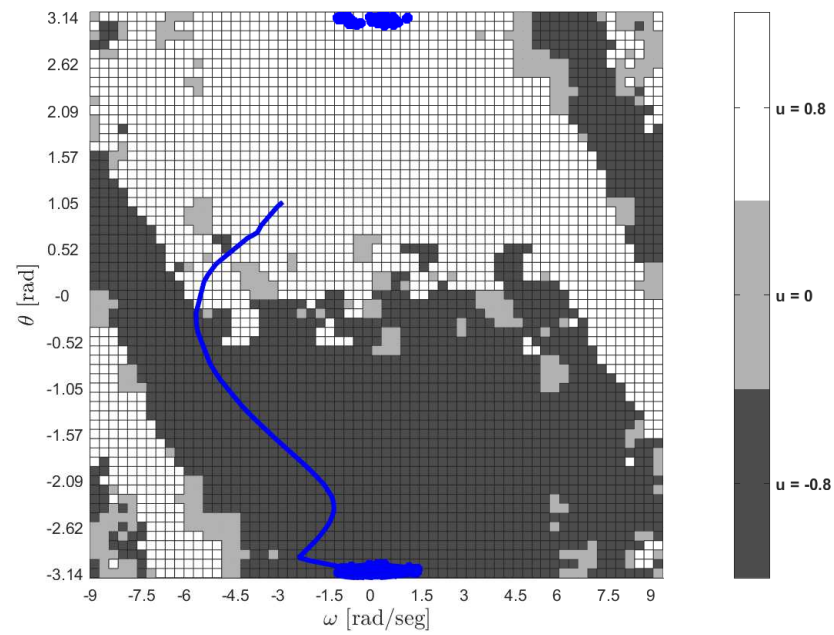
**Fonte:** Autoria própria

Nas Figuras 15 e 16 pode-se observar o comportamento do sistema com os estados iniciais em  $[\frac{\pi}{3}, -3]'$ . Foi apresentado o mesmo problema proposto no artigo (AST et al., 2008) e os resultados obtidos foram coerentes e bem parecidos com os do artigo original.



**Figura 15:** Resposta e entrada do sistema controlado com estados iniciais  $[\frac{\pi}{3}, -3]'$ .

**Fonte:** Autoria própria



**Figura 16:** Comportamento dos estados em um plano de fase com estados iniciais  $[\frac{\pi}{3}, -3]'$ .

**Fonte:** Autoria própria

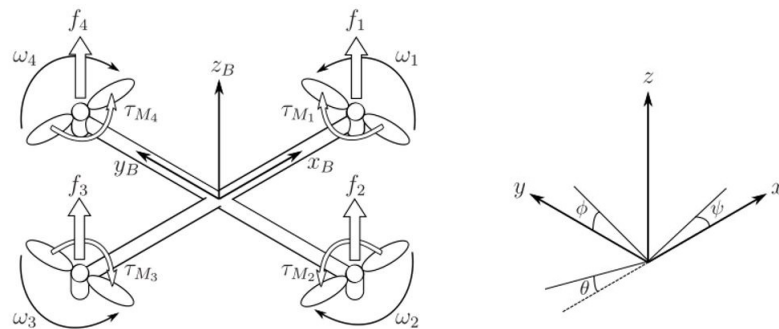
## 5 PROBLEMA II: QUADRICÓPTERO

Um quadricóptero pode ser modelado como um sistema de 12<sup>a</sup> ordem, sendo os estados do sistema tanto as posições, como velocidades lineares e angulares. Neste problema objetiva-se realizar apenas o controle das posições e velocidades angulares, portanto tem-se um sistema de 6<sup>a</sup> ordem. No método ACL é feita uma busca das melhores ações de controle para cada partição, sendo cada uma delas uma combinação de estados. Em um caso de 6<sup>a</sup> ordem existiria um número excessivamente elevado de partições, portanto escolheu-se utilizar uma versão simplificada do modelo, de segunda ordem, no treinamento do ACL. Embora tenha sido feita uma grande simplificação, esta só foi utilizada em uma etapa do processo e na simulação do controle foi utilizado o sistema de ordem completa.

A Seção 5.1 descreve o modelo matemático de um quadricóptero retirado diretamente da literatura (LUUKKONEN, 2011). A Seção 5.2 demonstra a simplificação utilizada para o treinamento do ACL e a Seção 5.4 mostra a simulação e os resultados.

### 5.1 MODELAGEM

Na Figura 17 tem-se a estrutura de um quadricóptero incluindo as velocidades angulares, os torques e as forças criadas pelos quatro motores.



**Figura 17:** Forças e torques que atuam sobre o quadricóptero

**Fonte:** (LUUKKONEN, 2011)

A posição linear absoluta do quadricóptero em relação ao eixo inercial  $x, y, z$  é definida por  $\boldsymbol{\xi}$ . A atitude, isto é, a posição angular, é definida por  $\boldsymbol{\eta}$  no referencial inercial com os três ângulos de Euler  $\phi, \theta$  e  $\psi$ . O ângulo de rolagem (*roll*)  $\phi$  determina a rotação do quadricóptero em torno do eixo  $x$ . O ângulo de arfagem (*pitch*)  $\theta$  estabelece a rotação em torno do eixo  $y$ . O ângulo de guinada (*yaw*)  $\psi$  indica a rotação em torno do eixo  $z$ .

$$\boldsymbol{\xi} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \quad (23)$$

A origem do referencial do corpo é o centro de massa do quadricóptero. No referencial do corpo, as velocidades angulares são determinadas por  $\boldsymbol{\nu}$

$$\boldsymbol{\nu} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (24)$$

A matriz de rotação do referencial do corpo para o referencial inercial é dada por

$$\mathbf{R} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \quad (25)$$

sendo  $c_\phi \triangleq \cos(\phi)$  e  $s_\phi \triangleq \sin(\phi)$ . Como a matriz de rotação  $\mathbf{R}$  é ortogonal, tem-se  $\mathbf{R}^{-1} = \mathbf{R}^T$  que é a matriz de rotação do referencial inercial para o referencial do corpo.

A matriz de transformação para as velocidades angulares do referencial inercial para o referencial do corpo é  $\mathbf{W}_\eta$ , e do referencial do corpo para o inercial é  $\mathbf{W}_\eta^{-1}$

$$\begin{aligned} \dot{\boldsymbol{\eta}} = \mathbf{W}_\eta^{-1} \boldsymbol{\nu}, \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \\ \boldsymbol{\nu} = \mathbf{W}_\eta \dot{\boldsymbol{\eta}}, \quad \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \end{aligned} \quad (26)$$

em que  $t_\theta \triangleq \tan(\theta)$ . A matriz  $\mathbf{W}_\eta$  é inversível se  $\theta \neq (2k - 1)\phi/2$ ,  $k \in \mathbb{Z}$ . Assume-se que o quadricóptero tem sua estrutura simétrica. Ademais, a matriz de inércia é a matriz

diagonal  $\mathbf{I}$  na qual  $I_{xx} = I_{yy}$

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (27)$$

A velocidade angular do rotor  $i$ , denotada por  $w_i$ , cria uma força  $f_i$  na direção do eixo do rotor. A velocidade angular e a aceleração do rotor também criam o torque  $\tau_{M_i}$  em volta do eixo do motor.

$$f_i = k\omega_i^2, \quad \tau_{M_i} = b\omega_i^2 + I_M\dot{\omega}_i, \quad (28)$$

em que  $k$  é a constante de *lift*,  $b$  é a constante de *drag* e  $I_M$  é o momento de inércia do motor. Normalmente, o efeito de  $\dot{\omega}_i$  é considerado pequeno, sendo então desconsiderado.

A combinação das forças dos rotores cria a propulsão  $T$  na direção do eixo  $z$  do quadricóptero. O torque  $\boldsymbol{\tau}_B$  é composto pelos torques  $\tau_\phi$ ,  $\tau_\theta$  e  $\tau_\psi$  na direção correspondente aos ângulos do referencial do corpo.

$$T = \sum_{i=1}^4 f_i = k \sum_{i=1}^4 \omega_i^2, \quad \mathbf{T}^B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \quad (29)$$

$$\boldsymbol{\tau}_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} \ell k(-\omega_2^2 + \omega_4^2) \\ \ell k(-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^4 \tau_{M_i} \end{bmatrix}, \quad (30)$$

sendo  $\ell$  a distância entre o rotor e o centro de massa do quadricóptero.

Como assume-se que o quadricóptero é um corpo rígido, as equações de Newton-Euler podem ser utilizadas para descrever sua dinâmica. No referencial inercial, a força centrífuga é anulada. Assim, apenas a força gravitacional e a magnitude e direção da propulsão contribuem para a aceleração do quadricóptero.

$$m\dot{\boldsymbol{\xi}} = G + RT_B, \quad (31)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi s_\theta c_\phi - c_\psi s_\theta \\ c_\theta c_\psi \end{bmatrix}.$$

No referencial do corpo, a aceleração angular da inércia  $\mathbf{I}\dot{\boldsymbol{\nu}}$  e as forças centrípetas

$\boldsymbol{\nu} \times (\mathbf{I}\boldsymbol{\nu})$  são iguais ao torque externo  $\boldsymbol{\tau}$

$$\begin{aligned} \mathbf{I}\dot{\boldsymbol{\nu}} + \boldsymbol{\nu} \times (\mathbf{I}\boldsymbol{\nu}) &= \boldsymbol{\tau}, \\ \dot{\boldsymbol{\nu}} &= \mathbf{I}^{-1} \left( - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} p \\ I_{yy} q \\ I_{zz} r \end{bmatrix} - I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_\tau + \boldsymbol{\tau} \right), \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} (I_{yy} - I_{zz}) qr / I_{xx} \\ (I_{zz} - I_{xx}) pr / I_{yy} \\ (I_{xx} - I_{yy}) pq / I_{zz} \end{bmatrix} + \begin{bmatrix} \tau_\phi / I_{xx} \\ \tau_\theta / I_{yy} \\ \tau_\psi / I_{zz} \end{bmatrix}. \end{aligned} \quad (32)$$

## 5.2 REDUÇÃO DO MODELO

Como demonstrado na Seção 5.1,  $\boldsymbol{\eta}$  e  $\boldsymbol{\nu}$  resultam em um sistema de sexta ordem, quando não consideradas as posições e as velocidades lineares. Assim, tem-se como estados as posições angulares  $\phi$ ,  $\theta$  e  $\psi$  e as velocidades angulares  $p$ ,  $q$  e  $r$ . Os torques  $\tau_\phi$ ,  $\tau_\theta$  e  $\tau_\psi$  compõem a entrada do sistema.

A equação diferencial de  $\dot{r}$ , dada em (32), pode ser reescrita como

$$\dot{r} = \frac{\tau_\psi}{I_{zz}}, \quad (33)$$

quando  $I_{xx} = I_{yy}$ .

Considerando que se possui total controle sobre  $\tau_\psi$ , é possível então controlar a velocidade angular  $r$ . Estabilizando  $r$  em zero, é possível reescrever as equações diferenciais de  $\dot{p}$  e  $\dot{q}$ , mostradas em (32), de modo que estas também fiquem apenas em função dos torques  $\tau_\phi$  e  $\tau_\theta$ . Assim, tem-se:

$$\dot{p} = \frac{\tau_\phi}{I_{xx}}, \quad \dot{q} = \frac{\tau_\theta}{I_{yy}}. \quad (34)$$

Tendo-se total controle sobre  $\tau_\phi$  e  $\tau_\theta$ , os valores de  $p$  e  $q$  podem assumir qualquer valor de referência, contanto que a velocidade angular  $r$  seja mantida nula.

A posição angular  $\psi$  pode ser desconsiderada para nosso objetivo, uma vez que esta não interfere na estabilidade almejada pois deseja-se que as hélices do quadricóptero estejam apontadas para cima, algo que depende exclusivamente de  $\phi$  e  $\theta$ . Assim, descon-



siderando  $\psi$ , obtém-se o seguinte modelo simplificado,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & \text{sen}(\phi)\text{tan}(\theta) \\ 0 & \text{cos}(\phi) \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix}, \quad (35)$$

sendo  $\phi$  e  $\theta$  os estados e  $p$  e  $q$  as entradas. Este modelo é de segunda ordem e será utilizado pelo ACL para o treinamento da política de controle.

### 5.3 APLICAÇÃO

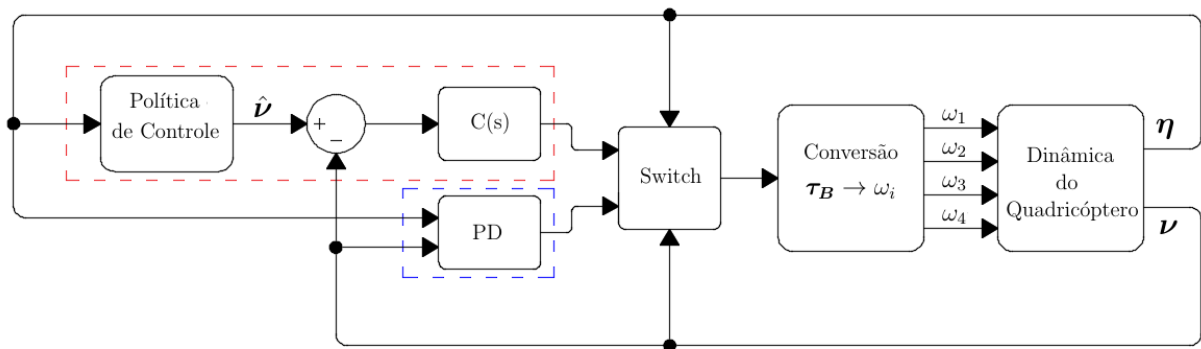
O objetivo do método de controle é levar o quadricóptero para uma posição angular onde o eixo  $z_B$  esteja próximo do eixo  $z$ , Figura 17, ou seja, com a força de propulsão voltada para cima. Este objetivo é satisfeito quando

$$\phi, \theta \in \{2k\pi : k \in \mathbb{Z}\}, \quad (36)$$

ou quando

$$\phi, \theta \in \{(2k + 1)\pi : k \in \mathbb{Z}\}. \quad (37)$$

Para isto, utiliza-se o sistema de controle mostrado no esquemático da Figura 18. Este é dividido em duas etapas, o controle meta-heurístico, em vermelho, e o controle clássico, em azul. Nesta estratégia o bloco *switch* é responsável por selecionar qual controlador será utilizado, fazendo uma transição suave entre estes. Ao final é feita a conversão de torque para velocidade angular nos rotores.



**Figura 18:** Esquemático do sistema de controle do quadricóptero.

**Fonte:** Autoria própria

### 5.3.1 CONTROLE META-HEURÍSTICO

O controle meta-heurístico é composto por uma política de controle, sendo esta treinada com o sistema simplificado (35), utilizando o método ACL. O objetivo da mesma é ditar qual velocidade angular melhor leva os estados da posição angular  $\phi$  e  $\theta$  para perto do objetivo, definido em (36) e (37). Além disso, há também um controlador que leva a velocidade angular atual  $\nu$  para o valor de referência determinado pela política de controle, definido por

$$\hat{\nu} = \begin{bmatrix} \hat{p} \\ \hat{q} \\ \hat{r} \end{bmatrix}, \quad (38)$$

em que os valores de  $\hat{p}$  e  $\hat{q}$  pertencem ao conjunto discreto  $\mathcal{U}$ , predefinido no treinamento da política de controle, e o valor de  $\hat{r}$  é em todos instantes nulo, pois o objetivo é zerar  $r$  para que o sistema se comporte de forma parecida com o sistema simplificado (35).

Foi inserido no controlador um filtro passa baixa, dimensionado empiricamente, com o intuito de obter uma curva com variação suave no sinal de controle e simular uma resposta mais próxima da realidade. O filtro utilizado é definido pela função de transferência

$$C(s) = \frac{29,4217}{s^2 + 33,0232s + 808,7361}. \quad (39)$$

### 5.3.2 CONTROLE CLÁSSICO

Um sistema controlado por políticas de controle advindas do método ACL apresenta respostas com características on/off. Assim, em determinados sistemas é interessante que se faça o uso conjunto de técnicas de controle. Neste trabalho, utiliza-se o controlador meta-heurístico para tratar as não-linearidades do sistema, levando o sistema para próximo dos estados desejados. Posteriormente, faz-se uso do controlador PD para que se mantenha a referência alcançada.

O controlador PD utilizado, juntamente com seus ganhos, também foi retirado de (LUUKKONEN, 2011), definido como

$$\tau_\phi = (K_D(\dot{\phi}_d - \dot{\phi}) + K_P(\phi_d - \phi))I_{xx}, \quad (40)$$

$$\tau_\theta = (K_D(\dot{\theta}_d - \dot{\theta}) + K_P(\theta_d - \theta))I_{yy}, \quad (41)$$

sendo  $K_P$  o ganho proporcional e  $K_D$  o ganho derivativo.

### 5.3.3 SWITCH

Quando as posições angulares  $\boldsymbol{\eta}$  estão fora de uma faixa próxima ao objetivo definido, utiliza-se o controle meta-heurístico, e quando estas estão dentro desta faixa, utiliza-se o controle clássico. A transição entre estes é feita utilizando funções sigmóides, de modo que o presente sinal de controle começa a diminuir, ao passo em que o novo sinal de controle vai aumentando e após um curto período de tempo, tem-se apenas o novo sinal de controle.

### 5.3.4 CONVERSÃO TORQUE PARA ROTAÇÃO DOS MOTORES

A partir dos controladores, tem-se a ação de controle  $\boldsymbol{\tau}_B$  e por meio desta, obtêm-se as rotações angulares dos motores segundo a equação

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} k & k & k & k \\ 0 & -\ell k & 0 & \ell k \\ -\ell k & 0 & \ell k & 0 \\ -b & b & -b & b \end{bmatrix}^{-1} \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}, \quad (42)$$

cujo o desenvolvimento ocorre a partir das equações (29) e (30) ajustando os valores da equação de  $\tau_\psi$  de forma que todos os rotores tenham o mesmo referencial de rotação. Considera-se para o cálculo que  $T = 0$ .

Embora na equação (42) seja possível obter valores de  $\omega_i^2 \in \mathbb{R}_*^-$ , todas as rotações dos motores devem ser positivas. Deste modo, quando obtém-se um ou mais valores negativos de  $\omega_i^2$  é necessário tornar  $\omega_i^2 \in \mathbb{R}^+$ . Assim, verifica-se qual é o  $\min\{\omega_i^2 \in \mathbb{R}_*^- : i = 1, 2, 3 \text{ e } 4\}$  para então somar seu módulo à todos os  $\omega_i^2$ . Por exemplo, caso sejam encontrados os valores  $\omega_1^2 = -1$ ,  $\omega_2^2 = 2$ ,  $\omega_3^2 = 3$  e  $\omega_4^2 = -6$ , é somado 6 em todos  $\omega_i^2$ . Deste modo, o impulso  $T$  deixa de ser nulo, porém, os valores dos torques continuam os mesmos, visto que aplica-se o mesmo incremento a todas variáveis  $\omega_i^2$ .

## 5.4 SIMULAÇÃO

As simulações do quadricóptero também são realizadas no software MATLAB<sup>®</sup> com a função ODE45. Os valores das constantes do sistema, utilizados nas simulações, Tabela 6, são os mesmos empregados em (LUUKKONEN, 2011).

Foram treinadas diversas políticas de controle variando os valores de  $\rho$ , das con-

**Tabela 6:** Parâmetros estruturais do quadricóptero utilizados na simulação. (LUUKKONEN, 2011)

Parâmetro	Valor	Unidade
$g$	9.81	$m/s^2$
$m$	0.468	$kg$
$\ell$	0.225	$m$
$k$	$2.980 \cdot 10^{-6}$	-
$b$	$1.140 \cdot 10^{-7}$	-
$I_{xx}$	$4.856 \cdot 10^{-3}$	$kg \ m^2$
$I_{yy}$	$4.856 \cdot 10^{-3}$	$kg \ m^2$
$I_{zz}$	$8.801 \cdot 10^{-3}$	$kg \ m^2$

figurações de  $N_1 \times N_2$  e do universo de ações da política de controle  $\mathcal{U}$ . Para todos os treinamentos das políticas de controle aplicadas ao problema do sistema quadricóptero, consideraram-se  $\tau_0 = 1$ ,  $\alpha = 1$  e 50 iterações no máximo. O tempo de amostragem utilizado foi de 100 milissegundos, com um número máximo de 30 passos para cada formiga encontrar o objetivo, resultando em tempo de simulação máximo de 3 segundos. O número de formigas utilizado é igual a quantidade de partições.

#### 5.4.1 VARIACÃO DA TAXA DE EVAPORAÇÃO

Foram treinadas cinco políticas de controle para cada opção de  $\rho$  no conjunto 0,001; 0,01; 0,1; 0,3 e 0,7; utilizando  $41 \times 41$  partições e  $\mathcal{U}$  composto por todas as combinações de  $(\hat{p}, \hat{q}, \hat{r})$  com  $\hat{p}$  e  $\hat{q}$  podendo assumir os valores  $-2, 0$  e  $2$ ; e  $\hat{r}$  sempre igual a 0. Para todas as 25 políticas de controle, verificou-se o tempo gasto para que cada um dos 2197 estados iniciais considerados, definido em (43) e (44), atingisse a faixa de estados desejados, definida em (36) e (37), admitindo uma faixa de erro de  $\pm 0,4$  radianos.

$$\boldsymbol{\eta}_0 = \begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \end{bmatrix}, \quad \boldsymbol{\nu}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (43)$$

$$\phi_0, \theta_0, \psi_0 \in \{0, 45k : k \in \mathbb{Z}, -7 < k < 7\}. \quad (44)$$

Na Tabela 7 pode-se observar o tempo de treinamento, a média do tempo que cada política levou para atingir o objetivo e o desvio padrão.

**Tabela 7:** Tempo computacional de treinamento, tempo médio de acomodação e desvio padrão obtidos com diferentes políticas de controle, analisando distintas taxas de evaporação.

$\rho$	tempo de treinamento (s)	tempo médio (s)	desvio padrão
0,001	397	1,1329	0,4569
0,001	397	1,0195	0,4587
0,001	401	1,0486	0,4218
0,001	400	1,1277	0,4796
0,001	420	1,1005	0,4464
0,01	404	1,0789	0,4465
0,01	398	1,0687	0,4193
0,01	394	1,0250	0,4461
0,01	398	1,0050	0,4126
0,01	396	1,0570	0,4588
0,1	356	1,0417	0,4398
0,1	364	1,0159	0,4155
0,1	357	1,0641	0,4300
0,1	357	1,0782	0,4130
0,1	355	1,0960	0,3957
0,3	304	1,0781	0,4185
0,3	301	1,1218	0,4096
0,3	308	1,0678	0,3845
0,3	301	1,0068	0,4281
0,3	303	1,1296	0,4192
0,7	254	1,0991	0,4185
0,7	253	1,1009	0,4024
0,7	252	1,0793	0,4023
0,7	262	1,1320	0,3999
0,7	258	1,0361	0,4310

**Tabela 8:** Média do tempo computacional de treinamento e média do tempo médio de acomodação obtidas para cada distinta taxa de evaporação.

$\rho$	média tempo de treinamento (s)	tempo médio(s)
0,001	403	1,0858
0,01	398	1,0469
0,1	358	1,0592
0,3	303	1,0808
0,7	256	1,0895

Pode-se observar uma diferença pequena entre as performances das cinco diferentes configurações de  $\rho$ , sendo que  $\rho = 0,01$  obteve um resultado um pouco melhor. Assim, será utilizado este valor nos próximos treinamentos.

#### 5.4.2 VARIAÇÃO DO NÚMERO DE PARTIÇÕES

Desta vez, treinou-se novas políticas de controle utilizando  $\rho = 0,01$ , o mesmo  $\mathcal{U}$  utilizado anteriormente e variando o número de partições  $N_1 \times N_2$ . As configurações testadas foram definidas arbitrariamente:  $21 \times 21$ ,  $31 \times 31$ ,  $41 \times 41$ ,  $51 \times 51$  e  $61 \times 61$ , sendo os valores ímpares de modo que o estado zero fosse o centro de uma partição ao invés de uma borda de duas partições. Para cada configuração foi feito o treinamento de cinco políticas de controle, e a simulação dessas com os estados iniciais, (43) e (44), definidos anteriormente. O tempo médio para que todos os estados iniciais atinjam os estados almejados, com uma faixa de tolerância de  $\pm 0,4$  radianos, pode ser observado na Tabela 9, na qual também é mostrado o desvio padrão e tempo de treinamento.

**Tabela 9:** Tempo computacional de treinamento, tempo médio de acomodação e desvio padrão obtidos com diferentes políticas de controle, analisando distintas configurações de particionamento.

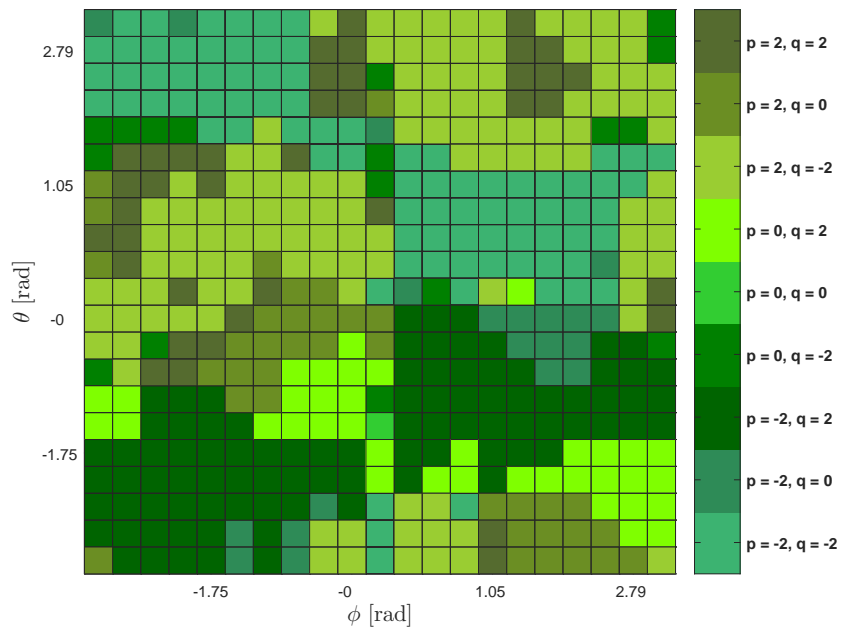
$N_1 \times N_2$	tempo de treinamento (s)	tempo médio (s)	desvio padrão
21 × 21	108	1,1162	0,4041
21 × 21	129	1,0934	0,4232
21 × 21	116	1,0109	0,3942
21 × 21	156	1,0994	0,3756
21 × 21	111	1,0147	0,4309
31 × 31	251	1,0610	0,4124
31 × 31	230	1,1543	0,4405
31 × 31	245	1,1285	0,4490
31 × 31	246	1,0954	0,4123
31 × 31	253	1,1094	0,4708
41 × 41	404	1,0789	0,4465
41 × 41	398	1,0687	0,4193
41 × 41	394	1,0250	0,4461
41 × 41	398	1,0050	0,4126
41 × 41	396	1,0570	0,4588
51 × 51	726	1,0656	0,4871
51 × 51	691	0,9759	0,3974
51 × 51	693	1,0811	0,4341
51 × 51	706	1,0512	0,4231
51 × 51	648	0,9887	0,4141
61 × 61	1026	0,9533	0,4209
61 × 61	980	1,0798	0,4792
61 × 61	928	1,0209	0,4100
61 × 61	975	1,0696	0,4267
61 × 61	988	1,0228	0,4612

Na Tabela 10 pode-se observar as médias referentes aos 5 tipos de partições testadas, novamente, nota-se uma diferença pequena entre as performances das configurações com a configuração  $61 \times 61$  apresentando o menor tempo médio.

**Tabela 10:** Média do tempo computacional de treinamento e média do tempo médio de acomodação obtidas para cada distinta configuração de particionamento.

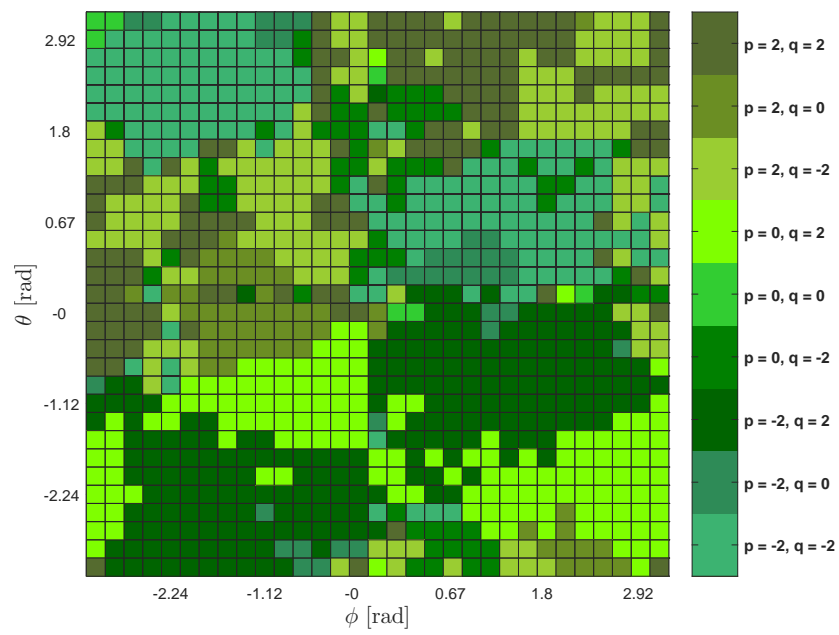
$N_1 \times N_2$	média do tempo de treinamento (s)	média do tempo médio (s)
$21 \times 21$	124	1,0669
$31 \times 31$	245	1,1097
$41 \times 41$	398	1,0469
$51 \times 51$	693	1,0325
$61 \times 61$	979	1,0292

Nas Figuras 19-23, são mostradas as políticas de controle para diferentes configurações de particionamento, representadas na Tabela 10. Cada partição possui uma tonalidade de verde, sendo que cada tonalidade representa uma opção de  $\hat{\nu}$ . Analisando ainda as figuras, nota-se um padrão na disposição das cores que se repete nas diferentes políticas, o que é coerente visto que, as configurações apresentadas tiveram uma diferença sutil em suas performances.

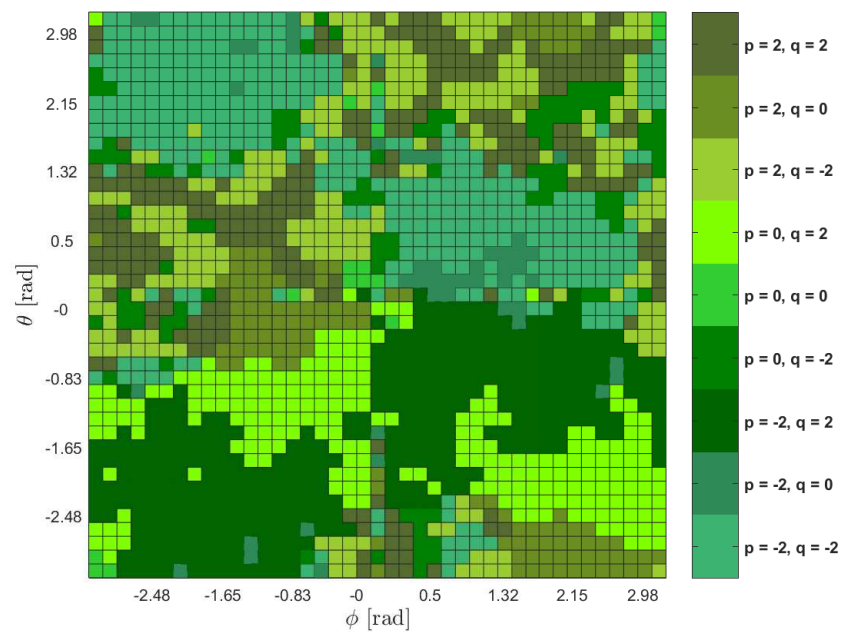


**Figura 19:** Política de Controle do quadricóptero adotando configuração de partição  $21 \times 21$ .

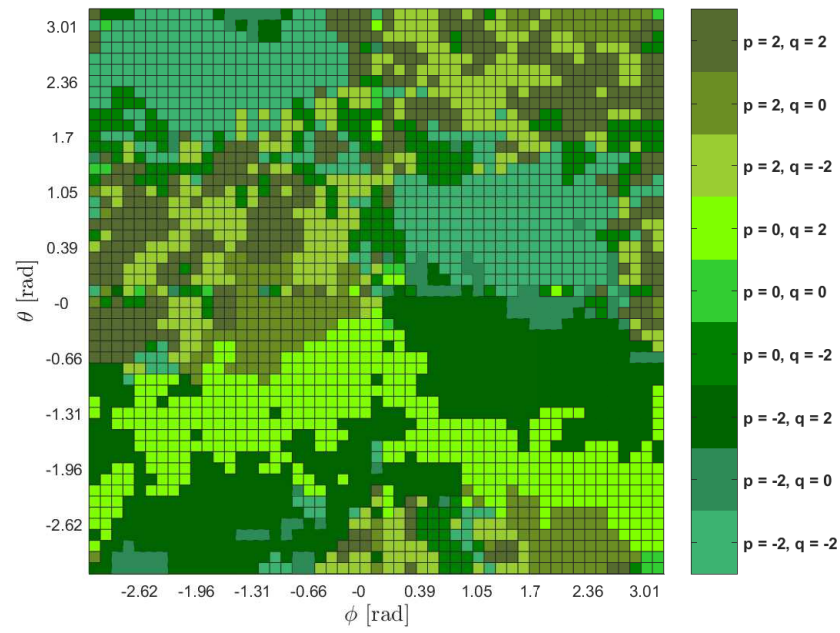




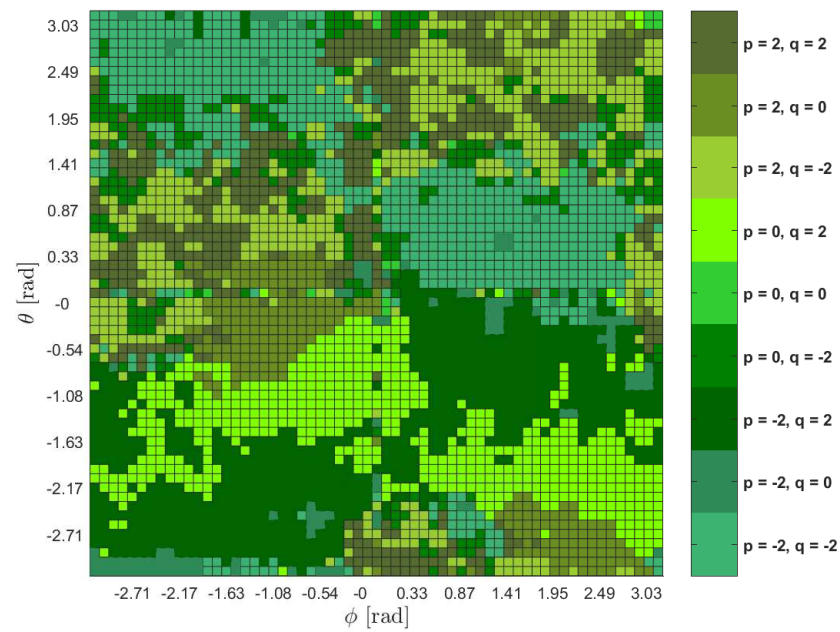
**Figura 20:** Política de Controle do quadricóptero adotando configuração de partição  $31 \times 31$ .



**Figura 21:** Política de Controle do quadricóptero adotando configuração de partição  $41 \times 41$ .



**Figura 22:** Política de Controle do quadricóptero adotando configuração de partição  $51 \times 51$ .



**Figura 23:** Política de Controle do quadricóptero adotando configuração de partição  $61 \times 61$ .

### 5.4.3 VARIACÃO NO UNIVERSO DE AÇÕES DA POLÍTICA DE CONTROLE

Nas seções anteriores foram apresentados testes para verificar qual taxa de evaporação,  $\rho$ , e configuração de partições,  $N_1 \times N_2$ , deveriam ser utilizados para o treinamento da política de controle. A escolha desses parâmetros baseou-se na análise de qual combinação de  $\rho$  e  $N_1 \times N_2$  faz com que o sistema alcance o estado desejado, dentro da margem de erro admitida, em um menor tempo possível. Ao final dos testes observou-se que os melhores parâmetros, para os casos considerados, foram  $\rho = 0,01$  e o número de partições de  $61 \times 61$ .

Na sequência, é verificado o melhor universo de ações,  $\mathcal{U}$ , considerando-se o mesmo critério adotado para encontrar  $\rho$  e o número de partições. Os conjuntos considerados foram

$$\mathcal{U} = \{(\hat{p}, \hat{q}, \hat{r}) : \hat{p}, \hat{q} \in \{-u_{max}, 0, +u_{max}\}, \hat{r} = 0\}, \quad (45)$$

com  $u_{max}$  assumindo os valores 0,5; 1; 2; 3 e 5.

Novamente, treinam-se cinco políticas para cada opção de  $\mathcal{U}$  e verifica-se, por meio de simulações, o tempo necessário para alcançar o mesmo nível de estabilidade com os estados iniciais definidos anteriormente, (43) e (44). As políticas de controle em que  $u_{max} = 0, 5$  e  $5$  não obtiveram êxito em controlar o sistema no tempo proposto e, portanto, foram desconsideradas. A Tabela 11 mostra os tempos médios, desvio padrão e tempo de treinamento de cada política com  $u_{max}$  assumindo os valores 1, 2 e 3.

**Tabela 11:** Tempo computacional de treinamento, tempo médio de acomodação e desvio padrão obtidos com diferentes políticas de controle, analisando distintos universos de ações.

$u_{max}$	tempo computacional de treinamento (s)	tempo médio (s)	desvio padrão
1	1268	1,2563	0,5953
1	1232	1,2647	0,5854
1	1230	1,2557	0,5888
1	1242	1,2491	0,5874
1	1232	1,2554	0,5895
2	1026	0,9533	0,4209
2	980	1,0798	0,4792
2	928	1,0209	0,4100
2	975	1,0696	0,4267
2	988	1,0228	0,4612
3	709	1,6533	0,3305
3	722	1,6689	0,2972
3	712	1,6731	0,3193
3	671	1,6556	0,3135
3	663	1,6648	0,3159

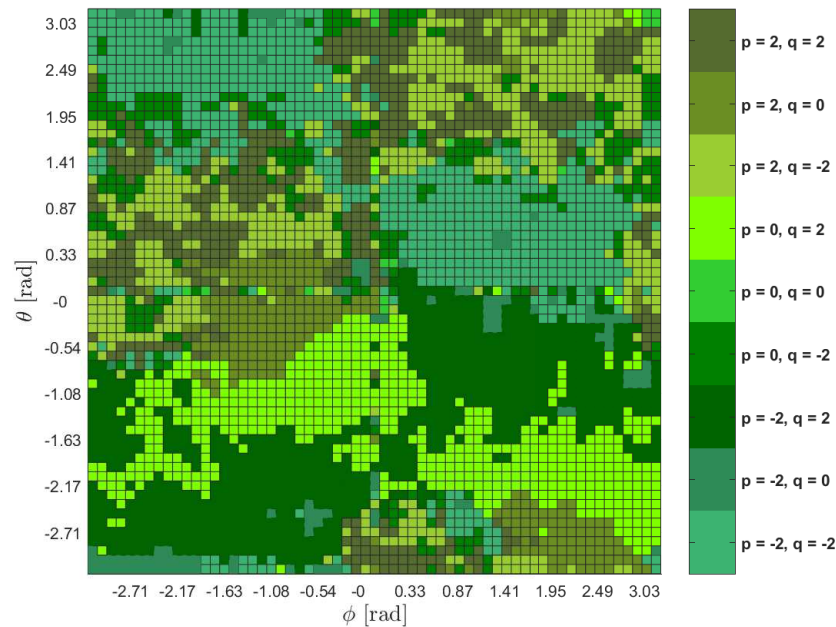
Na Tabela 12, analisando os tempos obtidos para as variações de  $u_{max}$ , fica claro que para  $u_{max} = 2$ , tem-se um melhor resultado, já que uma política de controle treinada com este valor faz com que os estados iniciais atinjam o estado objetivo, dentro da margem de erro admitida, em um tempo médio menor que os demais, como pode ser verificado comparando as média destes valores. Na Tabela 12 é mostrada também a média do tempo de treinamento para cada  $u_{max}$ .

**Tabela 12:** Média do tempo computacional de treinamento e média do tempo médio de acomodação obtidas para cada distinto universo de ações.

$u_{max}$	média do tempo de treinamento (s)	média do tempo médio (s)
1	1241	1,2562
2	979	1,0325
3	696	1,6631

#### 5.4.4 SIMULAÇÃO DA REPOSTA DO QUADRICÓPTERO CONTROLADO

Nesta seção, aborda-se a simulação do controle do sistema tipo quadricóptero realizada no SIMULINK. A política de controle empregada neste problema é treinada com os parâmetros que obtiveram um melhor resultado, mostrados anteriormente. Deste modo, o valor de  $\rho$  utilizado é 0,01, com  $61 \times 61$  partições e  $\mathcal{U}$  composto por todas as combinações de  $(\hat{p}, \hat{q}, \hat{r})$  em que  $\hat{p}$  e  $\hat{q}$  podem assumir os valores  $-2, 0$  e  $2$ , com  $\hat{r}$  sempre igual a  $0$ . A política de controle gerada com estes parâmetros pode ser visualizada na Figura 24.



**Figura 24:** Política de Controle do quadricóptero utilizando  $\rho = 0,01$ ,  $61 \times 61$  partições e  $u_{max} = 2$ .

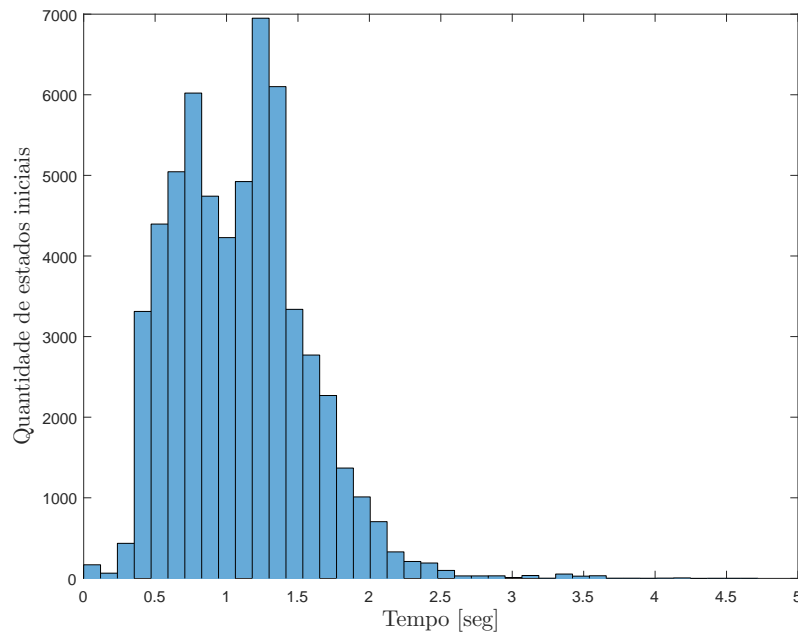
Para verificar a eficácia da política de controle obtida, o sistema foi controlado a partir de 59319 distintos estados iniciais, sendo estes definidos por (46), (47) e (48). Deste modo, pode-se analisar o tempo que este demora pra entrar na faixa de  $\pm 0,4$  radianos em torno do objetivo. Durante as simulações do controle do sistema, verificou-se que uma parcela, mais especificamente 0,62%, dos estados iniciais não alcançaram o objetivo proposto. O tempo despendido para que o sistema alcançasse a faixa considerada, a partir de cada estado inicial, pode ser observado no histograma da Figura 25.

$$\boldsymbol{\eta}_0 = \begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \end{bmatrix}, \quad \boldsymbol{\nu}_0 = \begin{bmatrix} p_0 \\ q_0 \\ r_0 \end{bmatrix}, \quad (46)$$

tal que

$$\phi_0, \theta_0, \psi_0 \in \{0, 45k : k \in \mathbb{Z}, -7 < k < 7\}, \quad (47)$$

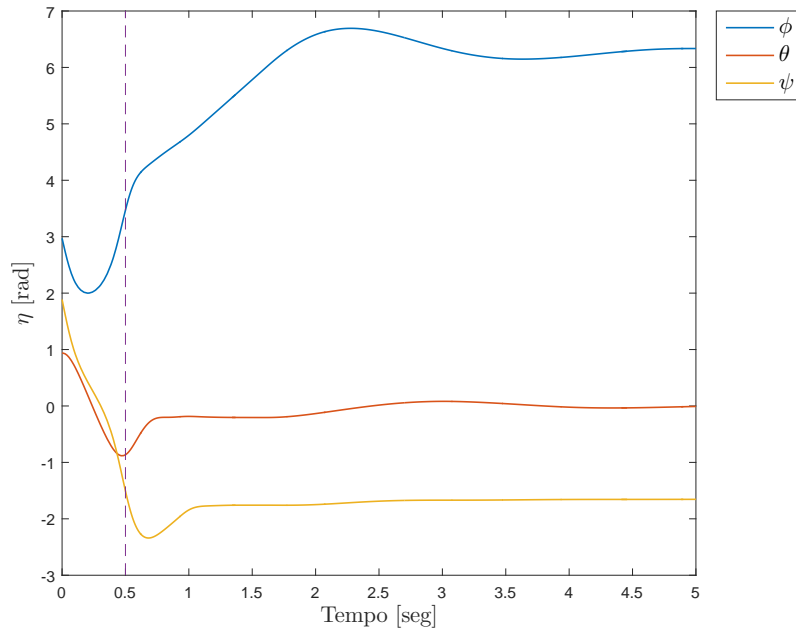
$$p_0, q_0, r_0 \in \{5k : k \in \mathbb{Z}, -2 < k < 2\}, \quad (48)$$



**Figura 25:** Histograma do tempo de acomodação do quadricóptero.

**Fonte:** Autoria própria

Nas Figuras 26, 27 e 28, pode-se visualizar um exemplo de resposta controlada pela política encontrada. Neste exemplo, é considerado que os motores estão inicialmente desligados e somente após 0,5 segundos que é aplicada a ação de controle, isto é feito para facilitar a visualização do comportamento do sistema no momento que o mesmo começa a ser controlado. Os estados no começo da simulação, neste exemplo, foram gerados aleatoriamente, sendo  $\boldsymbol{\eta} = [2.9718 \ 0.9361 \ 1.8870]'$  e  $\boldsymbol{\nu} = [-0.9240 \ -1.3522 \ 6.5063]'$ . Após o período de 0,5 s, tem-se os estados iniciais a serem controlados em  $\boldsymbol{\eta}_0 = [3.4705 \ -0.8648 \ -1.5072]'$  e  $\boldsymbol{\nu}_0 = [1.4604 \ 0.7413 \ 6.5063]'$ .

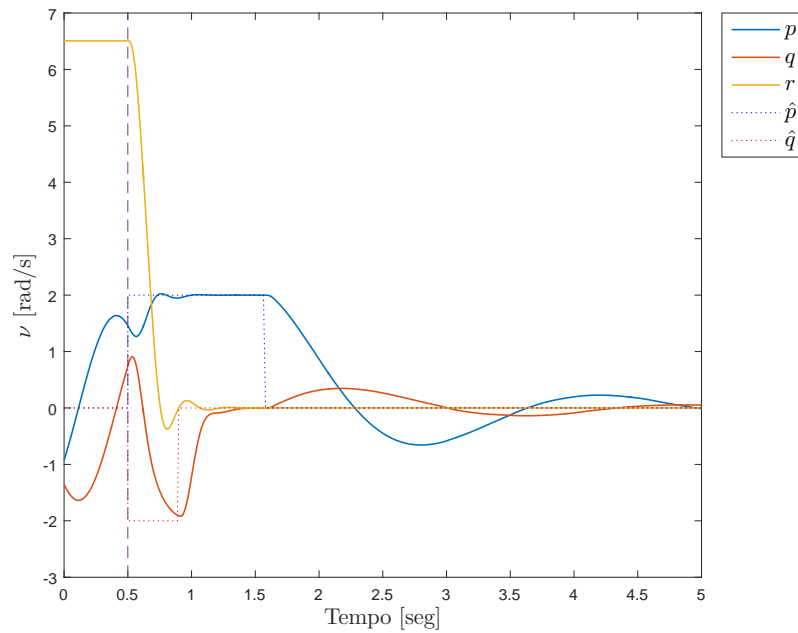


**Figura 26:** Resposta das posições angulares do sistema controlado.

**Fonte:** Autoria própria

Na Figura 26 pode-se observar o comportamento da posição angular  $\eta$ . O tracejado vertical indica o momento que as ações de controle se iniciam. Pode-se observar que os dois estados que se desejam controlar,  $\phi$  e  $\theta$ , entram numa faixa próxima ao estado desejado, definido por (36), em aproximadamente 1 segundo após o controle ser ativado e se estabilizam ao final de 5 segundos. Note que  $\phi$  se estabiliza em  $2\pi$  radianos e  $\theta$  estabiliza em 0 radianos.

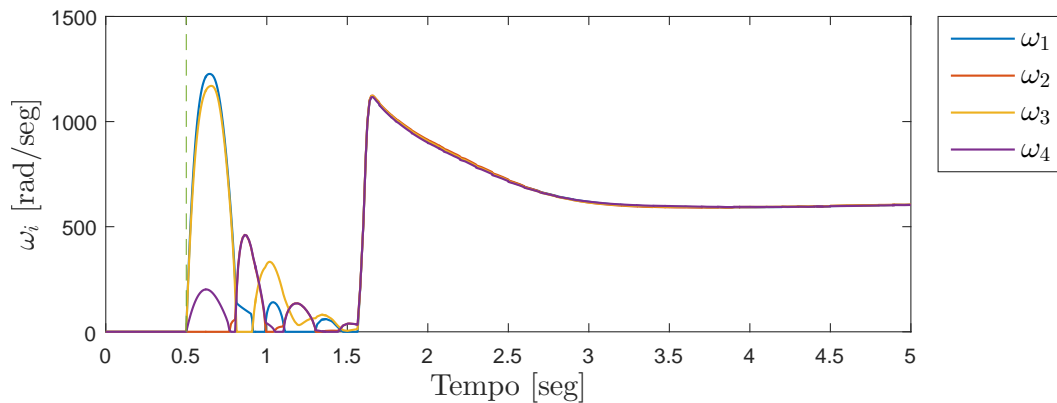
Na Figura 27 pode-se observar o comportamento da velocidade angular  $\nu$  e também de sua referência  $\hat{\nu}$ . No período de 0 a 0,5 segundos não há ação de controle, no instante 0,5 o valor de  $\hat{\nu}$  ditado pela política de controle é 2 rad/s para  $\hat{p}$  e  $-2$  rad/s para  $\hat{q}$ . Aproximadamente em 0,9 segundos, o valor de  $\hat{q}$  passa a ser 0, e o mesmo ocorre logo após a marca de 1,5 segundos com  $\hat{p}$  quando o sinal do controlador meta-heurístico vai sendo substituído pelo do controlador estacionário.



**Figura 27:** Resposta das velocidades angulares do sistema controlado.

**Fonte:** Autoria própria

Na Figura 28 são mostradas as velocidades de rotação dos quatro rotores do quadricóptero. Primeiramente, há um período de alta rotação para levar as velocidades angulares do quadricóptero para os valores de referência determinados pela política de controle. Na sequência, apenas mantêm-se as velocidades angulares do quadricóptero com pequenos ajustes. No instante 1,5s pode-se reparar a mudança na característica do sinal de controle, sendo que neste instante é feita a troca entre os dois controladores. Ao final é feito um controle com ajustes finos em que todos os rotores possuem velocidades bem próximas e assim geram pequenos torques no quadricóptero.



**Figura 28:** Velocidade angular dos motores do quadricóptero.

**Fonte:** Autoria própria



## 6 CONCLUSÃO

Neste trabalho fez-se o estudo e a aplicação da técnica ACL em dois sistemas não-lineares. Com o intuito de validar o aprendizado da técnica foi feita a replicação de uma das simulações apresentadas em (AST, 2010) que, de modo sucinto, consiste em estabilizar um pêndulo invertido partindo de qualquer estado inicial possível até sua posição ereta. Durante as simulações, com base na análise do sistema, utilizou-se uma forma de busca própria, baseada na característica espelhada do sistema. O uso desta busca apresentou como vantagem uma convergência mais rápida no treinamento da política de controle. Entretanto, uma desvantagem é que isto só pode ser feito em casos específicos, nos quais o sistema possui essa característica espelhada. Os resultados obtidos com o controle do pêndulo invertido se mostraram satisfatórios sendo próximos ao apresentado no artigo (AST, 2010).

No segundo problema objetiva-se utilizar a técnica ACL no controle da atitude de um quadricóptero. Encontrou-se dificuldades na aplicação do método devido ao sistema ser de sexta ordem, sendo a principal dificuldade o tempo computacional necessário nos treinamentos da política de controle. A alternativa proposta para lidar com essa limitação foi manipular as variáveis de estado do sistema, de modo que o ACL utilizasse um sistema apenas de segunda ordem para treinar a política de controle. Desta forma, a política de controle fica responsável somente por definir quais velocidades angulares devem ser utilizadas como referência por um controlador externo. Para validar o controle obtido utilizou-se o sistema original e foi possível realizar o controle proposto de modo satisfatório.

Foram observadas algumas limitações com o método ACL. Uma delas está na maneira com que o número total de partições do espaço de estados, e conseqüentemente o tempo necessário para se realizar o treinamento, cresce de modo exponencial com o aumento da ordem do sistema. Outra limitação é a necessidade do número de possíveis entradas ser finito. Além disso, quanto maior for o número de entradas, maior será o tempo computacional despendido. Por último, a política de controle gerada pelo método

está atrelada a um único valor fixo de referência. No pêndulo, a política de controle sempre direcionava os estados à posição ereta, e no problema do quadricóptero, a política de controle levava dois dos estados de posição angular para zero. Assim, se o objetivo consistisse em alcançar diferentes referências, seria necessário gerar uma política de controle para cada uma dessas.

Apesar das limitações citadas, o método se demonstrou promissor para a solução de problemas em que as limitações pudessem ser contornadas. Em um trabalho futuro, poderia ser estudado a possibilidade de se utilizar particionamentos dinâmicos ao longo do treinamento na técnica ACL. Pode-se observar nas políticas de controle que a quantidade de partições necessárias nos espaço de estados não são homogêneas, sendo que se formaram agrupamentos de partições em que a mesma ação de controle é escolhida. Assim, uma possível melhoria seria fazer com que o particionamento deixasse de ser equidistante, tornando-o dinâmico ao longo das iterações do treinamento da política de controle. Esta mudança tem como objetivo diminuir o tempo de treinamento e tornar mais factível a aplicação da técnica em sistemas de ordens maiores.

## REFERÊNCIAS

- ALAYA, I.; SOLNON, C.; GHÉDIRA, K. Ant algorithm for the multi-dimensional knapsack problem. In: CITESEER. **International Conference on Bioinspired Optimization Methods and their Applications (BIOMA 2004)**. [S.l.], 2004.
- ALOISE, D. et al. Heurísticas de colônia de formigas com path-relinking para o problema de otimização da alocação de sondas de produção terrestre-spt. **Anais do XXXIV SBPO–Simpósio Brasileiro de Pesquisa Operacional. Rio de Janeiro, 2002.**
- ARAÚJO, A. F.; VITTORI, K.; FERNANDES, R. S. Roteamento alternativo em redes de telecomunicações baseado no comportamento de formigas. In: **Congresso Brasileiro de Automática CBA, 15**. [S.l.: s.n.], 2004.
- AST, J. M. V. Ant colony optimization for control. 2010.
- AST, J. V.; BABUSKA, R.; SCHUTTER, B. D. Fuzzy ant colony optimization for optimal control. In: IEEE. **American Control Conference, 2009. ACC'09**. [S.l.], 2009. p. 1003–1008.
- AST, J. van; BABUSKA, R.; SCHUTTER, B. D. Ant colony optimization for optimal control. In: IEEE. **2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)**. [S.l.], 2008. p. 2040–2046.
- CARVALHO, M. B. de et al. Aplicações de meta-heurística genética e fuzzy no sistema de colônia de formigas para o problema do caixeiro viajante. Campinas, SP, 2007.
- ÇATAY, B. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. **Expert Systems with Applications**, Elsevier, v. 37, n. 10, p. 6809–6817, 2010.
- CHIHA, I.; LIOUANE, N.; BORNE, P. Tuning pid controller using multiobjective ant colony optimization. **Applied Computational Intelligence and Soft Computing**, Hindawi Publishing Corp., v. 2012, p. 11, 2012.
- DENG, G.-F.; LIN, W.-T. Ant colony optimization for markowitz mean-variance portfolio model. In: SPRINGER. **International Conference on Swarm, Evolutionary, and Memetic Computing**. [S.l.], 2010. p. 238–245.
- DORIGO, M.; BIMTTARI, M. Thomas, "Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique. [S.l.], 2006.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. The ant system: An autocatalytic optimizing process. Technical report, 1991.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 26, n. 1, p. 29–41, 1996.

- DORIGO, M.; STÜTZLE, T. Ant colony optimization: overview and recent advances. **Techreport, IRIDIA, Université Libre de Bruxelles**, 2009.
- DORMAND, J. R.; PRINCE, P. J. A family of embedded runge-kutta formulae. **Journal of computational and applied mathematics**, Elsevier, v. 6, n. 1, p. 19–26, 1980.
- GLOVER, F. W.; KOCHENBERGER, G. A. **Handbook of metaheuristics**. [S.l.]: Springer Science & Business Media, 2006.
- GOSS, S. et al. Self-organized shortcuts in the argentine ant. **Naturwissenschaften**, Springer, v. 76, n. 12, p. 579–581, 1989.
- HASSANZADEH, I.; MOBAYEN, S. Controller design for rotary inverted pendulum system using evolutionary algorithms. **Mathematical Problems in Engineering**, Hindawi Publishing Corporation, v. 2011, 2011.
- HSIAO, Y.-T.; CHUANG, C.-L.; CHIEN, C.-C. Ant colony optimization for designing of pid controllers. In: IEEE. **Computer Aided Control Systems Design, 2004 IEEE International Symposium on**. [S.l.], 2004. p. 321–326.
- KHALIL, H. K. **Nonlinear Systems**. [S.l.]: Prentice-Hall, New Jersey, 1996.
- LUUKKONEN, T. Modelling and control of quadcopter. 2011.
- MANIEZZO, V.; COLORNI, A. The ant system applied to the quadratic assignment problem. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 11, n. 5, p. 769–778, 1999.
- MULATI, M. H.; CONSTANTINO, A. A.; SILVA, A. F. da. Otimização por colônia de formigas. **LOPES, HS; RODRIGUES, LCDA; STEINER, MTA Meta-Heurísticas em Pesquisa Operacional. Ominipax**, p. 53–68, 2013.
- OGATA, K.; MAYA, P. Á.; LEONARDI, F. **Engenharia de controle moderno**. [S.l.]: Prentice Hall, 2003.
- RAMALHINHO-LOURENÇO, H.; SERRA, D. Adaptive approach heuristics for the generalized assignment problem. 1998.
- SERAPIÃO, A. B. d. S. Fundamentos de otimização por inteligência de enxames: uma visão geral. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, SciELO Brasil, v. 20, n. 3, p. 271–304, 2009.
- SHAMPINE, L. F.; REICHEL, M. W. The matlab ode suite. **SIAM journal on scientific computing**, SIAM, v. 18, n. 1, p. 1–22, 1997.
- SOLNON, C.; FENET, S. A study of aco capabilities for solving the maximum clique problem. **Journal of Heuristics**, Springer, v. 12, n. 3, p. 155–180, 2006.
- TALBI, E.-G. **Metaheuristics: from design to implementation**. [S.l.]: John Wiley & Sons, 2009.
- TANSCHKEIT, R. Sistemas fuzzy. **Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro**, 2004.