

SIMULAÇÃO E CONTROLO DE UM MANIPULADOR ROBÓTICO 2R USANDO ALGORITMOS GENÉTICOS

NUNO FILIPE AZEVEDO MARTINS

Setembro de 2021

SIMULAÇÃO E CONTROLO DE UM MANIPULADOR ROBÓTICO 2R USANDO ALGORITMOS GENÉTICOS

Nuno Filipe Azevedo Martins



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2021

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de
Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de
Computadores

Candidato: Nuno Filipe Azevedo Martins, N° 1130578, 1130578@isep.ipp.pt

Orientação científica: Ramiro de Sousa Barbosa, rsb@isep.ipp.pt



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

6 de setembro de 2021

Agradecimentos

Antes demais, cabe-me deixar uma menção honrosa e um pedido de agradecimento ao orientador Ramiro de Sousa Barbosa pelo apoio e disponibilidade ao longo do percurso de toda a etapa.

Estendo o sentimento aos demais docentes do mestrado de Engenharia Electrotécnica e de Computadores por todo o conhecimento que partilharam. Todos aqueles que, de qualquer forma, puderam estar presentes no decorrer deste capítulo que agora encerro, o meu mais sincero obrigado.

Por fim, mas não de todo menos importante, deixo aqui uma mensagem de profundo agradecimento ao herói e mentor da minha vida. Costuma-se dizer que os nossos ídolos nunca morrem e acabei por aprender da pior maneira possível.

Podes ter partido demasiado cedo, mas certificar-me-ei que a tua mensagem, o teu ser e a tua essência ecoarão pela eternidade e gerações vindouras, pois afinal de contas, é tudo isso que deixamos cá após a nossa passagem.

Resumo

Muitos problemas do mundo real resolvidos nos dias de hoje envolvem a tomada de decisões e a realização de tarefas deveras complexas. Se a grande generalidade das decisões são tomadas pelo ser humano, em sentido oposto, as tarefas acabam por ser realizadas por sistemas robóticos que permitem trazer uma maior eficácia, segurança e até mesmo desempenho em comparação com o Homem.

Na última década, foram feitas inúmeras pesquisas a respeito do controlo de trajetórias dos manipuladores robóticos. Paralelamente foram aplicados algoritmos genéticos em diversos campos na robótica de modo a ser possível obter uma optimização em várias áreas como controlo robótico, processamento de imagens, reconhecimento de padrões ou até mesmo reconhecimento da fala.

Assim, é estudado a abordagem do planeamento de duas trajetórias ideais, com um braço robótico, recorrendo aos algoritmos genéticos.

Como caso de estudo, os principais objetivos passam pela trajetória mais próxima possível da trajetória ideal, implicando o menor erro possível para realizar o movimento pretendido pelo manipulador robótico.

No que toca à cinemática do manipulador, o braço apresenta dois graus de liberdade, sendo que se recorre à técnica Newton-Euler para calcular as suas equações dinâmicas e utiliza-se o programa MatLab, no qual se encontra embebido a plataforma Simulink para verificar a simulação do sistema criado.

Palavras-Chave

Robótica, Manipulador, Braço, Algoritmo Genético, Optimização, MatLab, Simulink.

Abstract

Many real-world problems solved these days involve decision-making and performing complex tasks. If the great majority of decisions are made by the human being the tasks it selves end up being carried out by robotic systems that allow to bring greater efficiency, safety and even performance in comparison with the human kind.

In the last decade, a lot of research has been done on the trajectory control of robotic manipulators. At the same time, genetic algorithms were applied in several fields in robotics to be able to obtain optimization in various areas such as robotic control, image processing, pattern recognition or even speech recognition.

Thus, the approach to planning two ideal trajectories with a robotic arm is studied, resorting to genetic algorithms.

As a case study, the main goal of this study is to get the trajectory as close as possible to the ideal one to get the minimum error possible to perform the movement intended by the robotic manipulator.

Regarding the manipulator's kinematics, the arm has two degrees of freedom and using the Newton-Euler technique, it is possible to calculate its dynamics equations and to verify and test the simulation of the created system using MatLab, more concretely the Simulink platform.

Keywords

Robotics, Manipulator, Arm, Genetic Algorithm, Optimization, MatLab, Simulink.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS.....	XI
ACRÓNIMOS	XIII
1. INTRODUÇÃO.....	1
1.1. INTRODUÇÃO	1
1.2. OBJETIVOS	2
1.3. CALENDARIZAÇÃO	2
1.4. ORGANIZAÇÃO DO RELATÓRIO.....	3
2. ALGORITMOS GENÉTICOS.....	5
2.1. APLICAÇÃO DE UM ALGORITMO GENÉTICO.....	7
2.2. O ALGORITMO GENÉTICO APLICADO A DIVERSAS ÁREAS.....	11
2.3. TIPOS DE ALGORITMOS GENÉTICOS	13
3. A EVOLUÇÃO ROBÓTICA.....	25
3.1. ANÁLISE INTRODUTÓRIA E ORIGEM.....	25
3.2. ROBÔS INDUSTRIAIS	30
4. O ROBÔ RR.....	33
4.1. MODELAÇÃO DO ROBÔ	34
4.2. MODELO CINEMÁTICO	35
4.3. MODELO DINÂMICO	38
4.4. CONTROLADOR PID	41
5. IMPLEMENTAÇÃO DO SISTEMA.....	45
5.1. TRAJETÓRIAS PERCORRIDAS	46
5.2. CONTROLADOR	47
5.3. MODELO ROBÓTICO.....	48
5.4. PARÂMETROS DE INICIALIZAÇÃO	49

5.5.	ÍNDICES DE DESEMPENHO.....	50
6.	RESULTADOS.....	53
6.1.	GENETIC ALGORITHM (GA).....	53
6.2.	PARTICLE SWARM OPTIMIZATION (PSO).....	62
6.3.	GLOBAL PATTERN SEARCH (GPS).....	69
6.4.	ANÁLISE COMPARATIVA.....	77
7.	CONCLUSÕES	79
7.1.	TRABALHO FUTURO.....	80
	REFERÊNCIAS DOCUMENTAIS.....	81
	ANEXO A. DIAGRAMA SIMULINK.....	85
	ANEXO B. CÓDIGO MATLAB	87

Índice de Figuras

Figura 1	Princípio básico de aplicação de um AG	8
Figura 2	Exemplo do mecanismo “ <i>Roulette Wheel</i> ”	10
Figura 3	Trilho inicial da colônia ao alimento	14
Figura 4	Distância entre trilhos iniciais.....	14
Figura 5	Retorno à colônia com libertação do feromona.....	15
Figura 6	Início do segundo percurso através do trilho mais eficaz.....	15
Figura 7	Afluência de ambos os trilhos no segundo percurso	16
Figura 8	Otimização do trilho	16
Figura 9	Algoritmo ACO.....	17
Figura 10	Algoritmo ABC	19
Figura 11	Processo de procura do melhor ninho	20
Figura 12	Algoritmo CS	21
Figura 13	Algoritmo PSO.....	22
Figura 14	Área de estudo do algoritmo.....	24
Figura 15	Linha temporal dos robôs (1929-1977) [18]	26
Figura 16	Linha temporal dos robôs (1978-1989) [18]	27
Figura 17	Linha temporal dos robôs (1990-1999) [18]	28
Figura 18	Linha temporal dos robôs (2000-2009) [18]	28
Figura 19	Linha temporal dos robôs (2010-2015) [18]	29
Figura 20	Linha temporal dos robôs industriais (1960-2003) [18].....	30
Figura 21	Gama atual de robôs industriais [22]	31
Figura 22	Robô RR e braço humano.....	34
Figura 23	Relação de espaços entre cinemática direta e inversa	35
Figura 24	Manipulador RR (2 graus de liberdade).....	36
Figura 25	Ângulos auxiliares do manipulador RR.....	37
Figura 26	Algoritmo iterativo	41
Figura 27	Algoritmo não iterativo	42
Figura 28	Algoritmo paralelo.....	42
Figura 29	Comparação de ambos controladores PID usando o método do PID Tuner.....	44
Figura 30	Aplicação do AG no Modelo Robótico.....	44

Figura 31	Diagrama de blocos do sistema.....	45
Figura 32	Implementação do sistema em Simulink.....	46
Figura 33	Trajetória circular (esquerda) e linear (direita) estudadas	46
Figura 34	Trajetórias estudadas no plano XY: círculo e recta.....	47
Figura 35	Subsistema do Controlador.....	47
Figura 36	Modelo Robótico	48
Figura 37	Trajetória circular descrita e esperada: a) GA – IAE; b) GA – ITAE; c) GA – ISE.....	55
Figura 38	Comportamento circular descrito e esperado de cada junta: a) GA – IAE; b) GA – ITAE; c) GA- ISE	56
Figura 39	Funções objetivo: a) IAE - GA; b) ITAE - GA; c) ISE - GA.....	57
Figura 40	Trajetória linear descrita e esperada: a) GA – IAE; b) GA – ITAE; c) GA – ISE.....	59
Figura 41	Comportamento linear descrito e esperado de cada junta: a) GA – IAE; b) GA – ITAE; c) GA- ISE.....	60
Figura 42	Funções objetivo: a) IAE - GA; b) ITAE - GA; c) ISE - GA.....	61
Figura 43	Trajetória circular descrita e esperada: a) PSO – IAE; b) PSO – ITAE; c) PSO – ISE	63
Figura 44	Comportamento descrito e esperado de cada junta: a) PSO – IAE; b) PSO – ITAE; c) PSO - ISE	64
Figura 45	Funções objetivo: a) IAE - PSO; b) ITAE - PSO; c) ISE - PSO	65
Figura 46	Trajetória linear descrita e esperada: a) PSO – IAE; b) PSO – ITAE; c) PSO – ISE ...	67
Figura 47	Comportamento descrito e esperado de cada junta: a) PSO – IAE; b) PSO – ITAE; c) PSO – ISE	68
Figura 48	Funções objetivo: a) IAE - PSO; b) ITAE - PSO; c) ISE - PSO	69
Figura 49	Trajetória circular descrita e esperada: a) GPS – IAE; b) GPS – ITAE; c) GPS – ISE	71
Figura 50	Comportamento descrito e esperado de cada junta: a) GPS – IAE; b) GPS – ITAE; c) GPS – ISE	72
Figura 51	Funções objetivo: a) IAE - GPS; b) ITAE - GPS; c) ISE - GPS	73
Figura 52	Trajetória linear descrita e esperada: a) GPS – IAE; b) GPS – ITAE; c) GPS – ISE ...	75
Figura 53	Comportamento descrito e esperado de cada junta: a) GPS – IAE; b) GPS – ITAE; c) GPS - ISE	76
Figura 54	Funções objetivo: a) IAE - GPS; b) ITAE - GPS; c) ISE - GPS	77

Índice de Tabelas

Tabela 1	Calendarização	2
Tabela 2	Parâmetros de configuração do robô.....	49
Tabela 3	Parâmetros de inicialização	49
Tabela 4	Ganhos do Controlador GA - Círculo	54
Tabela 5	Desempenho da Malha GA - Círculo.....	54
Tabela 6	Ganhos do Controlador GA - Recta.....	58
Tabela 7	Desempenho da Malha GA – Recta.....	58
Tabela 8	Ganhos do Controlador PSO - Círculo	62
Tabela 9	Desempenho da Malha PSO – Círculo	62
Tabela 10	Ganhos do Controlador PSO - Recta	66
Tabela 11	Desempenho da Malha PSO – Recta	66
Tabela 12	Ganhos do Controlador GPS - Círculo	70
Tabela 13	Desempenho da Malha GPS – Círculo	70
Tabela 14	Ganhos do Controlador GPS - Recta	74
Tabela 15	Desempenho da Malha GPS – Recta	74
Tabela 16	Erro do sistema.....	78

Acrónimos

ABC	–	<i>Artificial Bee Colony</i>
ACO	–	<i>Ant Colony Optimization</i>
AG	–	<i>Algoritmos Genéticos</i>
AI	–	<i>Artificial Intelligence</i>
CS	–	<i>Cuckoo Search</i>
GDL	–	<i>Graus de liberdade</i>
GPS	–	<i>Global Pattern Search</i>
IAE	–	<i>Integral Absolute Error</i>
ISE	–	<i>Integral Square Error</i>
ITAE	–	<i>Integral Time Absolute Error</i>
MatLab	–	<i>Matrix Laboratory</i>
ML	–	<i>Machine Learning</i>
PID	–	<i>Proporcional Integral Derivativo</i>
PV	–	<i>Process Variable</i>
SP	–	<i>Set Point</i>
TEDI	–	<i>Tese/Dissertação</i>
WCCI	–	<i>World Congress on Computational Intelligence</i>

1. INTRODUÇÃO

A presente dissertação foi elaborada no âmbito da unidade curricular Tese/Dissertação (TEDI), do segundo semestre presente no segundo ano do curso de Mestrado em Engenharia Electrotécnica e de Computadores - Automações e Sistemas.

1.1. INTRODUÇÃO

O tema “Simulação e Controlo de um Manipulador Robótico 2R Usando Algoritmos Genéticos”, advém do estudo do papel em que os algoritmos genéticos podem desempenhar nas mais diversas e complexas tarefas dentro da robótica.

Qualquer sistema robótico pode acarretar um ambiente perigoso de funcionamento, no qual é essencial garantir que o robô se move do ponto inicial ao ponto final sem qualquer colisão com obstáculos ou com o ambiente circundante. Igualmente importante, são o tempo de processamento, de comunicação e o consumo de energia, que podem ser reduzidos quanto maior for a eficácia do movimento descrito.

Como tal, foi usado como tema de estudo um braço robótico com dois graus de liberdade, no qual é verificado o controlo do mesmo e em que medida se pode aperfeiçoar, tendo como base os algoritmos genéticos estudados e aplicados.

1.2. OBJETIVOS

O objetivo principal desta dissertação, prende-se ao facto de encontrar e projetar a melhor trajetória possível para que o braço do robô se movimente.

Para desenvolver e implementar uma abordagem eficaz, recorre-se à utilização de algoritmos genéticos, que irão otimizar os valores de controlo que alimentam o braço do robô, de modo que este seja capaz de determinar e prever a melhor solução, e percorrer eficazmente a trajetória que é proposta.

Adicionalmente foi implementada e desenvolvida uma topologia de dois graus de liberdade, na qual a modelação do braço robótico cobriu áreas como a cinemática e dinâmica.

1.3. CALENDARIZAÇÃO

Na Tabela 1 é possível ver uma calendarização da realização do projeto. As tarefas foram divididas distintivamente, usando a escala de duração semanal.

Tabela 1 Calendarização

ID	Tarefas	Data Inicial	Data Final	Duração	Março				Abril				Maio				Junho				Julho			
					S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
1	Análise de Algoritmos Genéticos	04/03	17/03	3S																				
2	Exemplos de Algoritmos Genéticos	12/03	29/03	3S																				
3	Pesquisa sobre Manipuladores Robóticos	02/04	18/04	3S																				
4	Cálculo do Modelo Dinâmico do Robô	24/04	14/05	3S																				
5	Cálculo do Modelo Cinemático do Robô	24/04	14/05	3S																				
6	Estudo sobre o controlador PID	03/06	12/06	2S																				
7	Implementação do sistema em Simulink	17/06	16/07	4S																				
8	Teste do sistema global	22/07	30/07	2S																				
9	Elaboração do Relatório	04/05	31/07	10S																				

É possível também ver que ocorre uma paragem, a meio do mês de Maio que representa o período de pausa.

1.4. ORGANIZAÇÃO DO RELATÓRIO

Esta secção destina-se à explicação do propósito e necessidade de cada um dos capítulos presentes na dissertação.

O relatório inicia-se por uma breve introdução aos algoritmos genéticos, o seu significado em concreto e nas suas diferentes características (Capítulo 2). Após isso é dado a conhecer o ciclo de cada algoritmo em maior detalhe e como são aplicados no nosso dia-a-dia através de vários exemplos em diferentes áreas. São também dados a conhecer três exemplos correntes de algoritmos, de como funcionam na natureza e qual o seu pseudocódigo.

No terceiro capítulo, discute-se a história dos mecanismos robóticos, baseando-se numa linha temporal evolutiva desde os seus primórdios. É então descrito em maior detalhe o manipulador robótico em particular estudado nesta tese, calculando-se os seus modelos dinâmicos e cinemáticos (Capítulo 4). De seguida é dada uma breve revisão do que consiste um controlador PID de acordo com a sua função, utilidade e o porquê de ser implementado neste sistema.

A seguir, é discutido o sistema geral criado, olhando para o projeto trabalhado na ferramenta Simulink, que se encontra inerente à plataforma MATLAB (Capítulo 5). Esta implementação foi regida por uma metodologia previamente delineada: análise, estudo, planeamento, execução, e por fim, teste. Neste capítulo é também discutido os diversos métodos e detalhes do fluxo da simulação, assim como o fator que os algoritmos genéticos desempenham no decorrer da trajetória descrita pelo manipulador robótico. Não menos importante, são discutidos os resultados de todos os dados recolhidos da simulação e identifica-se maneiras de definir os erros que acabaram por surgir nos resultados da simulação (Capítulo 6).

Por fim, o Capítulo 7, sendo que neste último capítulo há um espaço descrito para as devidas conclusões e sobre futuro trabalho que poderia ser realizado com a aproveitação deste projeto.

2. ALGORITMOS GENÉTICOS

Algoritmos Genéticos (AG) são métodos adaptativos que podem ser utilizados para resolver problemas de pesquisa e otimização baseando-se nos processos genéticos de organismos biológicos.

Ao longo de várias gerações, populações naturais evoluíram de acordo com os princípios da seleção natural e da sobrevivência do mais apto, tal como declarado primeiramente por Charles Darwin. Ao imitar esse processo, os algoritmos genéticos são igualmente capazes de otimizar soluções para problemas do mundo real, se tiverem sido adequadamente codificados.

Embora os detalhes da evolução biológica sejam ainda não completamente compreendidos (até aos dias de hoje), existem alguns pontos apoiados por forte evidência experimental:

- Evolução é um processo que opera mais por cromossomas que por organismos. Os primeiros são ferramentas orgânicas que codificam a estrutura de um ser vivo, isto é, a estrutura é “construída” a partir da decodificação de um conjunto de cromossomas.
- A seleção natural é o mecanismo que relaciona os cromossomas com a eficiência da entidade que representam, permitindo assim que um organismo eficiente seja

adaptado ao ambiente para se reproduzir com mais frequência do que aqueles que não o são.

- O processo evolutivo ocorre durante a fase da reprodução. Existe um grande número de mecanismos reprodutivos na natureza, sendo o mais comum denominado de mutação (que faz com que os cromossomos do filho sejam diferentes dos pais) e recombinação (que combina os cromossomos dos pais para produzir o descendente).

Os AG simulam esses processos de maneira natural em populações essenciais para a evolução. Exatamente quais processos biológicos são considerados essenciais para a evolução e quais os processos que têm pouco ou nenhum papel a desempenhar, ainda é motivo de pesquisa, mas os fundamentos são claros.

Os Algoritmos Genéticos diferem dos métodos tradicionais de busca e otimização, nomeadamente em quatro aspetos:

- Trabalham com uma codificação do conjunto de parâmetros (em vez de utilizarem os próprios parâmetros).
- Trabalham com um conjunto de pontos (população) e não com um único.
- Utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar.
- Utilizam regras de transição probabilísticas e não determinísticas.

Na natureza, os indivíduos dentro de uma população competem entre si por recursos como comida, água e abrigo.

Além disso, membros da mesma espécie costumam competir para atrair um companheiro do sexo oposto. Os indivíduos com maior taxa de sucesso a sobreviver e atrair companheiros, irão produzir um número relativamente maior de descendentes. Já indivíduos com baixo desempenho terão pouca ou nenhuma influência nas gerações seguintes.

Isso significa que os genes dos seres altamente adaptados ao meio em que vivem vão-se espalhar para um número crescente de indivíduos em cada geração sucessiva. A combinação de boas características de diferentes linhas ancestrais às vezes pode criar um ser super-apto, cuja aptidão é classificada como melhor que qualquer um dos seus descendentes. Dessa

forma, as espécies evoluem para se tornarem cada vez mais adequadas ao meio ambiente em que vivem e à população em que estão integrados.

Com base nas características acima descritas conclui-se que um algoritmo genético necessita de um conjunto de indivíduos, formando uma população, sendo que cada um destes indivíduos representa uma tentativa de solução para o problema que está a ser estudado [1].

2.1. APLICAÇÃO DE UM ALGORITMO GENÉTICO

Antes de se proceder à aplicação de um algoritmo genético, é necessário perceber os termos que integram o mesmo, tal como: função objetivo, indivíduo, população, geração, função de *fitness*, seleção, reprodução e critério de paragem.

Começando pela função objetivo, esta representa o que pretendemos otimizar. Ou seja, num problema com imensos parâmetros (entradas), cada saída irá representar uma solução, que por sua vez será denominada de função objetivo. Esta função permite calcular a aptidão de cada indivíduo dentro do problema.

Um indivíduo é expresso como a representação da solução do problema, através de um arranjo ou vetor dos quais os elementos são obtidos a partir de um alfabeto finito.

Passando à população, esta é nada mais nada menos que um conjunto de indivíduos que por sua vez irão formar uma geração. Já esta, representa o resultado da aplicação da seleção, reprodução e cálculo da função *fitness* de cada indivíduo que está inserido dentro de uma população.

A função de *fitness* é uma função de avaliação, na qual é possível medir a taxa de adaptação de cada indivíduo ao ambiente onde está integrado. Já a seleção representa um processo de seriação pelos indivíduos que se encontram mais adaptados (*fitness* superior), de modo a que possam passar à reprodução, cujo objetivo passa por ter os indivíduos mais aptos a reproduzirem-se, de modo a haver uma maior probabilidade de prolongar os seus materiais genéticos, ou parte deles, pelas gerações seguintes, através da aplicação dos operadores genéticos de *crossover* ou mutação. Finalmente, o critério de paragem é caracterizado pela obtenção de uma solução satisfatória ou pura e simplesmente pelo término do tempo computacional pré-estabelecido.

A Figura 1 exemplifica e simplifica o ciclo inteiro processual da aplicação de um algoritmo genético através de um fluxograma [2].

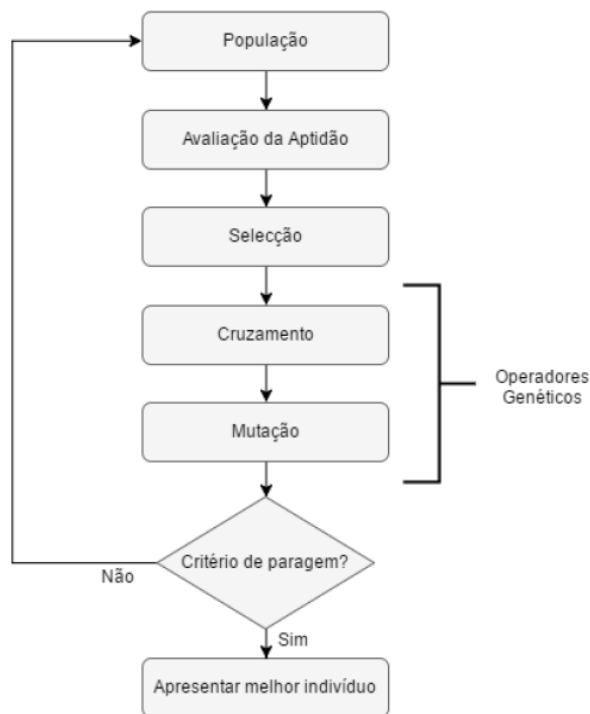


Figura 1 Princ pio b sico de aplica o de um AG

2.1.1. POPULA O

O princ pio b sico dos operadores gen ticos passa por transformar uma popula o atrav s das suas sucessivas gera es, at  finalmente chegar a um resultado satisfat rio. Assim, utiliza-se estes operadores, para que seja poss vel ver diversifica o dentro da popula o e, ao mesmo tempo, que as caracter sticas de adapta o adquiridas pelas gera es anteriores n o se percam.

Relativamente a uma nova popula o, por norma surgem duas maneiras distintas de a formar: elitismo e cruzamento.

Quanto   primeira hip tese, tal como o termo sugere, s o adicionados alguns clones dos melhores membros dispon veis na popula o. Isto permite que, durante o caso geracional, a

população não piore, uma vez que é assegurada o bom estado da genética da população a partir desses clones.

Um aspeto muito importante a ser considerado para o sucesso do processo evolutivo refere-se ao tamanho da própria população inicial e da população nas subseqüentes gerações, pois é possível comprometer a evolução dos indivíduos caso a quantidade escolhida seja muito grande ou muito pequena. Esta medida, exceto em casos particulares, deve ser estabelecida empiricamente e de acordo com a disponibilidade dos recursos computacionais a utilizar [3].

2.1.2. SELEÇÃO

Uma outra etapa de possível grande impacto no desempenho de um algoritmo genético, passa pela fase de seleção de indivíduos para a reprodução. Existem vários tipos de seleção que podem ser aplicados, entre os quais destacam-se o “*Roulette Wheel*” e a seleção por torneio.

A seleção por torneio é recomendada quando há uma grande diferença relativa no valor do *fitness* dos indivíduos. Ou seja, um par de indivíduos é escolhido aleatoriamente e o indivíduo que possuir maior *fitness* será selecionado para a reprodução.

O processo repete-se sucessivamente, obedecendo à probabilidade de seleção que foi previamente estabelecida.

Já pelo mecanismo “*Roulette Wheel*” de seleção, a cada indivíduo é associada uma porção de uma roleta, que representa diretamente o valor de *fitness*. Obrigatoriamente os indivíduos com melhor *fitness*, irão ocupar uma maior porção na Figura 2 e uma maior probabilidade de serem escolhidos para a reprodução [3].

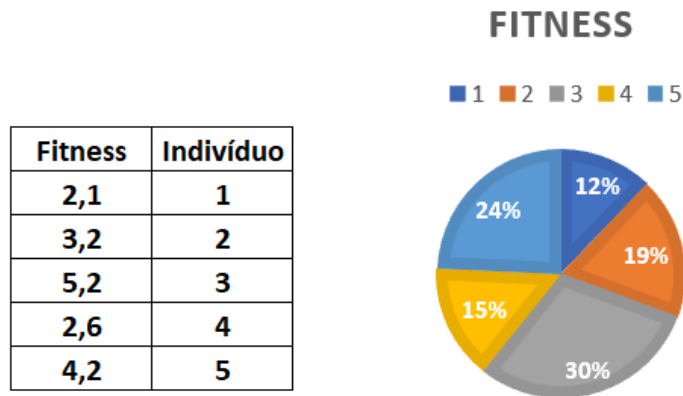


Figura 2 Exemplo do mecanismo “Roulette Wheel”

2.1.3. OPERADORES GENÉTICOS

No que toca aos operadores genéticos, a mutação pode ser considerada como um clone modificado, uma vez que apresenta praticamente as mesmas características do indivíduo pai, no entanto sofre pequenas alterações em apenas um cromossoma. Tal como o cruzamento, a mutação também tem várias maneiras de modificar o cromossoma, havendo obrigatoriamente vários operadores de mutação.

Se compararmos a mutação à codificação binária, então é possível dizer que a mutação irá influenciar no código genético apenas e só na alteração de um bit pelo seu complemento.

Já o cruzamento consiste em escolher aleatoriamente dois membros da população e recombinar ambos os genes dos pais para criar um filho. Este processo pode ter várias formas de recombinação, permitindo haver vários operadores de cruzamento [4].

2.1.4. CRITÉRIO DE PARADA

Quanto ao critério de parada, este ocorre quando numa iteração é gerado um indivíduo que acaba por ser praticamente igual à sua população. O critério de parada é a condição necessária para o AG encerrar sua execução. Essa condição deve ser verificada antes de se evoluir para uma nova geração [4].

2.2. O ALGORITMO GENÉTICO APLICADO A DIVERSAS ÁREAS

2.2.1. ANÁLISE DO MERCADO DE AÇÕES

A análise do mercado de ações é um assunto desafiador que desperta o interesse de pesquisadores e especialistas em diversas áreas.

Ao longo do tempo foram criados e desenvolvidos diferentes métodos com o simples propósito de tentar prever as flutuações dos preços das ações, de modo a maximizar os lucros dos investidores e das próprias instituições financeiras. Uma metodologia muito utilizada, baseada no acompanhamento do movimento dos preços das ações ao longo do tempo, é a Análise Técnica, na qual investidores utilizam dados estatísticos e indicadores para tomar a decisão sobre a compra ou a venda de uma determinada ação.

A partir desta metodologia e com o auxílio dos algoritmos genéticos, torna-se possível aumentar a rentabilidade das ações, expandindo a modelação da decisão de compra ou venda das ações e por meio de vários operadores lógicos (AND, OR, XOR, NAND).

Assim, neste estudo em específico, foi possível concluir que os investidores que utilizaram o operador lógico NAND, acabaram por ter uma rentabilidade final superior e um desvio padrão inferior na ordem dos 40,16%. Foi então possível não só aumentar os lucros assim como diminuir o próprio risco durante a amostragem de tempo analisada [5].

2.2.2. TELECOMUNICAÇÕES

Um dos exemplos mais conhecidos da implementação dos algoritmos genéticos na área das telecomunicações, é representado pela distribuição de redes óticas especializadas ao longo de um país em estudo (Estados Unidos da América).

Neste caso em específico, este estudo foi apresentado no WCCI'94 – *World Congress on Computational Intelligence* – onde uma simulação com diversas soluções otimizadas a partir dos algoritmos genéticos, têm como objetivo distribuir a rede otimizada ao longo de toda a área do país. Foi então projetado esse estudo que durou aproximadamente duas horas em contrapartida com os seis meses expectados pelos especialistas para realizar a mesma projeção. Para além disso, também foi possível concluir que o sistema em si era cerca de 10% mais eficaz comparativamente com os resultados propostos pelos humanos.

No cômputo geral, a simulação previa uma poupança a rondar os 100 milhões de dólares até ao fim do século 20, em comparação com a alternativa humana [6].

2.2.3. MEDICINA

Na quimioterapia, o desenvolvimento de um tumor é controlado pela administração de drogas tóxicas sujeitas a uma série de restrições. A natureza conflitante e a não linearidade das restrições, juntamente com as possibilidades combinatórias para esquemas de vários tipos de drogas, acabam por tornar difícil resolver o problema da optimização da quimioterapia do cancro por meio da experimentação clínica empírica, ou por meio de métodos de optimização tradicionais.

Os medicamentos que ajudam a combater o cancro são geralmente administrados de acordo com um programa de dosagem discreta, em que as doses são administradas em tempos restritos já predefinidos. No caso de múltiplos medicamentos, cada dose é como se de uma mistura de fármacos se tratasse, pelo que é caracterizado pelos níveis de concentração no plasma sanguíneo. A optimização do tratamento é alcançada pela modificação dessas variáveis. Portanto, o espaço de solução do problema de optimização da quimioterapia é o conjunto de vetores de controlo que representam os perfis de concentração das várias doses. Vários modelos podem ser usados para simular a resposta de um tumor à quimioterapia.

Um tratamento de quimioterapia pode ser curativo ou paliativo. Os tratamentos curativos tentam erradicar o tumor. Acredita-se que a quimioterapia por si só não pode erradicar o cancro, mas se a carga global do tumor for mantida abaixo de um certo nível, outros mecanismos (por exemplo, o sistema imunológico ou a própria morte celular programada) acabam por remover as células tumorais remanescentes [7].

Já os tratamentos paliativos são aplicados apenas quando o tumor é considerado incurável e tem o objetivo de manter uma qualidade de vida razoável para o paciente pelo maior tempo possível. Os objetivos dos tratamentos curativos e paliativos entram em conflito entre si no sentido de que os regimes das drogas que tendem a minimizar o tamanho do tumor são altamente tóxicos e, portanto, têm um efeito negativo na qualidade de vida do paciente. Além disso, foi demonstrado que um regime severo que acaba por falhar na remoção completa do tumor, pode resultar num tempo de vida do paciente mais curto do que um regime paliativo brando.

O sistema de simulação a partir de algoritmos genéticos pode ser usado como um projeto de tratamento pelos oncologistas. Assim, é possível explorar uma ampla gama de cenários de tratamento, desde os parâmetros para as taxas de crescimento do tumor, como as taxas de eficácia dos medicamentos, ou até mesmo os efeitos tóxicos e as configurações de restrição que variam com os medicamentos usados e com o tipo de cancro.

Os diferentes modelos de resposta serão apropriados para diferentes tipos de cancro e é simples substituir as várias componentes na simulação conforme necessário. Para um determinado cenário, o sistema irá sugerir estratégias de tratamento ideais em que o oncologista pode então avaliá-las antes de decidir sobre os novos regimes. Potencialmente, isso representa uma melhoria substancial a nível de custos, tanto em termos da vida humana como em termos financeiros [8].

2.3. TIPOS DE ALGORITMOS GENÉTICOS

Durante este período, foram estudados três exemplos de algoritmos genéticos que estão aplicados na vida real, mais propriamente na vida animal.

2.3.1. OPTIMIZAÇÃO DA COLÓNIA DE FORMIGAS (ACO)

A optimização da colónia de formigas – *Ant Colony Optimization* (ACO) foi introduzida pela primeira vez por Marco Dorigo nos anos 90. Este algoritmo é introduzido com base no comportamento de busca de alimento por parte da formiga, que traça um caminho entre a sua colónia e a localização do alimento.

As formigas são insetos sociais, que vivem em colónias de milhares de elementos. O comportamento das formigas é controlado pelo objetivo de procurar comida. Enquanto procuram, as formigas percorrem trilhos à volta da sua colónia. Uma formiga consegue repetidamente mover-se de um lugar para outro para encontrar a comida e, enquanto se move, deposita no trilho um composto orgânico chamado feromona. Assim, torna-se possível que os outros membros da colónia consigam percorrer o mesmo trilho, seguindo o feromona deixado pela primeira formiga. Quando uma formiga encontra algum alimento, ela tenta transportar o máximo que consegue e, ao retornar, deposita feromona nos caminhos com base na quantidade e qualidade dos alimentos. As restantes formigas vão então sentir o cheiro e irão seguir o caminho quanto mais alto for o nível de feromona deixado, assumindo um maior tráfego nesse trilho e, por consequente, o melhor e maior alimento possível [9].

Exemplificando, considera-se que existem dois caminhos para alcançar o alimento desejado, como demonstrado pela Figura 3. No início, como nenhum dos trilhos foi utilizado, não haverá qualquer quantidade de feromonas nos mesmos, portanto, a probabilidade de escolher os dois caminhos é igual, o que significa 50%. Considera-se duas formigas que escolhem os dois caminhos diferentes para alcançar o alimento, pois a probabilidade de escolher esses caminhos é equivalente à moeda ao ar.

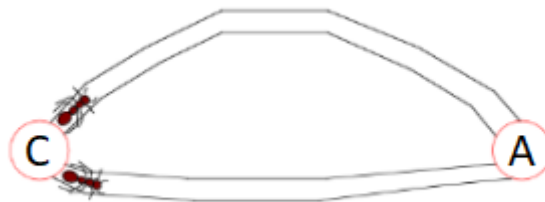


Figura 3 Trilho inicial da colônia ao alimento

É possível então ver que a distância direta da colônia ao alimento nos dois trilhos é diferente, pelo que a formiga que segue o trilho de baixo irá chegar ao destino mais rapidamente, como visto pela Figura 4.

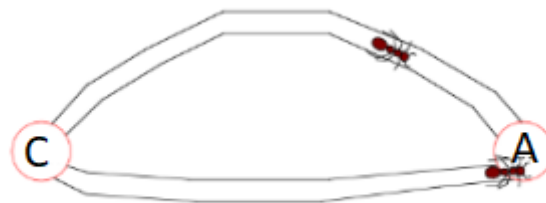


Figura 4 Distância entre trilhos iniciais

Após a formiga alcançar o alimento, irá então tentar transportar o máximo possível e retornar para a sua colônia, deixando o traço de feromona para que os restantes membros consigam seguir o mesmo trilho ao alimento. Como a formiga alcançou o alimento mais cedo, irá por consequente, também regressar à colônia mais cedo que a formiga que percorreu o trilho superior, apresentado pela Figura 5.

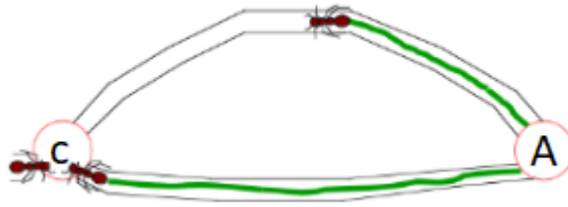


Figura 5 Regresso à colónia com libertação do feromona

Quando a formiga finalmente regressa à sua colónia, irá incentivar as restantes a seguir o seu trilho devido ao feromona deixado ao longo do caminho, como tal, a terceira formiga irá recriar o percurso inferior, até porque sendo inferior, significa que haverá uma maior distribuição de feromona em toda a sua distância, comparativamente com o trilho superior, como é possível ver pela Figura 6.

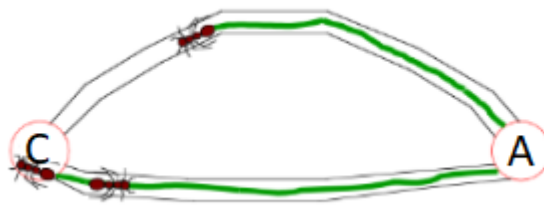


Figura 6 Início do segundo percurso através do trilho mais eficaz

Aquando da chegada da formiga que percorreu o trilho mais distante, já várias formigas seguiram o caminho inferior devido ao nível de feromona que cheiraram. No entanto uma vez que nenhuma dessas formigas ainda regressou à colónia, pode-se considerar que o nível de feromona entre ambos os trilhos são bastantes similares, pois apenas um par de formigas completou a viagem de regresso à colónia, como se vê pela Figura 7.

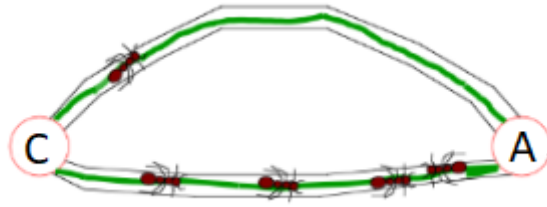


Figura 7 Afluência de ambos os trilhos no segundo percurso

É possível concluir que a larga maioria das formigas utilizam o trilho mais eficaz devido a cheirarem o feromona mais cedo que no trilho superior. Esta afluência superior irá gerar cada vez uma maior quantidade de feromona e uma maior discrepância para o trilho superior, como mostra a Figura 8.

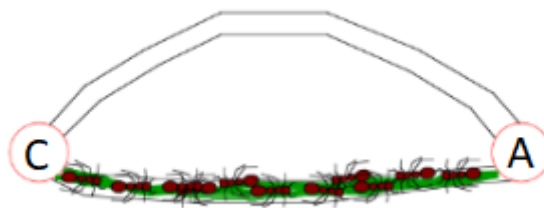


Figura 8 Otimização do trilho

Repetindo este processo algumas vezes, pode-se concluir que a maioria das formigas vão acabar por utilizar apenas o trilho inferior, otimizando o percurso da colónia até ao alimento [10].

É possível então descrever todo este processo no algoritmo, tal como representado pela Figura 9.

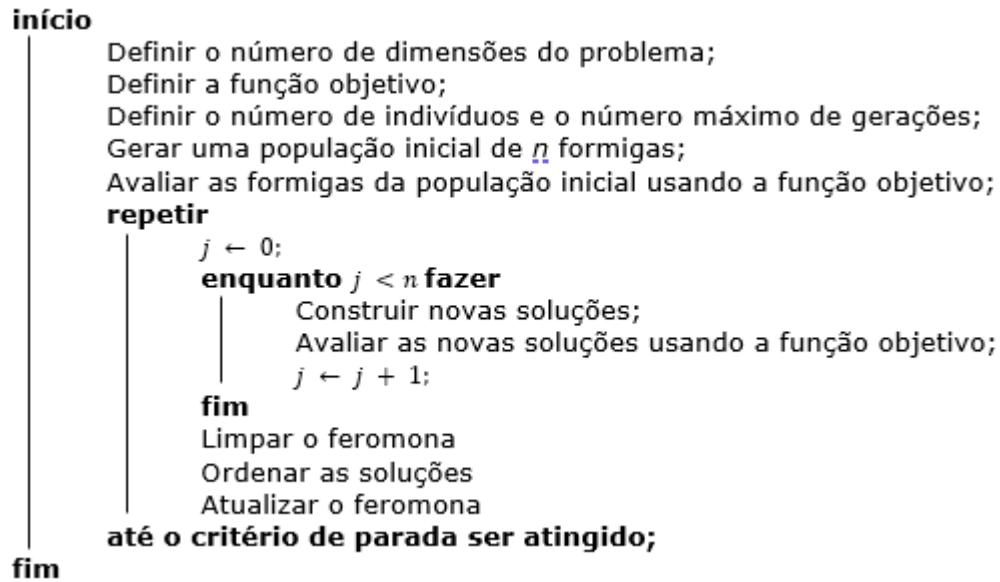


Figura 9 Algoritmo ACO

2.3.2. COLÔNIA DE ABELHAS ARTIFICIAIS (ABC)

As abelhas, tais como as formigas, representam um caso de estudo, uma vez que apresentam um comportamento coletivo inteligente, que permite realizar várias atividades em conjunto de modo que seja atingido um determinado objetivo que beneficie toda a população.

Exemplo disso, verifica-se na procura de fontes de alimento por parte das colmeias de abelhas, assim como a exploração de fontes conhecidas ou até mesmo o recrutamento de abelhas para exploração de novas fontes de alimento, quando as fontes já conhecidas acabam por se esgotar.

No que toca à procura de fontes de alimentos, os vários fatores decisivos passam pela proximidade à localização da colónia, a concentração de alimento na área estudada, a facilidade com que é possível extrair o alimento e a própria qualidade de alimento presente.

Morfologicamente numa colmeia, as abelhas podem ser descritas em três tipos: operária, rainha e zangão.

As abelhas operárias realizam as atividades externas à colônia, tal como a procura de alimento, assim como barro, água ou outros materiais para a colmeia. Também realizam trabalhos de manutenção dos ninhos caso seja necessário.

A abelha rainha tem o único propósito de reproduzir e colocar ovos que se tornarão na próxima geração de abelhas operárias, rainhas ou zangões. Este tipo de abelha apresenta uma alimentação distinta, à base de geleia real, que irá possibilitar a fertilidade para que posteriormente possa encontrar-se com o zangão e copular. Posteriormente regressa à colônia e inicia o processo de postura dos ovos. Já o macho, o zangão, após a copulação não tem tanta sorte, uma vez que na maioria dos casos acaba por morrer, ou, sem caso de sobrevivência, acaba por ser expulso da colmeia, uma vez que já não apresenta qualquer utilidade [11].

Numa colônia, as abelhas operárias também podem subdividir-se em duas funções: abelhas exploradoras, que têm como propósito a procura de novo alimento, e abelhas observadoras que estão na colônia e aguardam o regresso das abelhas exploradoras, para que possam receber informações das mesmas e tomar a decisão de qual a nova área a ser explorada. O processo de troca de informação é dado através de uma dança. O conjunto de movimentos que as abelhas exploradoras fazem quando regressam à colmeia, permite a identificação da direção, distância e qualidade do alimento na nova área que foi explorada.

Assim, consoante o maior número de abelhas com o mesmo tipo de movimento, irá dar uma perceção que a área descrita representa a melhor solução para alimento e uma taxa de rentabilidade superior, uma vez que o recrutamento é diretamente proporcional à qualidade dos vários parâmetros da fonte de alimento observada.

Passando-se ao processo de busca de alimento, quando as abelhas regressam à colmeia podem agir de três maneiras diferentes: dançar e recrutar outras abelhas operárias para intensificar a exploração de alimento na fonte, não dançar e abandonar o local, ou continuar a explorar o local escolhido sem recrutar outras abelhas.

Quando finalmente um local escolhido encontra-se esgotado de alimento, as abelhas exploradoras definem substitutos. Estes têm a tarefa de procurar uma nova fonte de alimento sem qualquer informação prévia. Quando se deparam com um novo local, regressam à colmeia e utilizam a sua capacidade de memorização para poderem situar essa fonte aos restantes membros, inicializando por completo o ciclo.

No que toca ao algoritmo da Colónia de Abelhas Artificiais (ABC – *Artificial Bee Colony*), este é utilizado para problemas de optimização multidimensional no qual cada fonte de alimento representa uma possível solução para o problema. A qualidade de uma fonte artificial equivale ao valor da função objetivo produzido pela solução.

Inicialmente o enxame é apenas composto por abelhas operárias observadoras, que aguardam a chegada das abelhas exploradoras. Neste algoritmo, cada abelha exploradora irá propor uma fonte diferente, representando uma solução única, a qual será avaliada pelas abelhas exploradoras pelos fatores já acima descritos. Após isso, de modo que o algoritmo de busca seja robusto, a fase de exploração e de intensificação por alimento deverá de estar presente. As abelhas operárias e observadoras ficarão encarregadas pela fase de intensificação enquanto que as abelhas exploradoras ficarão responsáveis pela fase de exploração.

Já a taxa de recrutamento de uma colónia real, vai permitir perceber o quão rápido o enxame localiza e explora novas fontes de alimento [12]. Este algoritmo irá demonstrar a facilidade e rapidez com que se é possível encontrar soluções para o problema proposto.

Assim sendo, o algoritmo estudado pode ser resumido e descrito pelo pseudocódigo da Figura 10.

```
início
    Definir o número de dimensões do problema;
    Definir a função objetivo;
    Definir o número de indivíduos, o valor limite para abandonar uma fonte de
    alimento e o número máximo de gerações;
    Gerar uma população inicial de  $n$  fontes de alimento;
    Avaliar as fontes de alimento da população inicial usando a função objetivo;
    repetir
        Colocar as abelhas trabalhadoras nas fontes de alimento;
        Calcular a probabilidade de escolha da fonte de alimento pelas
        abelhas observadoras;
        Colocar as abelhas observadoras nas fontes de alimento consoante os
        seus valores de probabilidade
        Interromper a exploração de fontes de alimento com o limite de
        tentativas esgotado;
        Enviar abelhas exploradoras para procurar novas fontes de alimento;
        Memorizar a melhor fonte de alimento encontrada;
    até o critério de parada ser atingido;
fim
```

Figura 10 Algoritmo ABC

2.3.3. ALGORITMO DE BUSCA DOS CUCOS (CS)

O algoritmo de busca de cucos é baseado no comportamento parasita da reprodução de algumas espécies que consiste no depósito dos ovos nos ninhos já de outros pássaros.

Daí resultam três cenários possíveis: no primeiro, os ovos acabam por ser bastante semelhantes aos do pássaro hospedeiro, pelo que não é detetada qualquer diferença e os cucos acabam por nascer e serem criados numa família “adotiva”. No segundo cenário possível, os ovos dos cucos são descobertos e, como consequência, são excluídos pelo pássaro hospedeiro, que simplesmente os deita fora. O último cenário consiste numa mudança de ninho por parte do pássaro hospedeiro, deixando o prévio ao abandono [13].

Neste algoritmo, cada solução representa um ovo no seu ninho e cada ovo de cuco representa uma nova solução. O objetivo consiste em utilizar as novas e potencialmente melhores soluções (cucos) para substituir uma não tão boa solução nos ninhos. Daí advêm três regras básicas:

1. Cada cuco deposita um ovo num ninho aleatoriamente.
2. Os melhores ninhos são mantidos para a geração seguinte.
3. Cada ninho tem apenas um ovo, tendo um número fixo de ninhos disponíveis. Aliado a isso, há uma probabilidade (p_a) de o pássaro hospedeiro descobrir o ovo do cuco no seu ninho. Esta probabilidade irá implicar que uma percentagem dos ninhos acabará por ser abandonada e substituída por novos ninhos, que irão representar uma nova solução aleatória.



Figura 11 Processo de procura do melhor ninho

O voo de Lévy é dado a um padrão que caracteriza a construção da trajetória, quer curta ou longa, que o cuco necessita de fazer para encontrar um ninho ideal para depositar o seu ovo, Figura 11.

Neste algoritmo, cada nova solução será testada e equiparada à solução atual. Se o valor do *fitness* for menor que o *fitness* da solução atual, então a nova solução será aceite. É possível então concluir que, com cada iteração, as piores soluções vão sendo progressivamente descartadas e substituídas por novas soluções [14], tal como explícito no algoritmo demonstrado pela Figura 12.

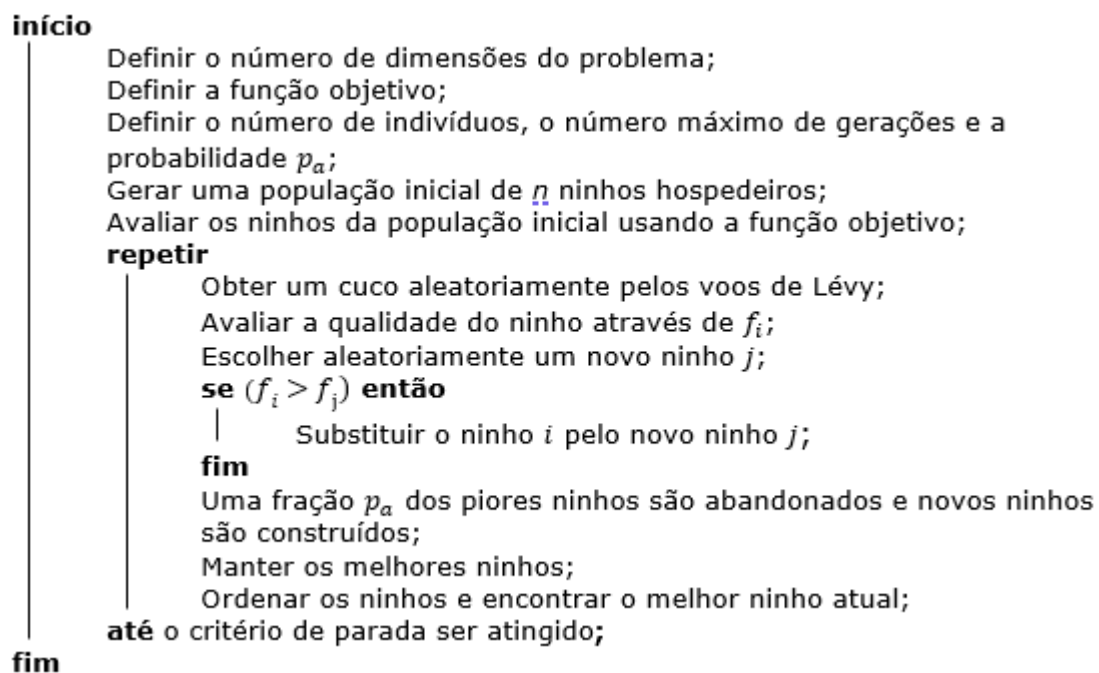


Figura 12 Algoritmo CS

2.3.4. OPTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO)

O algoritmo de otimização por enxame de partículas (*Particle Swarm Optimization* – PSO) foi inspirado no comportamento dinâmico de alguns animais que vivem em grupos, tais como bandos de pássaros ou cardumes de peixes, em busca de alimento. Este algoritmo foi proposto por Kennedy e Eberhart em 1995 e, hoje em dia, é considerado um dos principais pilares da área da Inteligência de Enxames, consistindo na otimização de uma função objetivo por meio da troca de informações entre indivíduos (partículas) de uma população

(enxame). O comportamento do grupo é influenciado pela experiência individual de cada membro que o representa e também pela experiência acumulado de cada um como um todo [15].

Assim, inicia-se o algoritmo com n partículas, distribuindo-as aleatoriamente por um espaço de busca. Como cada partícula representa um indivíduo, estará também a representar uma possível solução do problema, armazenando a sua posição no espaço de busca x_i , uma velocidade de deslocamento v_i e a sua melhor posição encontrada até à iteração atual - p_{best_i} .

Já a melhor posição de todas as partículas será armazenada na variável (g_{best}), sendo considerada como a melhor posição global.

Neste algoritmo também entram em conta certos parâmetros como o peso de inércia (w), que tem como objetivo limitar a velocidade das partículas, o coeficiente cognitivo referente à melhor posição encontrada pela partícula (c_1) e o coeficiente social, referente à melhor posição encontrada pelo conjunto da população no algoritmo (c_2). Por fim, a taxa de velocidade, v_r , a que cada partícula se move por iteração [16].

O algoritmo PSO encontra-se então descrito na Figura 13.

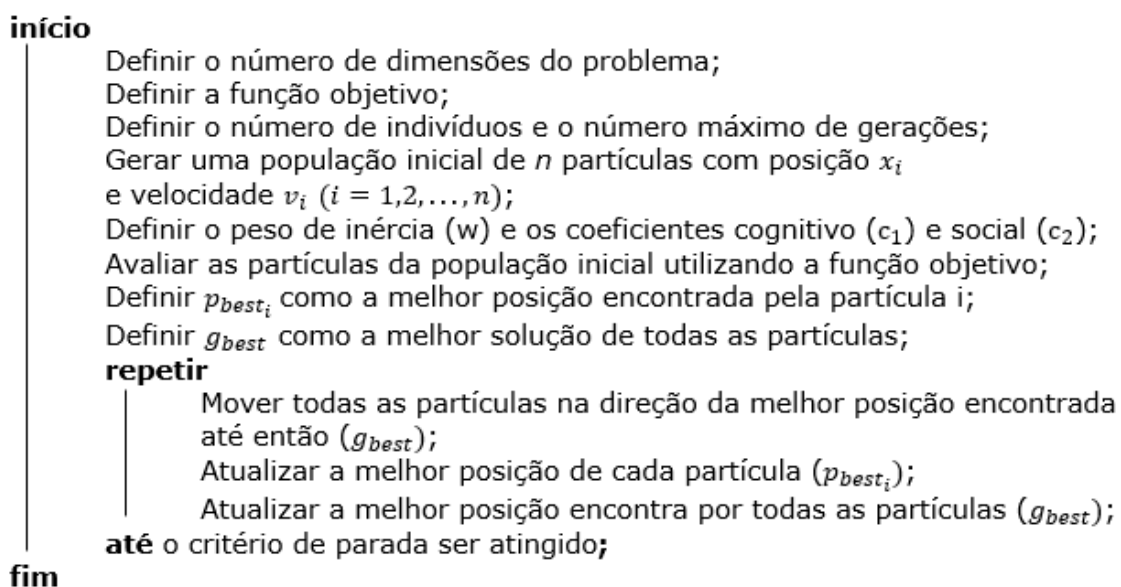


Figura 13 Algoritmo PSO

2.3.5. **GLOBAL PATTERN SEARCH (GPS)**

O algoritmo de otimização *Global Pattern Search* tem vindo a ser cada vez mais utilizado, uma vez que lida diretamente com quantidades de dados em grande escala. Aliado a isso, tenta descobrir padrões de movimento, comportamento e comunicação que possam indicar áreas ou atividades de interesse.

Este algoritmo representa uma família de métodos de otimização numérica que pode ser usado em funções que não são contínuas ou diferenciáveis. Um desses métodos de pesquisa de padrões é chamado de convergência, que se baseia na teoria das bases positivas. A otimização tenta então encontrar a melhor combinação (a solução que tem o menor valor de erro) num espaço de análise multidimensional de possibilidades.

Como demonstração, estudou-se o padrão das palavras mais utilizadas de vários artigos escolhidos aleatoriamente, sob uma área de estudo – Califórnia, tal como é explícito na Figura 14.

Desta maneira, cada artigo estudado é representado a partir de um vetor com comprimento V , sendo que contém o número de palavras distintas dentro do artigo. O objetivo deste vetor passa por conseguir transmitir o significado e a essência do que está exposto no artigo, com os termos que ficam guardados.

No entanto, há palavras que não trazem conteúdo para o artigo, uma vez que o único propósito é formar a sintaxe das frases escritas. Assim sendo, é necessária uma seleção de cada frase, de modo a filtrar e reduzir os termos guardados no vetor, para que estes sejam realmente importantes e para que o estudo não seja manipulado por palavras não relevantes sobre o tema.

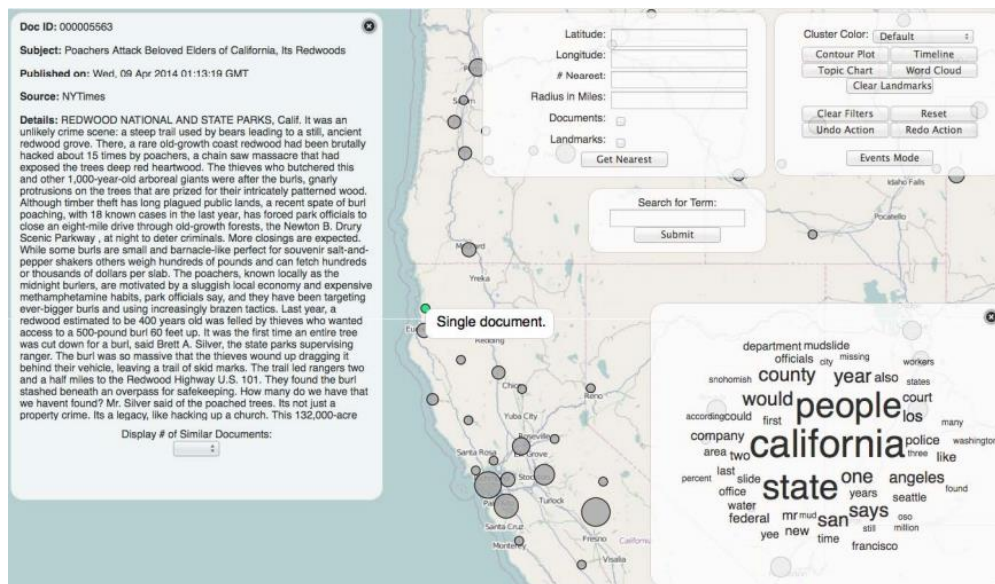


Figura 14 Área de estudo do algoritmo

Assim é possível formar um padrão com as palavras chaves presentes em cada artigo, permitindo não só perceber os temas que se delineiam a globalidade dos artigos estudados, assim como fazer uma comparação entre os mesmos e perceber a importância individual de cada tema aliado ao grupo de artigos escolhidos para estudo [17].

3. A EVOLUÇÃO ROBÓTICA

A dissertação proposta foca-se na área da modelação e projeto do braço robótico com 2 graus de liberdade, para aplicações que recorrem a movimentos precisos e otimizados. No decorrer deste processo, foi feito um estudo da história da robótica, assim como do modelo com duas juntas, enaltecendo as suas principais características.

3.1. ANÁLISE INTRODUTÓRIA E ORIGEM

A fim de se perceber a estrutura, função e a interação dos robôs, é necessário examinar um pouco da sua história e da constante evolução que foram sofrendo ao longo de vários anos.

Desde bem cedo que a criação de um processo ou uma tarefa automatizada foi do interesse do ser humano, não só pela utilidade que a si advém, mas também pela complexidade e o desafio que está em si inerente.

Ao que tudo indica, os primeiros processos autónomos foram encontrados a partir da mitologia grega, no entanto é necessário avançar uma boa quantidade de tempo para se realmente ter acesso aos primórdios dos robôs móveis, tal como descrito pela Figura 15.

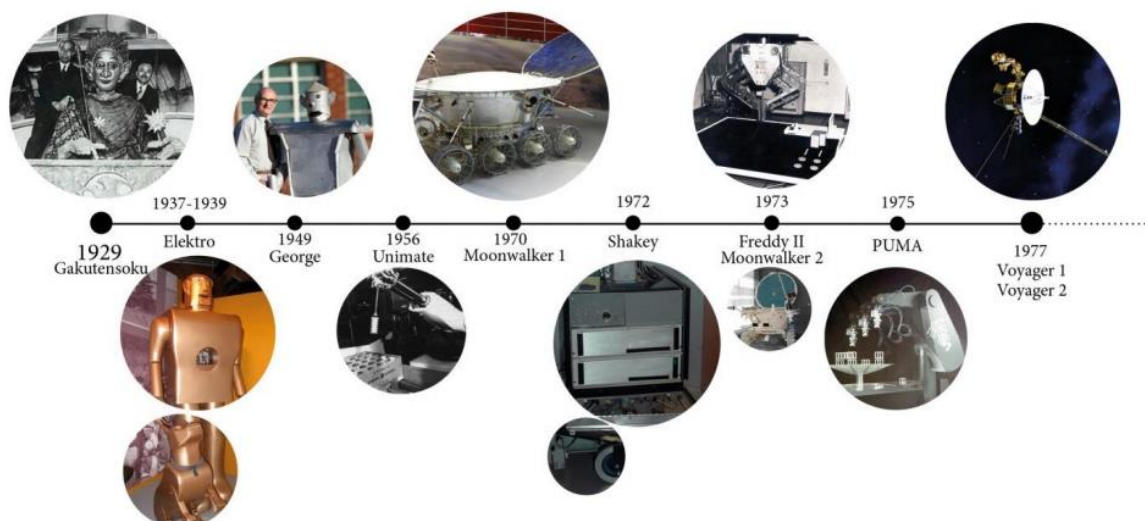


Figura 15 Linha temporal dos robôs (1929-1977) [18]

Um dos primeiros modelos a ser evidenciados, passou por terras orientais, mais concretamente no Japão durante o final da década de 1920, ao ser criado o robô *Gakutensoku*, que possuía a característica de mudar a sua expressão facial e mover ligeiramente as mãos e a cabeça. Em primeira instância, o propósito deste modelo servia apenas para divertir as pessoas em exposições públicas, sendo que com algum aperfeiçoamento, este robô tornou-se capaz de replicar diversas emoções como chorar, rir, piscar o olho ou escrever com a sua mão direita, a qual segurava uma caneta.

No que toca às suas dimensões, este robô apresentava cerca de 3,5 metros de altura e 3 metros de largura. Em suma o tamanho do robô estava diretamente interligado ao tamanho dos mecanismos e autómatos que dele faziam parte, uma vez que na época em questão não havia qualquer optimização [19].

Durante os próximos 20 anos, acabaram por ser criados dois robôs com funções similares: *Elektro* e *George*, sendo que a única diferença consistia no tamanho dos mesmos, uma vez que foram aperfeiçoados e o seu tamanho reduzido em comparação com o robô *Gakutensoku*.

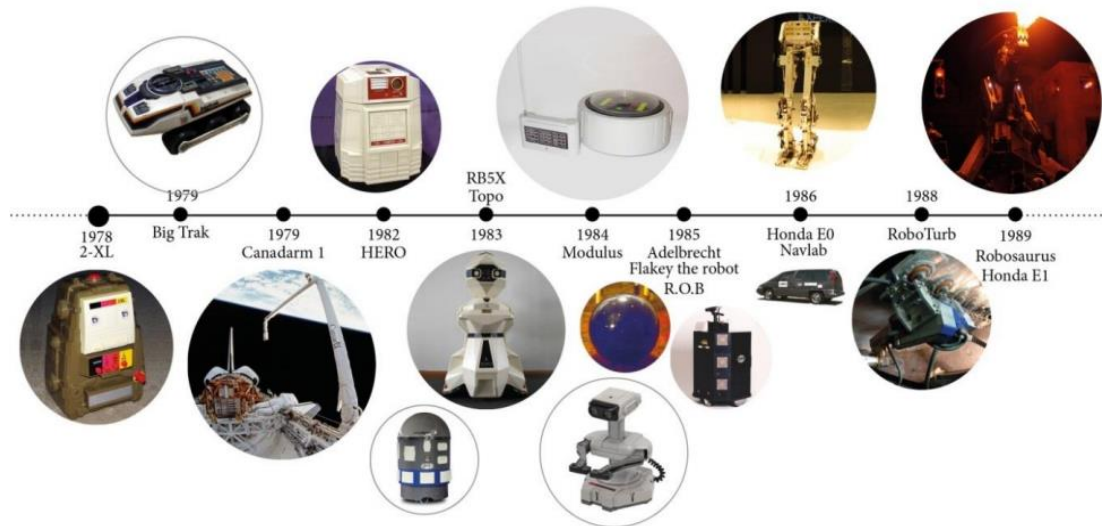


Figura 16 Linha temporal dos robôs (1978-1989) [18]

Já por volta da década de 60, o foco principal da robótica estaria interligado com o propósito espacial, uma vez que houve avanços significativos como a ida à lua e o lançamento de vários satélites para o espaço.

É então preciso avançar até 1986, como demonstrado pela Figura 16, até que a empresa Honda cria o modelo Honda E0, que acaba por ser tornar o primeiro robô desenvolvido com base nas mecânicas humanas.

Assim sendo, este modelo acaba por replicar os princípios fundamentais da locomoção bípede, no entanto apresentava diversas limitações, como o ciclo de dar um passo demorar cerca de 5 segundos.

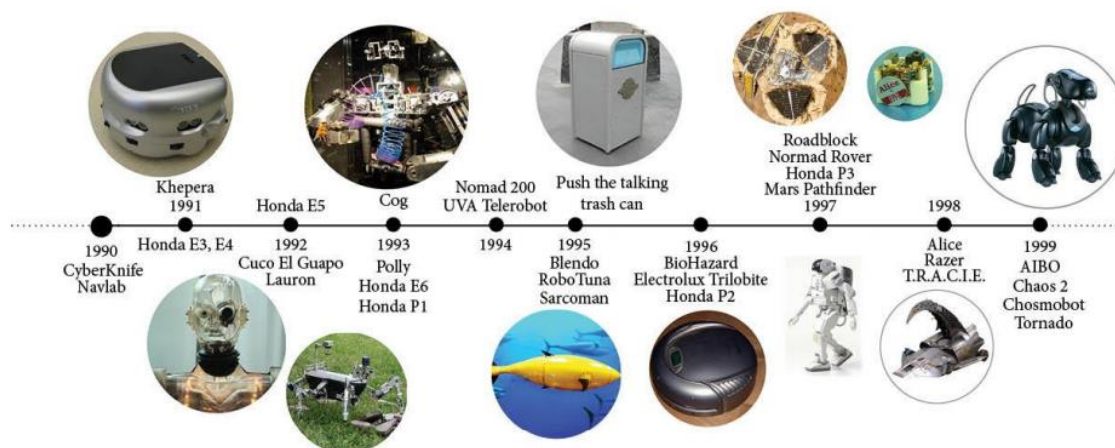


Figura 17 Linha temporal dos robôs (1990-1999) [18]

Através da Figura 17, é possível concluir que já em 1998, é criado um programa chamado *Robot Wars*, que consistia numa batalha épica entre dois robôs que eram criados especialmente com o intuito de danificar o seu adversário, de modo a saírem vitoriosos. Cada robô era então equipado com diversas armas, materiais e estruturas próprias para ter uma maior estabilidade e resistência durante a batalha, sendo que ambos eram colocados numa arena com espectadores no seu perímetro e a batalha só terminaria quando um dos robôs deixasse de funcionar.

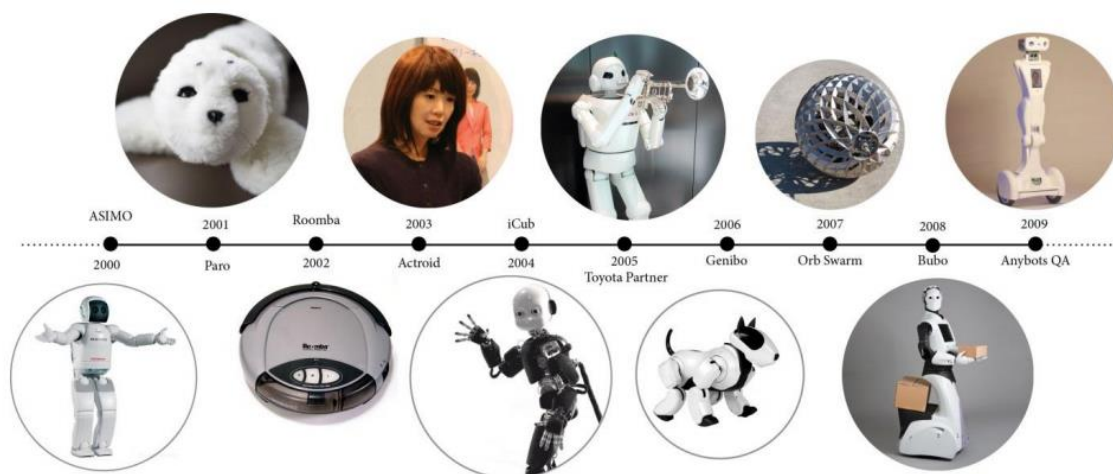


Figura 18 Linha temporal dos robôs (2000-2009) [18]

Já na entrada do novo milénio e com vários anos de experiência e de testes em robôs, a empresa HONDA acabou por lançar o seu segundo robô para o mercado, sobre o pretexto de ser o robô mais avançado da atualidade: ASIMO, como mostrado na Figura 18.

Este modelo tinha cerca de um metro de altura e uma estrutura física praticamente igual à do ser humano. Aliado a isso também a sua locomoção era bastante similar e era capaz de realizar várias tarefas simples dentro de um lar, como mexer em interruptores, abrir uma maçaneta, sentar-se numa cadeira, etc [20].

Outro grande avanço até à data, foi a mudança de direção no que toca à locomoção, uma vez que até à data os robôs tinham imensa dificuldade em realizar curvas apertadas, no entanto este modelo apresentava várias articulações no seu quadril que permitiria ultrapassar este obstáculo por completo e realizar vários padrões e trilhos de dificuldade avançada.



Figura 19 Linha temporal dos robôs (2010-2015) [18]

Já em 2015, fonte da Figura 19, foi anunciado o modelo Fleye, como o robô mais seguro do mundo.

Este modelo consiste num drone com uma hélice integrada, mas escondida de modo a evitar lesões, uma vez que as hélices expostas nos drones tradicionais podem apresentar algum perigo caso uma das suas hélices embata com qualquer obstáculo [20].

É possível ver e comparar a evolução e o crescimento exponencial que ocorreu num período de aproximadamente 100 anos. Os robôs ficaram mais inteligentes, menores, otimizados e mais seguros para o ser humano.

Igualmente desafiante é o futuro, o qual não apresenta limitações e possivelmente nos poderá mostrar outra evolução igual ou até superior, no que toca à robótica daqui a outros 100 anos.

3.2. ROBÔS INDUSTRIAIS

Desde bem cedo foi possível concluir que os robôs teriam um papel indispensável no que toca à evolução industrial. Assim, estas máquinas acabariam por executar tarefas que possam exigir alta precisão, velocidade ou até mesmo resistência, coisa que o ser humano não consegue competir.

Aliadamente, com a evolução dos manipuladores robóticos, houve também um crescimento dos robôs inteligentes, o que por si só gerou uma maior autonomia, independência e eficácia na concretização das tarefas que lhes estavam propostas.

Através da Figura 20, conclui-se que um dos primórdios robôs destinados à robótica é o modelo Versatran, construído pela empresa AMF Corporation, que consistia num braço robótico com estrutura coordenada cilíndrica programável [21].

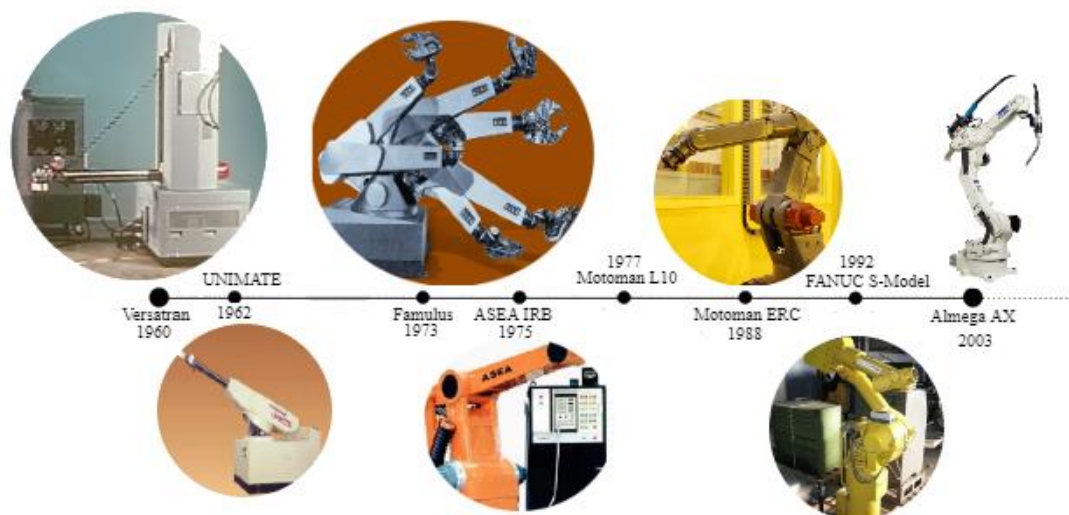


Figura 20 Linha temporal dos robôs industriais (1960-2003) [18]

Já em 1975, nasce o robô ASEA IRB, construído pela empresa europeia ASEA. Este modelo é recordado como o primeiro robô totalmente elétrico do mundo. Também foi o primeiro robô controlado por um microprocessador, usando, curiosamente, o primeiro *chipset* da Intel.

É então necessário chegar à década de 80 para se começar a produzir robôs em larga escala e todos eles controlados por microprocessadores.

Atenção especial para o modelo Motoman ERC, criado em 1988, que apresenta pela primeira vez 12 eixos e várias melhorias comparativamente o modelo prévio – Motoman L10.

Já à entrada do novo milênio, a empresa OTC DAIHEN lança para o mercado o robô Almega AX series, uma linha de robôs para soldadura a arco [21].

Desde o início dos anos 2000, os desenvolvimentos na robótica industrial têm sido amplamente impulsionados por avanços em *software*. Campos emergentes, como *Machine Learning* (ML) e *Artificial Intelligence* (AI), estão agora a dar liberdade ao que os robôs podem e conseguem fazer, dando-lhes a capacidade de aprender, melhorar e tomar decisões sem qualquer direção ou orientação do ser humano, como apresenta a Figura 21.

A maioria dos robôs industriais de hoje em dia, encontra-se equipada com uma vasta gama de sensores avançados que reúnem imensa quantidade e variedade de dados. Assim, esses dados podem ser integrados com *software* de ML, permitindo que os robôs consigam interpretar esses dados e usá-los para se adaptarem e evoluírem, alterando os movimentos mecânicos e completando a tarefa em mãos da melhor maneira possível.



Figura 21 Gama atual de robôs industriais [22]

Nos próximos anos, conforme os robôs industriais se tornem ainda mais inteligentes, acabarão por ser capazes de assumir tarefas ainda mais complexas e executá-las com uma eficácia que em nada pode ser comparada com as capacidades humanas.

4. O ROBÔ RR

É conhecido que os robôs manipuladores são sistemas altamente não lineares, acoplados dinamicamente e que variam no tempo, sendo amplamente utilizados nas indústrias de hoje.

Os manipuladores robóticos geralmente estão sujeitos a incertezas, e, por causa dessas incertezas e dos seus comportamentos não lineares, é considerada uma tarefa desafiadora, controlar o movimento de um robô manipulador numa posição precisa. Para uma tarefa com precisão, qualquer robô deve seguir os padrões predefinidos e as trajetórias o mais próximo possível. Assim, acaba por se tornar o controle da trajetória, a tarefa mais importante no controle de manipulador robótico em si [23].

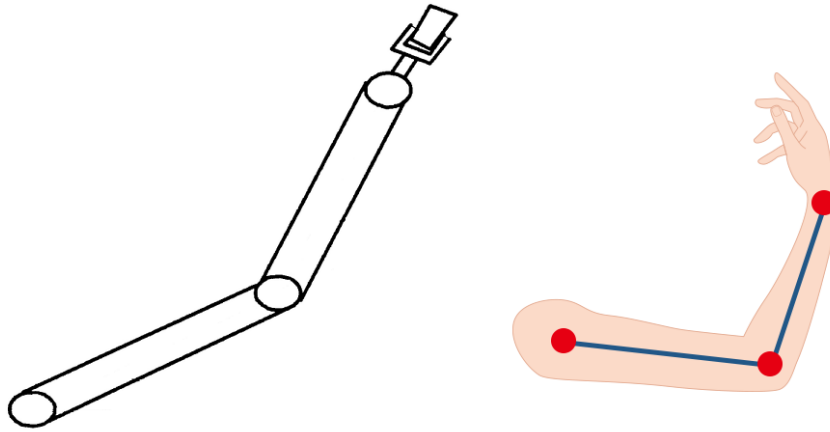


Figura 22 Robô RR e braço humano

O manipulador robótico de dois graus de liberdade (RR) age como um braço humano, tal como mostra a Figura 22. Para entender a complexidade dos braços humanos é possível equiparar à própria complexidade do movimento do robô manipulador de dois graus de liberdade [23].

A energia total que é a soma da energia cinética e energia potencial dos dois sistemas de ligação são definidos e usados para formar equações de Lagrange ou Newton-Euler, como serão explicados em maior pormenor a seguir.

4.1. MODELAÇÃO DO ROBÔ

Um robô, na sua generalidade, pode ser caracterizado por dois tipos de equações modelo: dinâmico e cinemático.

Tal como a palavra sugere, este primeiro representa o estudo do movimento de um robô, tendo em consideração as forças que estão a ser modeladas. Já o modelo cinemático diverge no sentido que é representado o estudo da matemática do movimento, desprezando qualquer força que possa afetar o mesmo.

Assim sendo, no modelo dinâmico é estudado a energia e velocidade que estão diretamente associadas ao movimento do robô com o intuito da obtenção de um modelo matemático capaz de descrever a trajetória do mesmo, enquanto no modelo cinemático, este lida com as relações geométricas que controlam o sistema, assim como relações entre parâmetros de controlo e comportamento de um sistema no seu espaço.

Simplificando, a cinemática robótica é essencial para descrever a posição, a orientação e o movimento de todas as juntas do robô enquanto que a modelação dinâmica é crucial para analisar e sintetizar o comportamento dinâmico do mesmo [24].

4.2. MODELO CINEMÁTICO

No que toca à cinemática direta, esta encontra-se diretamente dependente do valor no espaço cartesiano, como mostra a Figura 23.

Enquanto a cinemática direta resulta do desenvolvimento imediato das expressões do manipulador, a cinemática inversa procura determinar o conjunto de valores das juntas que se adequam a uma dada configuração do espaço operacional ou cartesiano. A cinemática inversa pode ser vista conceptualmente como o conjunto de processos para determinar as funções inversas do sistema das expressões da cinemática direta. Ou simbolicamente:

$$\vec{r} = \mathbf{F}(\vec{q}), \quad \vec{q} = \mathbf{F}^{-1}(\vec{r}), \quad \mathbf{F}^{-1}: \mathbb{R}^n \mapsto \mathbb{R}^6 \quad (1)$$

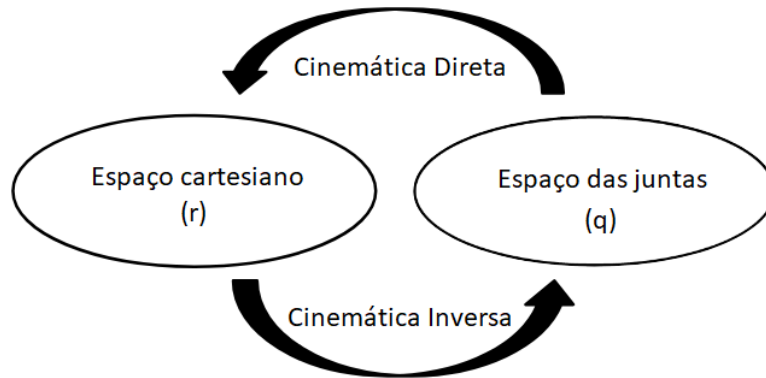


Figura 23 Relação de espaços entre cinemática direta e inversa

4.2.1. CINEMÁTICA DIRETA E INVERSA DO MANIPULADOR RR

No que toca à cinemática direta, esta pode ser rapidamente calculada a partir das juntas θ_1 e θ_2 descritas na Figura 24.

Para tal, resta necessariamente procurar as duas soluções:

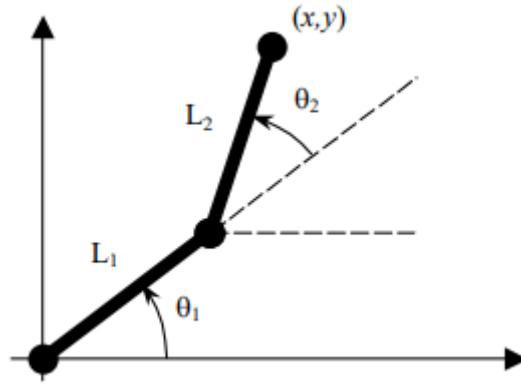


Figura 24 Manipulador RR (2 graus de liberdade)

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad r = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} \quad (2)$$

De onde pode-se concluir [27]:

$$x = L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2) \quad (3)$$

$$y = L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2) \quad (4)$$

$$\phi = \theta_1 + \theta_2 \quad (5)$$

Inicia-se então o processo de determinar a cinemática inversa do manipulador em estudo, permitindo calcular o valor das suas juntas θ_1 e θ_2 em função da posição (x,y) da sua extremidade.

Partindo das equações da cinemática direta, que são de fácil formulação, e que foram anteriormente deduzidas, o processo é o seguinte:

$$\begin{cases} x_1 = L_1 \sin(\theta_1) \\ y_1 = L_1 \cos(\theta_1) \\ x_2 = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \\ y_2 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \end{cases} \quad (6)$$

Elevando ambos os membros das duas expressões iniciais ao quadrado, e adicionando-os, fica:

$$\begin{aligned}
x^2 + y^2 &= L_1^2 \cos^2 \theta_1 + L_1^2 \sin^2 \theta_1 \\
&= L_2^2 \sin^2(\theta_1 + \theta_2) + L_2^2 \cos^2(\theta_1 + \theta_2) \\
&= 2L_1 L_2 \cos \theta_1 \cos(\theta_1 + \theta_2) + 2L_1 L_2 \sin \theta_1 \sin(\theta_1 + \theta_2)
\end{aligned} \tag{7}$$

Virá então sucessivamente:

$$\begin{aligned}
x^2 + y^2 &= L_1^2 + L_1^2 \\
&+ 2L_1 L_2 [\cos \theta_1 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\
&+ \sin \theta_1 (\sin \theta_1 \cos \theta_2 + \sin \theta_2 \cos \theta_1)]
\end{aligned} \tag{8}$$

$$\begin{aligned}
x^2 + y^2 &= L_1^2 + L_1^2 \\
&+ 2L_1 L_2 (\cos^2 \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_1 \sin \theta_2 \\
&+ \sin^2 \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 \cos \theta_1)
\end{aligned} \tag{9}$$

$$x^2 + y^2 = L_1^2 + L_1^2 + 2L_1 L_2 \cos \theta_2 \tag{10}$$

$$\theta_2 = \pm \arccos \frac{x^2 + y^2 - L_1^2 - L_1^2}{2L_1 L_2} \tag{11}$$

Pode verificar-se desde já a existência matemática de duas soluções para θ_2 reforçada pelo sinal $\pm$ na expressão. A obtenção de θ_1 não é tão imediata e uma solução possível passa pelo recurso à tangente trigonométrica de uma diferença angular, como se pode observar pela Figura 25.

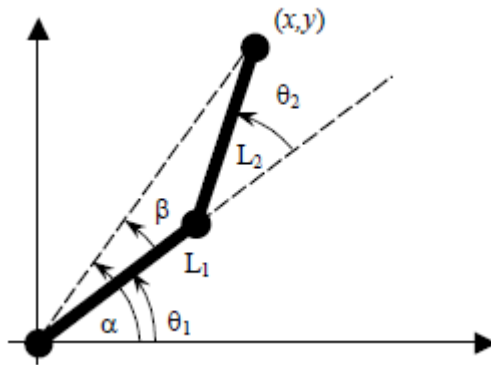


Figura 25 Ângulos auxiliares do manipulador RR

$$\tan(A - B) = \frac{\tan A - \tan B}{1 + \tan A \tan B} \quad (12)$$

$$\tan(\beta) = \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} \quad (13)$$

$$\tan(\alpha) = \frac{y}{x} \quad (14)$$

Através da Figura 25, é possível concluir que $\theta_1 = \alpha - \beta$, pelo que, através da tangente da diferença, é possível concluir [28]:

$$\begin{aligned} \tan(\alpha - \beta) = \tan(\theta_1) &= \frac{\frac{y}{x} - \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}}{1 + \frac{y}{x} \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}} \\ &= \frac{y(L_1 + L_2 \cos \theta_2) - xL_2 \sin \theta_2}{x(L_1 + L_2 \cos \theta_2) - yL_2 \sin \theta_2} \end{aligned} \quad (15)$$

Ou seja,

$$\theta_1 = \arctan \left[\frac{y(L_1 + L_2 \cos \theta_2) - xL_2 \sin \theta_2}{x(L_1 + L_2 \cos \theta_2) - yL_2 \sin \theta_2} \right] \quad (16)$$

4.3. MODELO DINÂMICO

O modelo dinâmico de um robô preocupa-se com o movimento e as forças envolvidas no braço do mesmo, estabelecendo relação matemática entre a localização das juntas do robô e os parâmetros dimensionais do próprio.

Para que seja possível calcular o seu modelo dinâmico, recorre-se a duas abordagens que acabam por gerar modelos equivalentes: Lagrange e Newton-Euler [25].

A primeira utiliza como equações base:

$$\mathbf{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T}(\dot{\mathbf{q}}) - \mathbf{U}(\mathbf{q}) \quad (17)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) + \frac{\partial L}{\partial q} = \eta \quad (18)$$

Onde:

$\mathbf{q}$: *Vetor de configuração do robô*

$\dot{\mathbf{q}}$: *Velocidade do robô*

$U(\mathbf{q})$: *Energia cinética*

$T(\dot{\mathbf{q}})$: *Energia potencial*

η : *Vetor de forças generalizadas*

Já a abordagem Newton-Euler utiliza as seguintes equações:

$$\mathbf{F} = m\ddot{\mathbf{x}} \quad (19)$$

$$\mathbf{N} = \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \quad (20)$$

Onde:

m : *Massa total do robô*

$\ddot{\mathbf{x}}$: *Aceleração do centro de massa*

$\boldsymbol{\omega}$: *Velocidade angular com origem no centro de massa*

$\mathbf{I}$: *Matriz 3x3 representativa da massa do robô*

$\mathbf{F}$: *Força em atuação no centro de massa do corpo*

$\mathbf{N}$: *Binário em atuação no centro de massa do corpo*

É também possível calcular a energia cinética e potencial do sistema robótico que é projetado [25]:

$$KE = \frac{1}{2} (M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_1^2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 + M_2 L_1 L_2 \cos(\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2) \quad (21)$$

$$PE = M_1 g L_1 \cos(\theta_1) + M_2 g (L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)) \quad (22)$$

Após a simplificação e a transição das equações cartesianas para polares na equação da energia cinética, a equação que descreve o movimento do manipulador do braço do robô considera-se:

$$B(q)\ddot{q} + C(\dot{q}, q) + g(q) = F \quad (23)$$

Onde:

$$q = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \quad (24)$$

Para um sistema de um manipulador robótico com n ligações em série, onde $B(q) \in R^{n \times n}$ é uma matriz de inércia definida, $C(\dot{q}, q) \in R^n$ representa o vetor de forças centrípetas, $g(q)$ é a matriz de gravidade e, por fim, $F \in R^n$ consiste no binário aplicado a cada uma das juntas.

Todas estas matrizes acabam por descrever o movimento dinâmico do sistema robótico:

$$B(q) = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \quad (25)$$

Assim:

$$B_{11} = (M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2 \cos(\theta_2) \quad (26)$$

$$B_{12} = B_{21} = M_2L_2^2 + 2M_2L_1L_2 \cos(\theta_2) \quad (27)$$

$$B_{22} = M_2L_2^2 \quad (28)$$

Já as matrizes de inércia de gravidade são:

$$C(\dot{q}, q) = \begin{pmatrix} -M_2L_1L_2 \sin(\theta_2)(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ -M_2L_1L_2 \sin(\theta_2)(\dot{\theta}_1\dot{\theta}_2) \end{pmatrix} \quad (29)$$

$$g(q) = \begin{pmatrix} -(M_1 + M_2)gL_1 \sin(\theta_1) - M_2gL_2 \sin(\theta_1 + \theta_2) \\ -M_2gL_2 \sin(\theta_1 + \theta_2) \end{pmatrix} \quad (30)$$

E a matriz de binário [26]:

$$F = \begin{pmatrix} f_{\theta_1} \\ f_{\theta_2} \end{pmatrix} \quad (31)$$

4.4. CONTROLADOR PID

O controlador Proporcional-Integral-Derivativo (PID) tem a função de ajustar automaticamente uma saída de controlo com base na diferença entre um ponto de ajuste (*set point* - SP) e uma variável de processo (*process variable* – PV).

O algoritmo do mesmo envolve três parâmetros constantes separados, sendo também chamado de controlo de três termos: o proporcional, o integral e o derivativo – P, I e D.

Destes três parâmetros, P depende do erro presente, I no acumular de erros passados e D representa uma previsão de erros futuros. Assim, somando as três ações, é possível ajustar qualquer processo por meio de um elemento de controlo [29].

Algumas aplicações podem apenas exigir o uso de apenas uma ou duas ações para fornecer o sistema apropriado de controlo. Daí divergem controladores como PI, PD, ou P.

Daí advêm três algoritmos do controlador onde o valor de saída do controlador $u(t)$ é transferido como a entrada do sistema em que:

1) Algoritmo iterativo (Figura 26)

$$u(t) = K_c \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right] \times \left[1 + T_d \frac{d}{dt} e(t) \right] \quad (32)$$

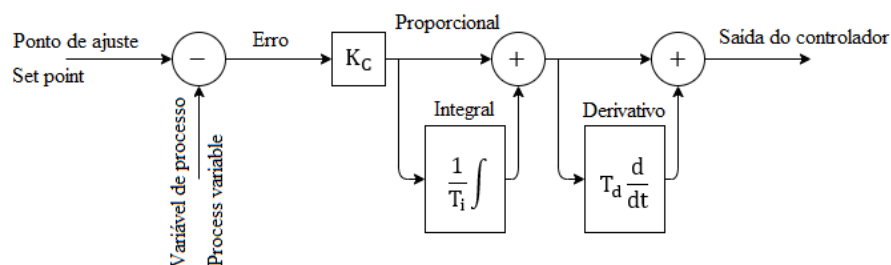


Figura 26 Algoritmo iterativo

2) Algoritmo não iterativo (Figura 27)

$$u(t) = K_c \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right] \quad (33)$$

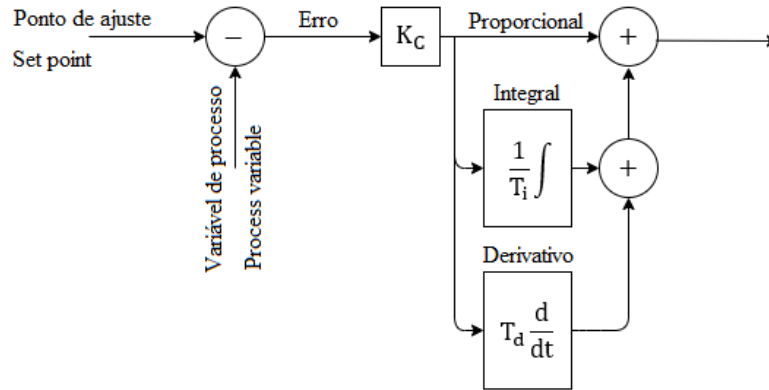


Figura 27 Algoritmo não iterativo

3) Algoritmo paralelo (Figura 28)

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (34)$$

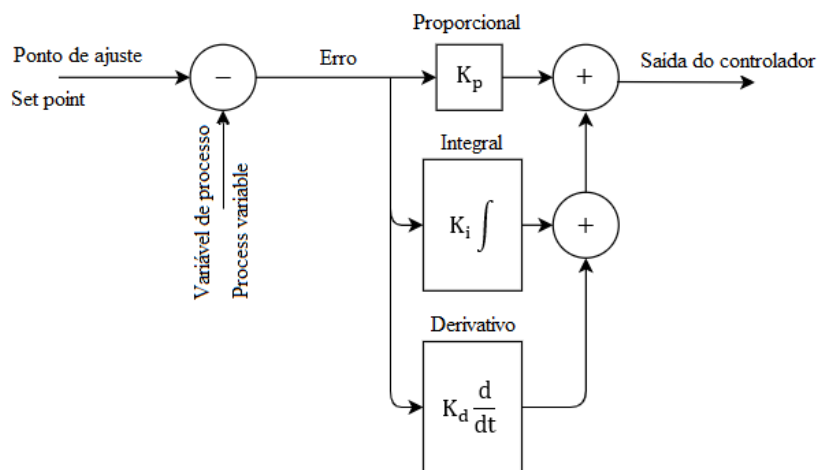


Figura 28 Algoritmo paralelo

Onde [30]:

$$K_p = K_c : \text{Ganho proporcional}$$

$$K_i = \frac{K_c}{T_i} : \text{Ganho Integral}$$

$$K_d = K_c T_d : \text{Ganho derivativo}$$

$$e(t) = r(t) - y(t)$$

Neste cenário em particular, o controlador PID tem como principal função atuar sobre a variável a ser manipulada, neste caso a trajetória pedida pelo utilizador, através de uma combinação adequada de três ações de controlo. A maior parte dos controladores industriais de hoje em dia, utilizam a configuração PID. Este método é particularmente utilizado, quando o modelo matemático do sistema não é conhecido e, como tal, os métodos analíticos não podem ser usados, acabando o controlador PID revelar-se como a ferramenta mais útil para este tipo de situação.

Ou seja, para o sistema que se pretende criar, recorre-se ao algoritmo paralelo da Figura 28.

O objetivo agora passa por encontrar os coeficientes adequados de K_p , K_i e K_d para cada uma das juntas, a fim de minimizar os erros da trajetória [31].

Uma vez que este sistema apresenta duas juntas, é necessário calibrar ambos os controladores PID ao mesmo tempo, o que não se revelou uma tarefa simples, uma vez que a sintonia de um controlador irá diretamente afetar o outro.

Numa primeira instância, escolheu-se a abordagem do ‘PID Tuner’, uma ferramenta presente no Simulink que permite manualmente manipular a resposta do sistema de modo a obter-se o mais próximo do pretendido. No entanto, tal como dito anteriormente, ao calibrar-se um controlador iria por conseguinte descalibrar o outro, o que acabou por não se utilizar este método exposto na Figura 29.

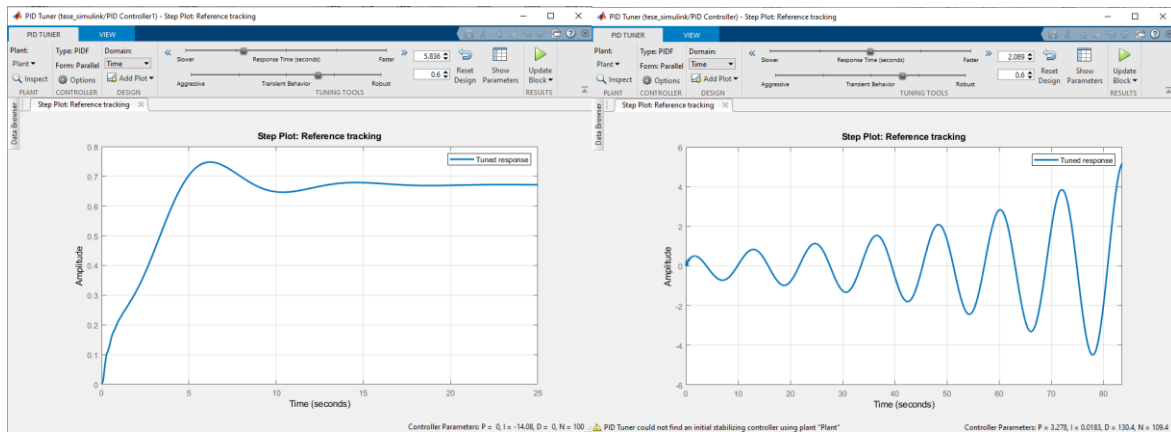


Figura 29 Comparação de ambos controladores PID usando o método do PID Tuner

Assim, como alternativa, aplicaram-se três algoritmos de otimização diretamente em cada controlador de cada junta, de modo a conseguir configurar ambos os controladores e aperfeiçoar o binário de saída de controlo de cada um, que irá controlar o braço do robô, tal como demonstrado na Figura 30. Assim será possível dar uma resposta otimizada por parte do braço robótico e irá melhorar qualquer trajetória que o braço descreva.

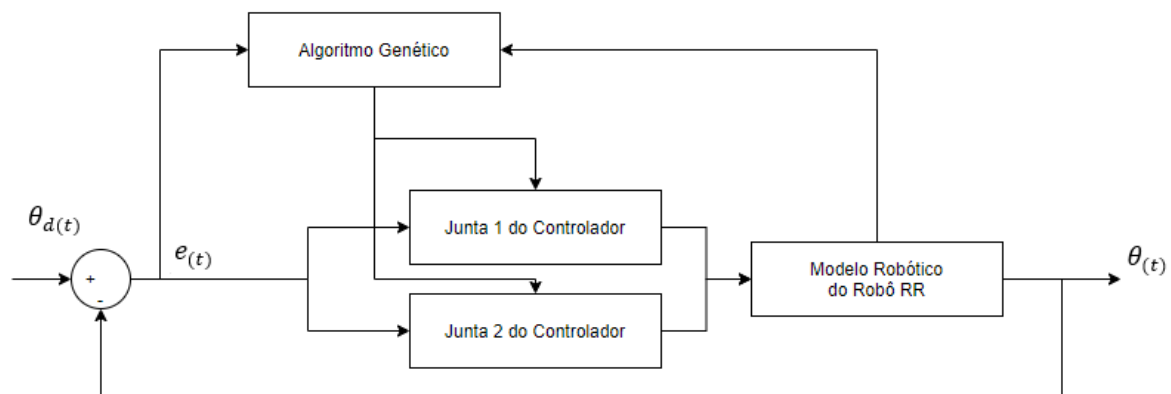


Figura 30 Aplicação do AG no Modelo Robótico

5. IMPLEMENTAÇÃO DO SISTEMA

Neste capítulo será discutida a implementação do sistema, focando-se detalhadamente em todas as partes que constituem o mesmo. Será explicado cada um dos blocos existentes na Figura 31, a sua configuração e o seu efeito no geral.

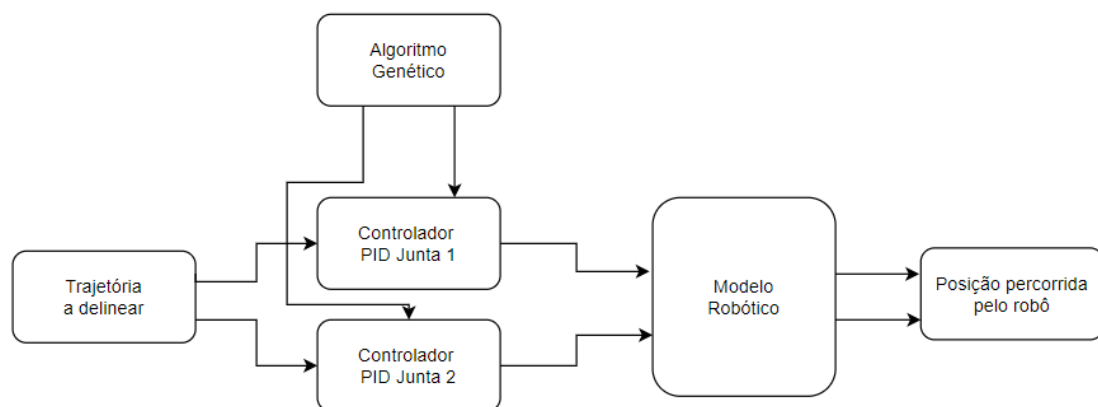


Figura 31 Diagrama de blocos do sistema

O sistema simulado e exposto na ferramenta Simulink pode ser dividido em três blocos, cada um representando um subsistema – Trajetória, Controlador e Modelo Robótico, como demonstrado pela Figura 32. Este modelo encontra-se também colocado no Anexo A.

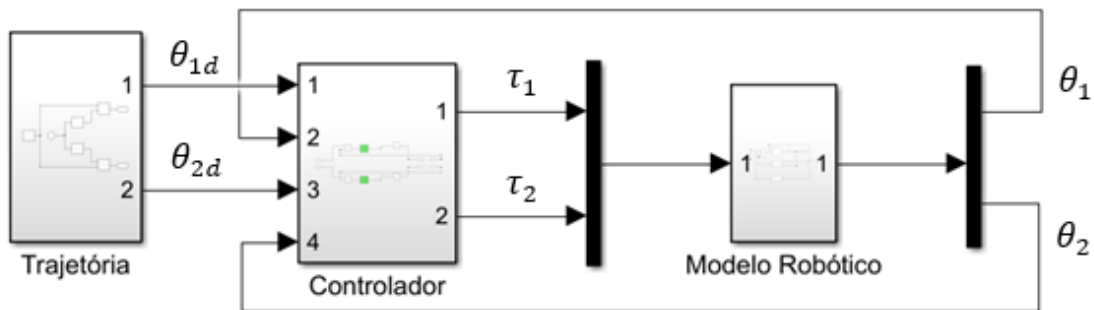


Figura 32 Implementação do sistema em Simulink

5.1. TRAJETÓRIAS PERCORRIDAS

Na Figura 33 estão demonstradas as duas trajetórias que foram escolhidas para testar a manipulação do braço do robô.

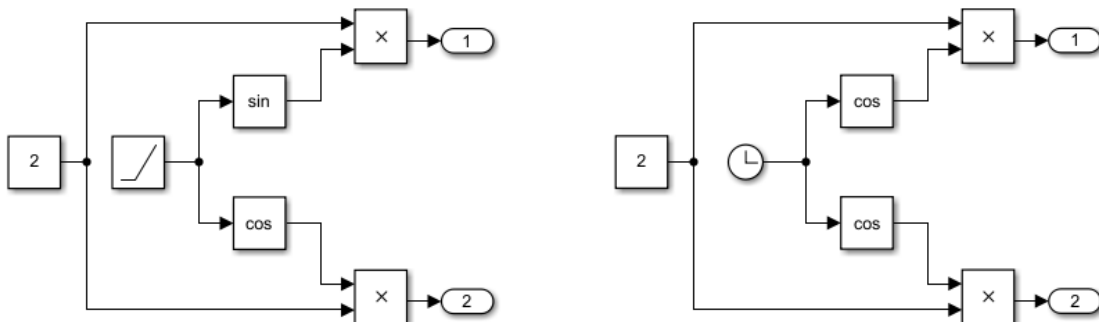


Figura 33 Trajetória circular (esquerda) e linear (direita) estudadas

O primeiro bloco consiste na delineação da trajetória que é pretendida que o robô percorra. Os dois exemplos estudados foram um círculo de raio 2 e centro em $[0,0]$ e uma linha recta com os pontos iniciais em $[2,2]$ e finais em $[-2,2]$, no espaço das juntas, como mostra a Figura 34.

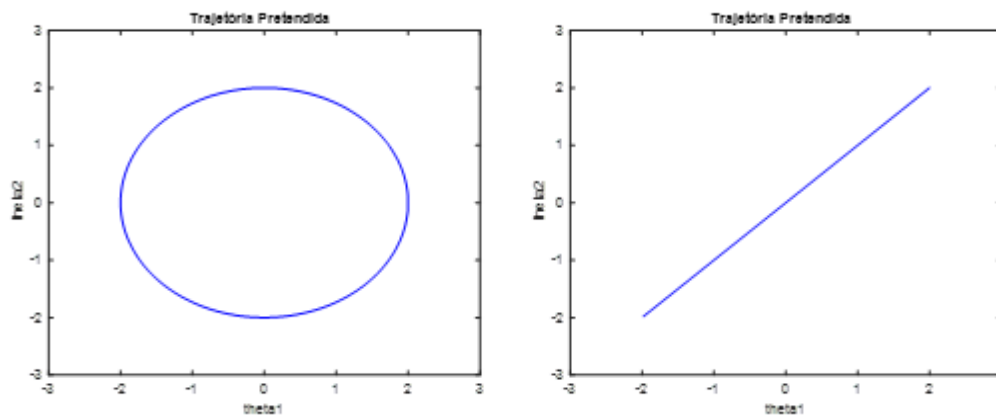


Figura 34 Trajetórias estudadas no plano XY: círculo e recta

5.2. CONTROLADOR

No que toca ao subsistema do controlador, apresentado na Figura 35, este recebe como entrada os valores iniciais de ambas as posições angulares, assim como as posições reais angulares que o robô percorre. O erro resultante é aplicado a um controlador PID em cada junta assim como uma saturação de binário. Esta última tem o objetivo de colocar restrições de binário máximo com que cada junta pode operar, uma vez que o binário é limitado em cada junta de robô real. Assim, é dado como saída o binário de cada junta que irá alimentar o sistema do Modelo Robótico.

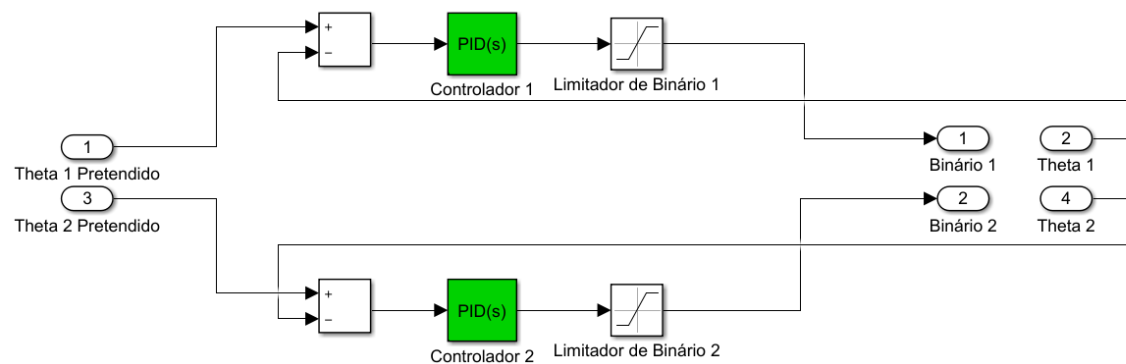


Figura 35 Subsistema do Controlador

5.3. MODELO ROBÓTICO

Quanto ao Modelo Robótico, este representa o cerne da implementação do sistema. É então alimentado pelos binários de cada junta e irá aplicar a Matriz de Inércia, Força Centrífuga e Força Gravítica, que já foram apresentadas e explicadas no Modelo Dinâmico e Cinemático de um robô com dois graus de liberdade no Capítulo 5, de modo a conseguir converter em aceleração. Integrando esse valor, será possível encontrar a velocidade, e a posição real que o robô percorre com uma segunda integração. Assim o robô percorrerá a trajetória de entrada o mais próximo possível. No entanto, a precisão da trajetória, como mostra a Figura 36, está diretamente dependente do controlo estar bem projetado, sendo que o algoritmo genético vai desempenhar o papel de catalisador de modo que os coeficientes K_p , K_i e K_d de cada junta consigam gerar o menor erro possível. O código dos três blocos do modelo robótico, encontra-se presente no Anexo B.

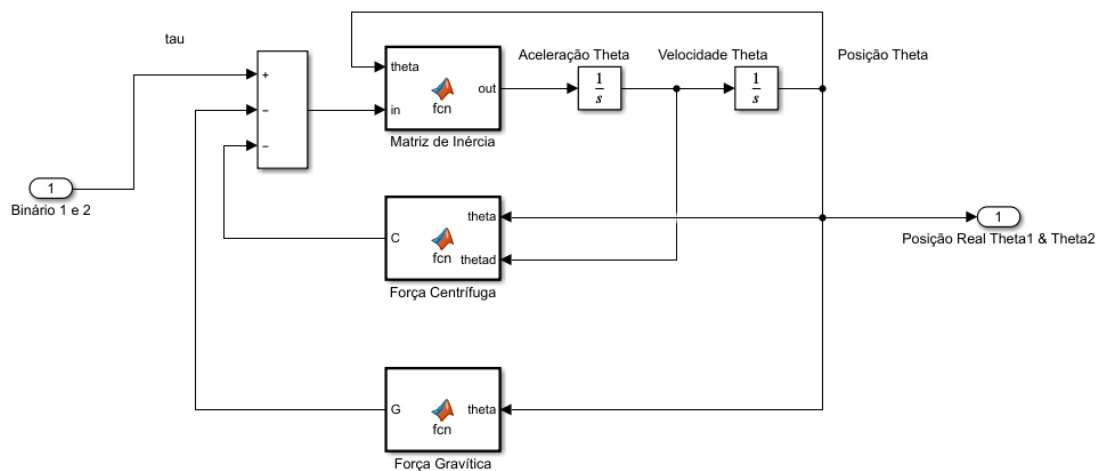


Figura 36 Modelo Robótico

A Tabela 2 apresenta os parâmetros de massa, comprimento e binário máximo do braço do robô. O binário foi calculado diretamente sob todos os outros parâmetros, de modo a desempenhar uma limitação que o braço apresenta na realidade. Também são definidos nesta tabela, os tempos necessários para percorrer cada uma das duas trajetórias impostas pelo utilizador.

Tabela 2 Parâmetros de configuração do robô

Parâmetros de configuração do robô		
	Junta 1	Junta 2
Massa (kg)	0,200	0,200
Comprimento (m)	1	1
Aceleração gravítica (m/s^2)	9,81	9,81
Binário máximo (Nm)	6,72	13,3
Tempo para percorrer trajetória circular (s)	7,00	7,00
Tempo para percorrer trajetória linear (s)	3,00	3,00

5.4. PARÂMETROS DE INICIALIZAÇÃO

No que toca aos parâmetros de inicialização dos algoritmos utilizados, estes encontram-se descritos na Tabela 3 e mantêm-se constantes nas simulações das duas trajetórias em estudo.

Tabela 3 Parâmetros de inicialização

Parâmetros de inicialização	
Tamanho da População	50
Número de Gerações	100
Limite dos valores de $[K_p; K_i; K_d]$	$lb = [0; 0; 0]$
	$ub = [5,000 \times 10^6; 1,900 \times 10^2; 5,000 \times 10^4]$

Para inicializar qualquer algoritmo, é preciso ter em conta vários conjuntos de parâmetros que vão acabar por quantificar e classificar o conjunto de soluções para o problema.

Como tal, é necessário gerar uma população e os seus limites, assim como os das variáveis a otimizar. Neste cenário em particular o tamanho da população é representado por 50 indivíduos. Em grande parte, este número deve-se às 6 variáveis que são necessárias otimizar

$(K_{p1}, K_{i1}, K_{d1}, K_{p2}, K_{i2}, K_{d2})$. Uma população maior iria provavelmente demonstrar resultados mais otimizados, no entanto iria também interferir no tempo de cada simulação, aumentando-o consideravelmente.

5.5. ÍNDICES DE DESEMPENHO

Para otimizar o desempenho de um sistema de controlo PID, os ganhos do controlador são ajustados para minimizar um determinado índice de desempenho. O índice de desempenho é calculado durante um intervalo de tempo, t , normalmente no intervalo de $0 \leq t \leq t_s$, onde t_s representa o tempo de estabelecimento do sistema. Os índices de desempenho (ID) utilizados foram o ITAE, IAE e ISE, a seguir descritos.

5.5.1. ITAE – INTEGRAL TIME ABSOLUTE ERROR

O índice ITAE (*Integral Time Absolute Error*) representa a soma das áreas acima e abaixo do valor de referência. Como critério é utilizado o módulo do erro, sendo que este índice penaliza erros que se mantêm ao longo do tempo. Pode ser definido a partir de:

$$I_{ITAE} = \int_0^{\infty} t|e(t)|dt \quad (35)$$

5.5.2. IAE – INTEGRAL ABSOLUTE ERROR

O índice IAE (*Integral Absolute Error*) representa a mesma abordagem que o anterior, no entanto não apresenta o vetor temporal presente no índice ITAE. Este índice de desempenho é definido por:

$$I_{IAE} = \int_0^{\infty} |e(t)|dt \quad (36)$$

5.5.3. ISE – INTEGRAL SQUARE ERROR

Por fim, o índice ISE (*Integral Square Error*) acaba por ser um critério inferior aos outros dois índices, uma vez que este ID tende a ter uma resposta mais rápida, o que faz diminuir a sua estabilidade e apresentar algumas oscilações. Matematicamente é definido por:

$$I_{ISE} = \int_0^{\infty} [e(t)]^2 dt \quad (37)$$

6. RESULTADOS

Neste capítulo será avaliado o código implementado pelos vários algoritmos utilizados, dando ênfase aos resultados obtidos em cada uma das suas variantes, assim como o bom desempenho do sistema. Será igualmente analisado toda e qualquer formulação matemática dos índices de desempenho, assim como os parâmetros dos erros entre a referência e a saída do sistema.

6.1. GENETIC ALGORITHM (GA)

O primeiro algoritmo a ser testado de modo a otimizar os valores dos controladores foi o GA. Em baixo segue-se uma tabela relativa aos ganhos dos controladores, para cada trajetória estudada, Tabela 4 e Tabela 6.

Posteriormente será dado os valores de desempenho das trajetórias, de acordo com os valores de cada controlador: Tabela 5 e Tabela 7.

Por fim, serão visualizadas as trajetórias reais, assim como os comportamentos de cada junta e os respetivos erros.

6.1.1. TRAJETÓRIA CIRCULAR

No que toca à optimização dos ganhos para cada controlador, a Tabela 4 permite dar uma noção global dos valores, assim como as variações de acordo com cada índice de desempenho.

É possível ver, numa vertente geral, que os ganhos K_{p1} e K_{p2} designam valores bastantes elevados, a par de K_{d1} e K_{d2} . Já os ganhos da parte integral do sistema, K_{i1} e K_{i2} apresentaram valores mais baixos no decorrer de todas as simulações.

Tabela 4 Ganhos do Controlador GA - Círculo

	Método	Ganhos do controlador 1			Ganhos do controlador 2		
		K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
GA	IAE	$4,427 \times 10^5$	$7,068 \times 10^1$	$6,451 \times 10^3$	$2,665 \times 10^5$	$1,606 \times 10^2$	$1,823 \times 10^4$
	ITAE	$3,260 \times 10^5$	$9,730 \times 10^1$	$4,895 \times 10^3$	$6,088 \times 10^3$	$1,789 \times 10^2$	$2,759 \times 10^2$
	ISE	$4,936 \times 10^5$	$1,895 \times 10^2$	$7,616 \times 10^3$	$4,641 \times 10^5$	$6,620 \times 10^1$	$5,381 \times 10^3$

No que toca ao estudo dos índices de desempenho do sistema, utilizando os valores acima declarados, é possível ver que, os índices do sistema apresentam valores muito baixos, o que quer dizer que a trajetória real é muito próxima da pretendida. Neste caso em particular, o índice que teve melhor desempenho foi o ISE, uma vez que o valor da segunda junta do robô neste cenário é praticamente inexistente.

Tabela 5 Desempenho da Malha GA - Círculo

	Método	Índice Junta 1	Índice Junta 2	Índice Global
GA	IAE	$3,457 \times 10^{-3}$	$3,388 \times 10^{-5}$	$3,490 \times 10^{-3}$
	ITAE	$3,514 \times 10^{-4}$	$5,050 \times 10^{-3}$	$5,401 \times 10^{-3}$
	ISE	$1,144 \times 10^{-4}$	$9,116 \times 10^{-11}$	$1,144 \times 10^{-4}$

Assim sendo, é possível visualizar as trajetórias descritas com os parâmetros otimizados e concluir que ambas as trajetórias são bastante similares, deixando uma margem de erro por parte das juntas do robô muito pequena, como é desejado.

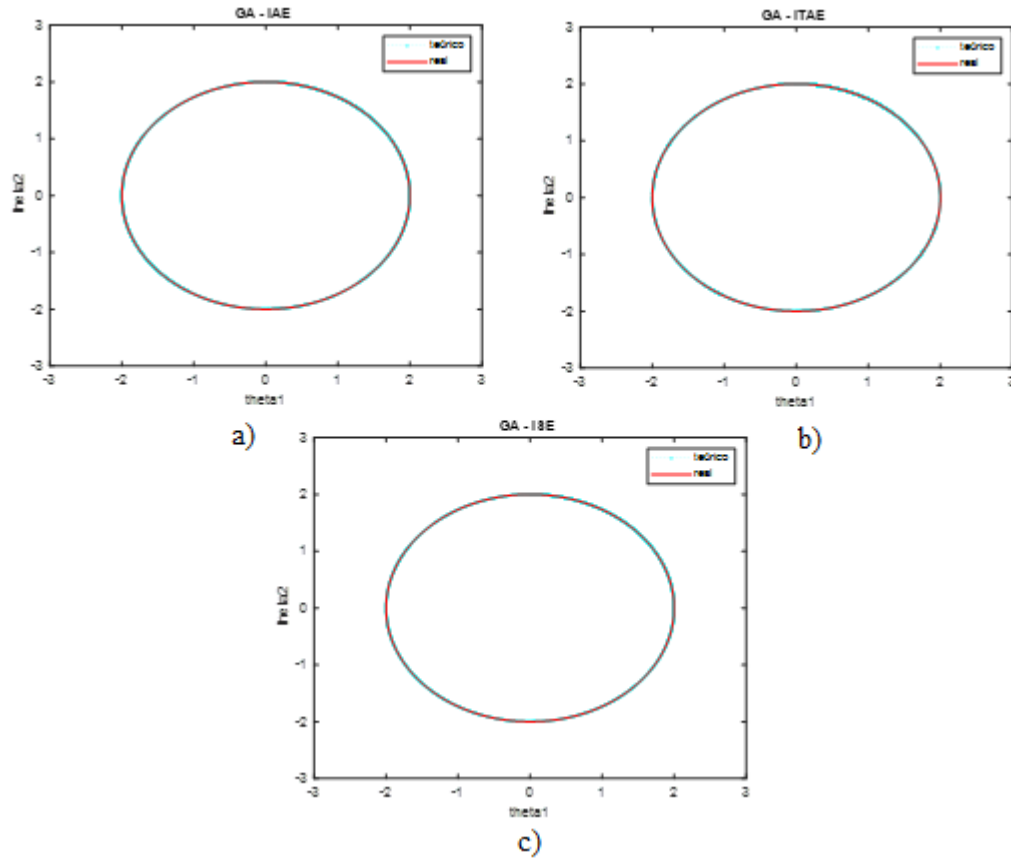


Figura 37 Trajetória circular descrita e esperada: a) GA – IAE; b) GA – ITAE; c) GA – ISE

De modo a visualizar-se da melhor maneira possível, decidiu-se colocar a trajetória desejada a azul ciano, e a trajetória real feita pelo manipulador robótico a vermelho. Assim torna-se possível ver a sobreposição de ambas as trajetórias, uma vez que há um contraste bastante grande entre as cores escolhidas da Figura 37.

É igualmente importante referir que, por norma, a maior discrepância entre ambas as trajetórias ocorre no início das mesmas, ou seja, no intervalo $[0,2]$.

Posteriormente, é também analisado o comportamento individual de cada junta, no tempo que percorre a trajetória a ser efetuada, como mostra a Figura 38.

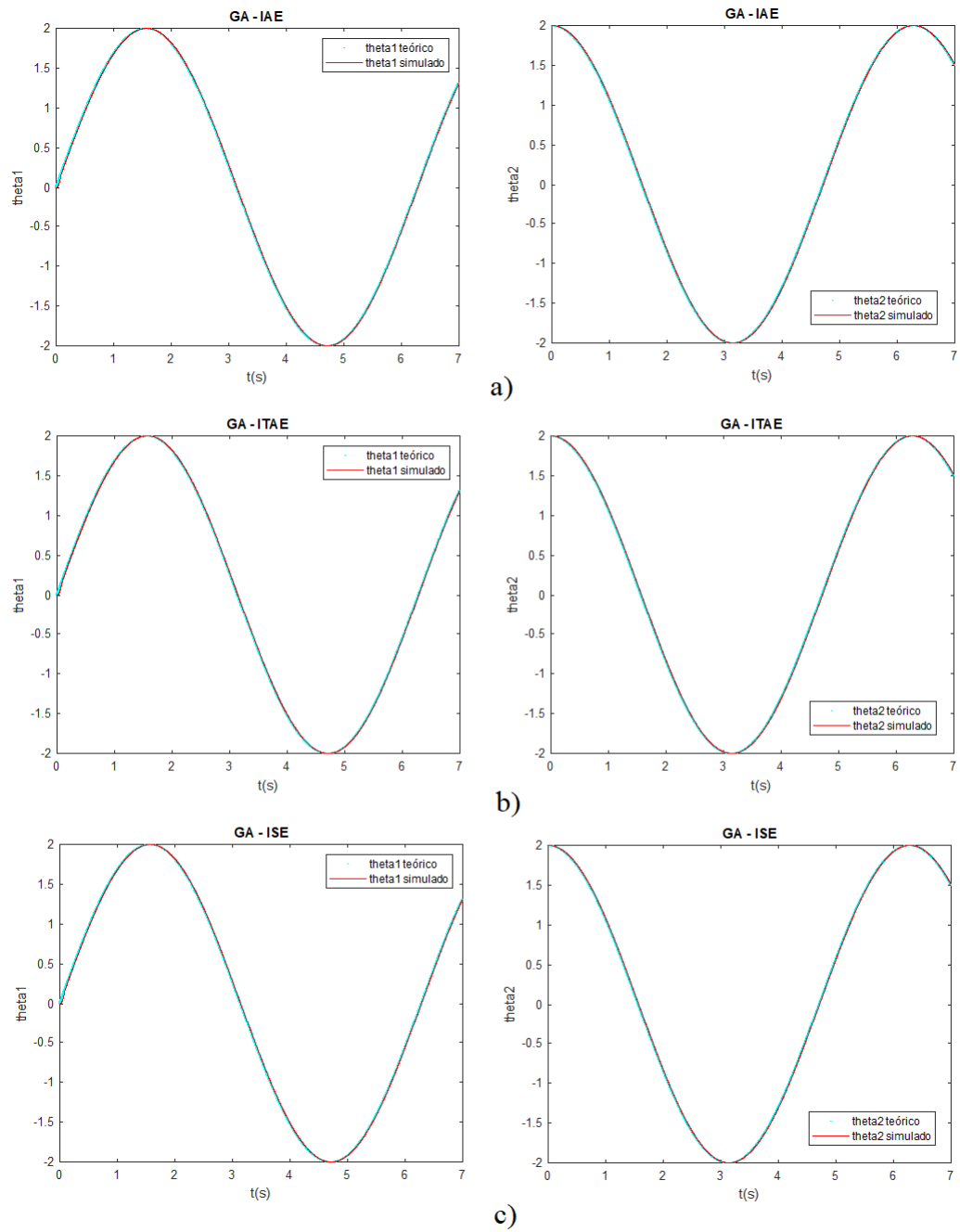


Figura 38 Comportamento circular descrito e esperado de cada junta: a) GA – IAE; b) GA – ITAE; c) GA- ISE

Por fim, é possível ver que os valores das funções objetivo, encontram-se com valores muito diminutos, o que vai de acordo com os gráficos das trajetórias percorridas, conforme mostra a Figura 39.

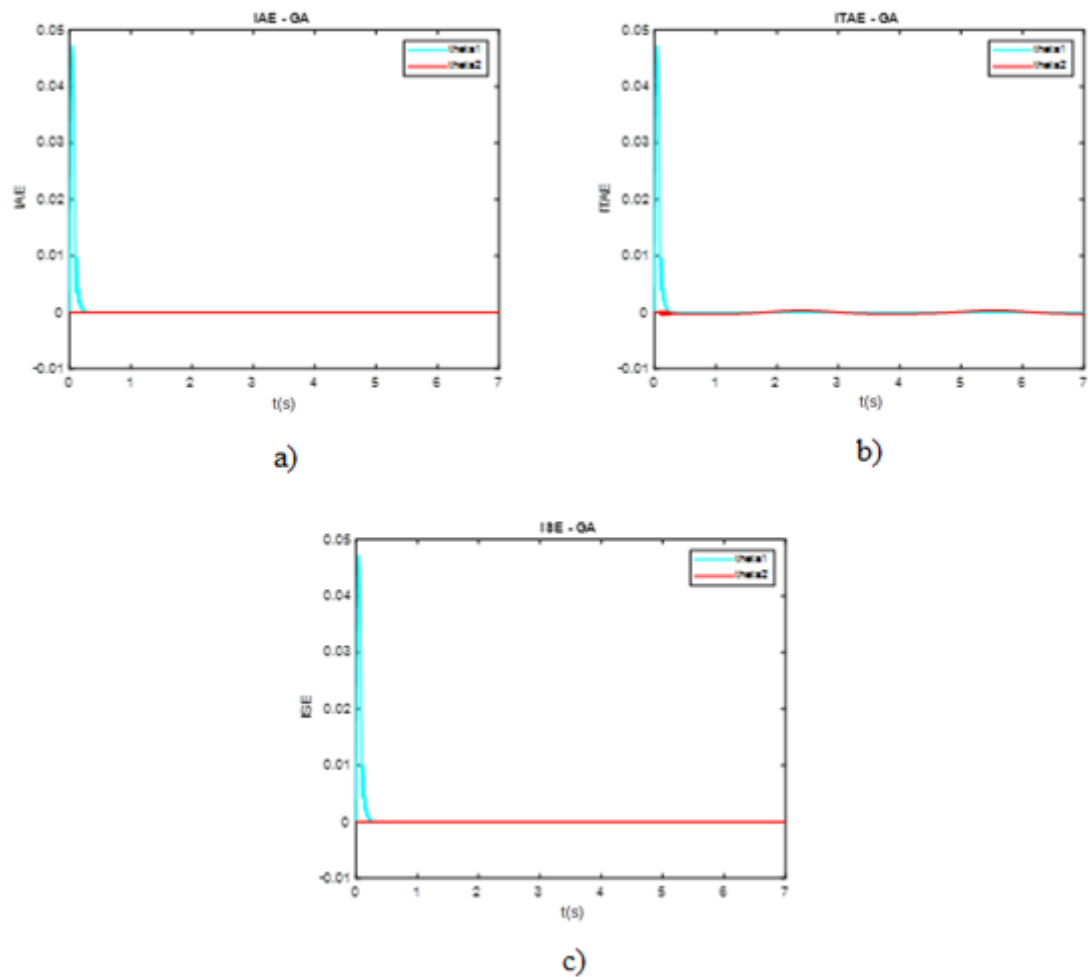


Figura 39 Funções objetivo: a) IAE - GA; b) ITAE - GA; c) ISE - GA

6.1.2. TRAJETÓRIA LINEAR

No que toca ao cenário estudado da trajetória linear, é possível rapidamente concluir que os valores dos ganhos vão de acordo com a generalidade das simulações. K_{p1} e K_{p2} continuam a ter valores altos, assim como K_{d1} e K_{d2} . A Tabela 6 apresenta os valores obtidos.

Tabela 6 Ganhos do Controlador GA - Recta

	Método	Ganhos do controlador 1			Ganhos do controlador 2		
		K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
GA	IAE	$2,508 \times 10^5$	$1,079 \times 10^1$	$2,107 \times 10^4$	$1,741 \times 10^5$	$1,859 \times 10^2$	$1,806 \times 10^3$
	ITAE	$4,647 \times 10^5$	$1,182 \times 10^1$	$3,950 \times 10^4$	$2,448 \times 10^5$	$1,232 \times 10^2$	$4,130 \times 10^3$
	ISE	$4,966 \times 10^5$	$8,178 \times 10^0$	$4,875 \times 10^4$	$4,866 \times 10^5$	$1,815 \times 10^2$	$4,889 \times 10^4$

Quanto ao desempenho da malha, com especial ênfase em cada junta, conclui-se que o melhor índice de desempenho foi o ISE, uma vez que apresenta o menor valor global registado na Tabela 7.

Tabela 7 Desempenho da Malha GA – Recta

	Método	Índice Junta 1	Índice Junta 2	Índice Global
GA	IAE	$3,100 \times 10^{-5}$	$2,443 \times 10^{-5}$	$5,543 \times 10^{-5}$
	ITAE	$2,458 \times 10^{-5}$	$2,555 \times 10^{-5}$	$5,013 \times 10^{-5}$
	ISE	$1,052 \times 10^{-10}$	$2,573 \times 10^{-11}$	$1,309 \times 10^{-10}$

No que toca às trajetórias descritas, estas mais uma vez apresentam grandes semelhanças com o que é pretendido, provando que o braço do robô consegue efetuar o movimento com uma alta precisão, conforme se verifica a partir da Figura 40.

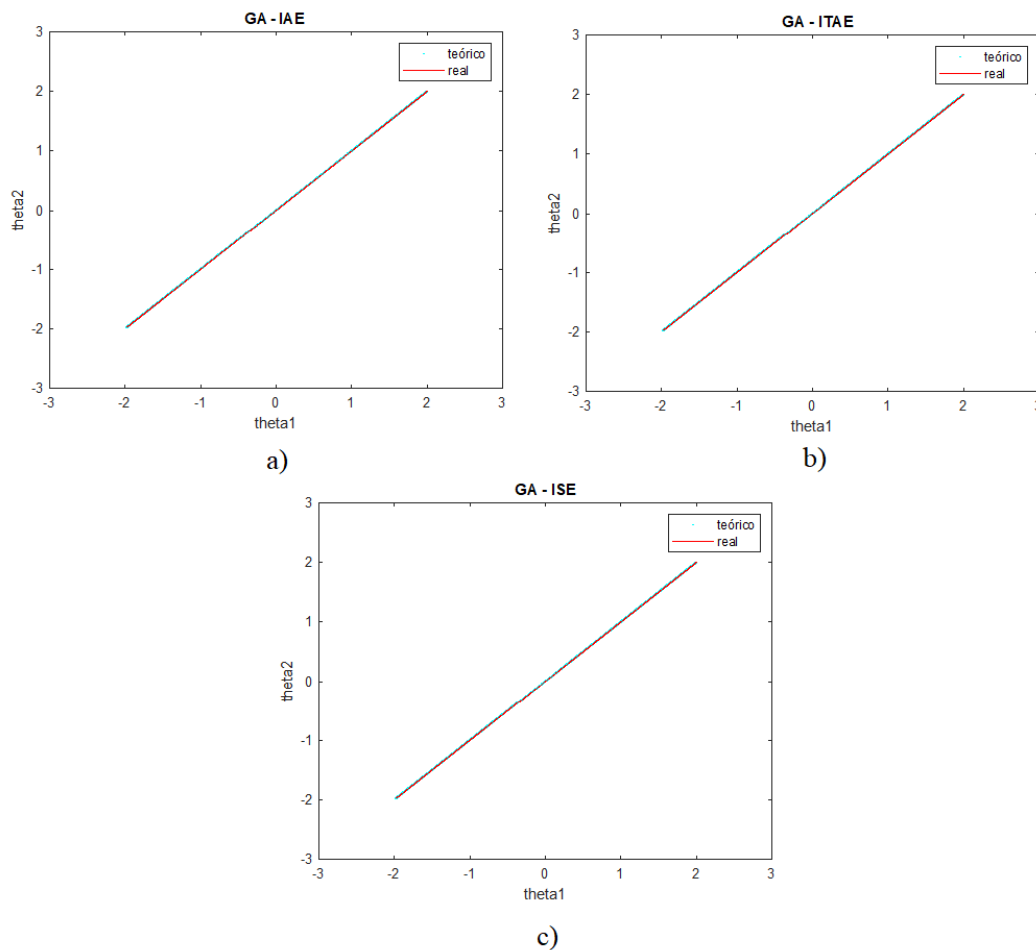


Figura 40 Trajetória linear descrita e esperada: a) GA – IAE; b) GA – ITAE; c) GA – ISE

Passando ao estudo individual de cada junta, de acordo com os três índices de desempenho simulados, a opinião mantém-se que o movimento efetuado é equivalente ao movimento desejado, pelo que há uma boa precisão por parte da manipulação do braço robótico, como é possível confirmar pela Figura 41.

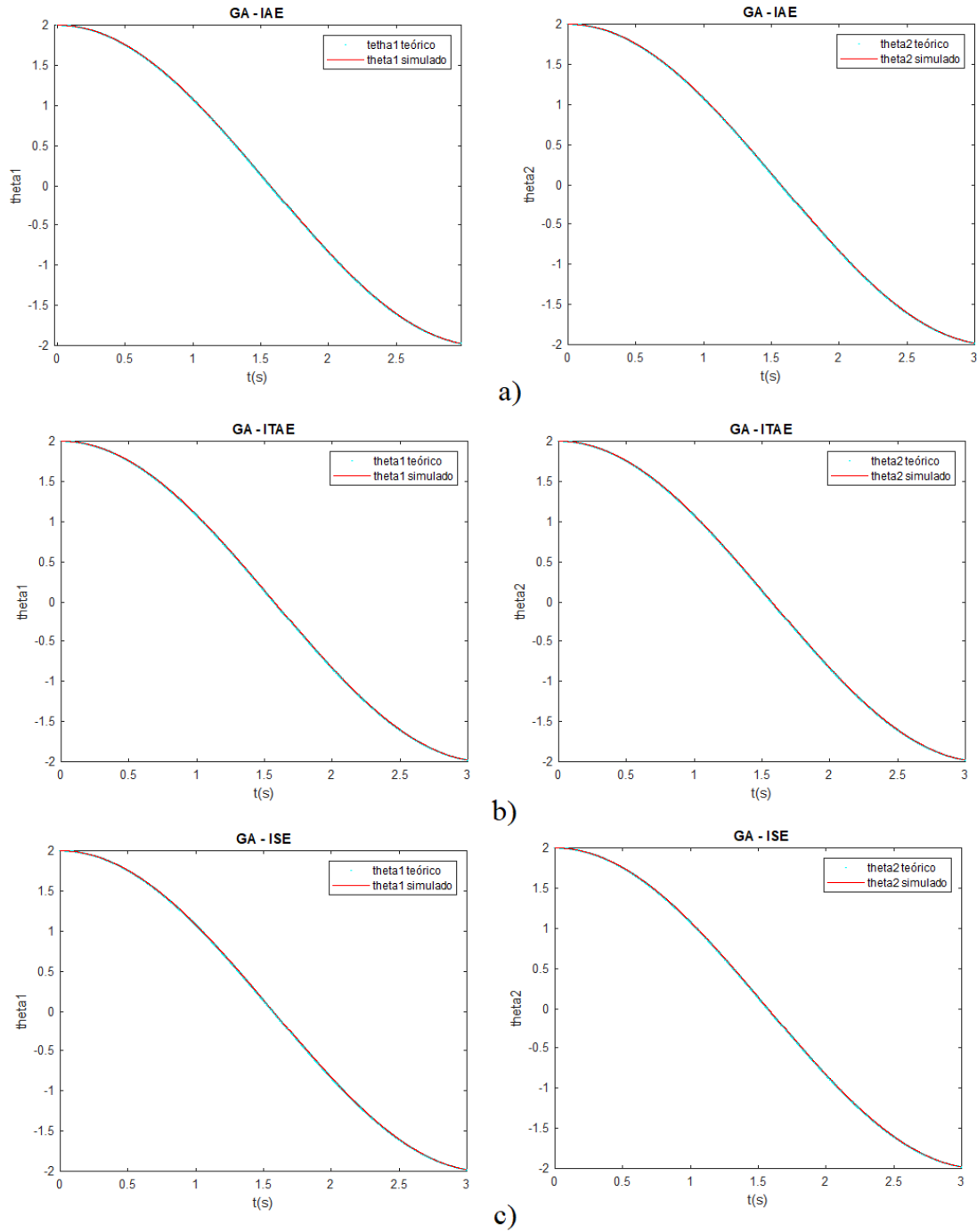


Figura 41 Comportamento linear descrito e esperado de cada junta: a) GA – IAE; b) GA – ITAE; c) GA- ISE

Por fim, é possível visualizar alguma discrepância nos valores das funções objetivo, no entanto estes são facilmente desprezáveis de acordo com a escala que foi utilizada. Esta escala tem puro propósito visual, de modo a perceber que, durante o movimento, ocorre

também uma variação de valores, em todos os cenários estudados, como é verificado pela Figura 42.

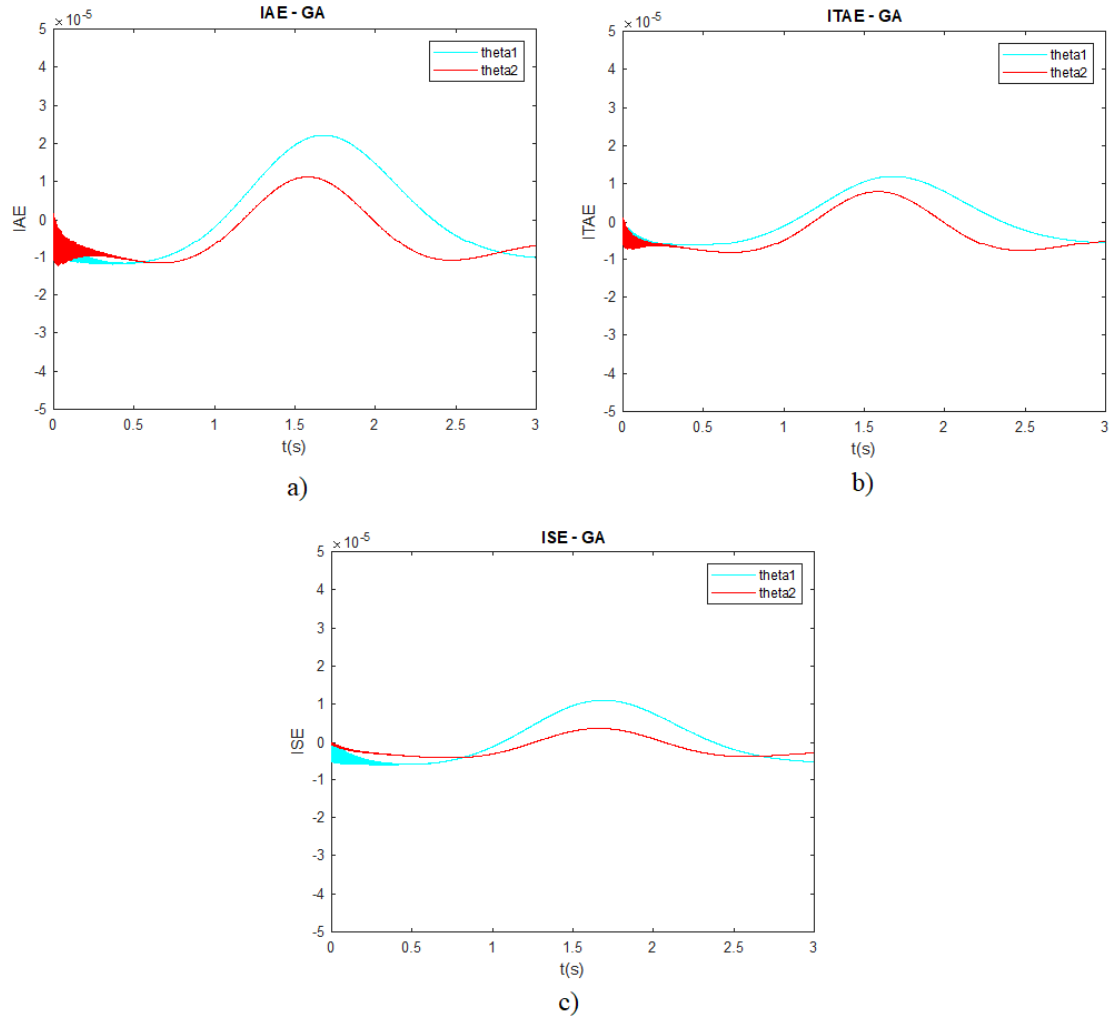


Figura 42 Funções objetivo: a) IAE - GA; b) ITAE - GA; c) ISE - GA

6.2. PARTICLE SWARM OPTIMIZATION (PSO)

6.2.1. TRAJETÓRIA CIRCULAR

No que toca ao estudo de cenário com a implementação do algoritmo PSO, o padrão de valores dos ganhos mantém-se, como se pode ver pela Tabela 8.

Tabela 8 Ganhos do Controlador PSO - Círculo

	Método	Ganhos do controlador 1			Ganhos do controlador 2		
		K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
PSO	IAE	$5,750 \times 10^5$	$1,475 \times 10^0$	$3,418 \times 10^4$	$2,641 \times 10^6$	$4,579 \times 10^1$	$4,999 \times 10^4$
	ITAE	$9,951 \times 10^5$	$8,870 \times 10^0$	$4,806 \times 10^4$	$1,874 \times 10^6$	$6,543 \times 10^1$	$2,960 \times 10^4$
	ISE	$7,525 \times 10^5$	$1,900 \times 10^2$	$4,089 \times 10^4$	$2,694 \times 10^6$	$1,172 \times 10^2$	$4,975 \times 10^4$

Os valores de desempenho apresentam resultados semelhantes e expectáveis, sendo que, neste caso em estudo, o valor mais optimizado no que toca ao índice global, passa pelo ITAE, como se pode verificar pela Tabela 9.

Tabela 9 Desempenho da Malha PSO – Círculo

	Método	Índice Junta1	Índice Junta2	Índice Global
PSO	IAE	$5,079 \times 10^{-3}$	$3,625 \times 10^{-6}$	$5,083 \times 10^{-3}$
	ITAE	$4,337 \times 10^{-3}$	$1,679 \times 10^{-5}$	$4,505 \times 10^{-4}$
	ISE	$1,399 \times 10^{-3}$	$3,635 \times 10^{-12}$	$1,399 \times 10^{-3}$

Assim sendo, de acordo com os valores da Tabela 8 e Tabela 9, facilmente se pode verificar que as trajetórias delineadas estão muito próximas das propostas. Assim, uma vez mais nota-se alguma facilidade no braço do robô em descrever a trajetória que lhe é pedida, como se vê pela Figura 43.

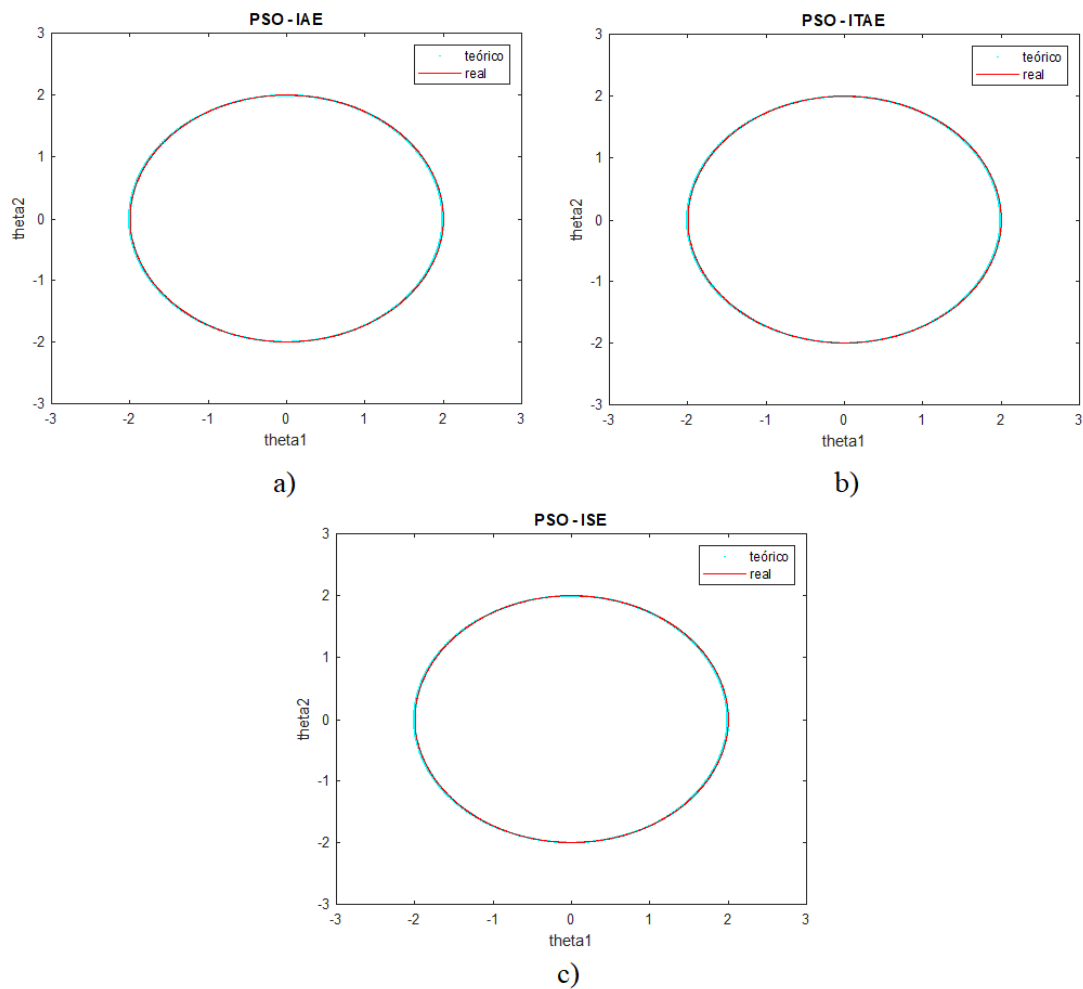


Figura 43 Trajetória circular descrita e esperada: a) PSO – IAE; b) PSO – ITAE; c) PSO – ISE

Relativamente ao estudo individual de cada junta, este obrigatoriamente também terá um bom desempenho, uma vez que a trajetória foi efetuada com sucesso. Assim, uma vez mais, não é possível notar grandes discrepâncias entre os comportamentos teóricos e simulados, como se confere pela Figura 44.

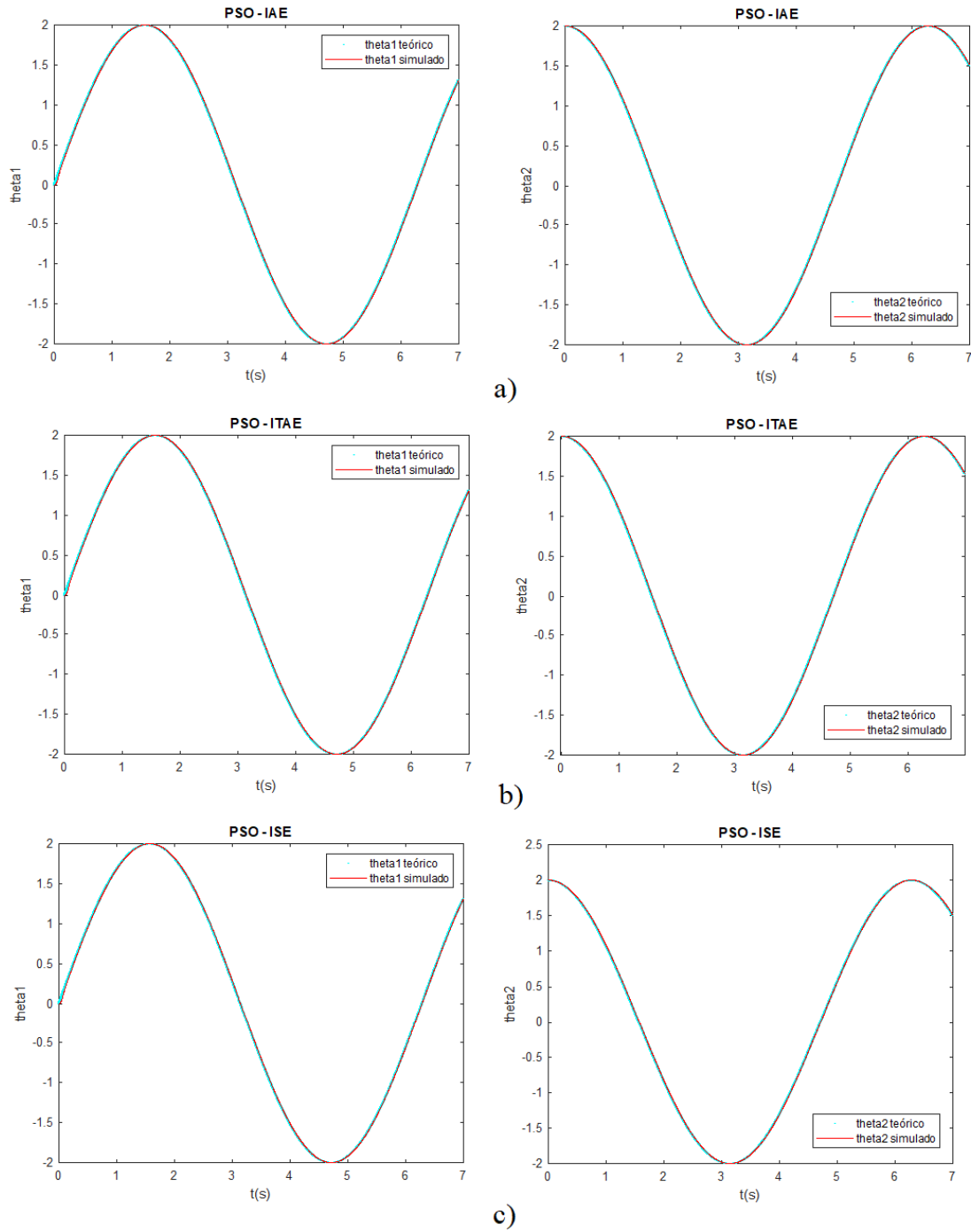


Figura 44 Comportamento descrito e esperado de cada junta: a) PSO – IAE; b) PSO – ITAE; c) PSO - ISE

No que toca às funções objetivo, todas elas têm um comportamento similar ao longo da escala de tempo, como se pode verificar pela Figura 45.

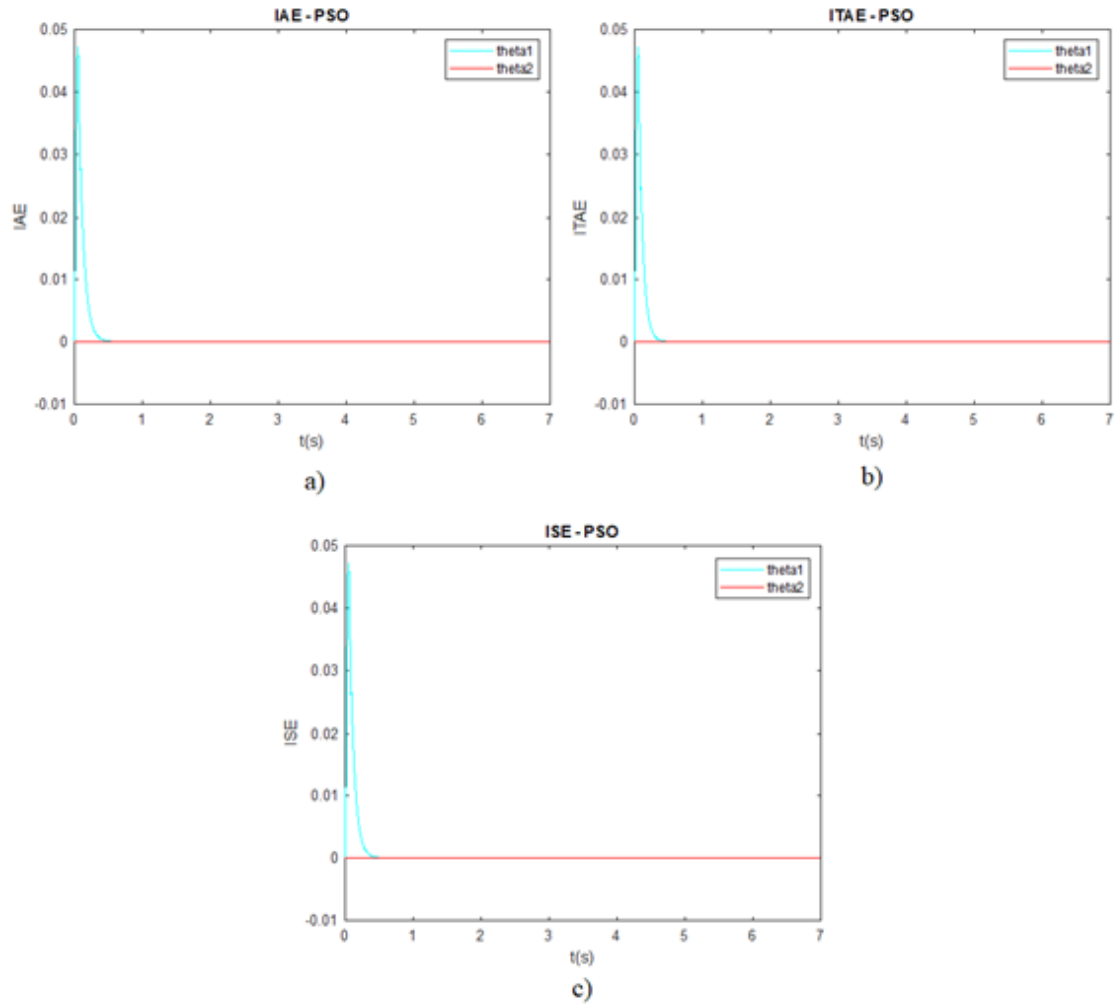


Figura 45 Funções objetivo: a) IAE - PSO; b) ITAE - PSO; c) ISE - PSO

6.2.2. TRAJETÓRIA LINEAR

Quanto ao estudo da trajetória linear, através do algoritmo PSO, os valores dos ganhos encontram-se bastante similares aos demais. De notar que utilizando o índice de desempenho ISE, este apresenta K_{i1} e K_{i2} nulos, como demonstrado na Tabela 10.

Tabela 10 Ganhos do Controlador PSO - Recta

	Método	Ganhos do controlador 1			Ganhos do controlador 2		
		K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
PSO	IAE	$5,000 \times 10^6$	$8,460 \times 10^1$	$5,000 \times 10^4$	$5,000 \times 10^6$	$1,900 \times 10^2$	$5,000 \times 10^4$
	ITAE	$5,000 \times 10^6$	$3,646 \times 10^1$	$5,000 \times 10^4$	$5,000 \times 10^6$	$1,900 \times 10^2$	$5,000 \times 10^4$
	ISE	$5,000 \times 10^6$	0	$5,000 \times 10^4$	$5,000 \times 10^6$	0	$4,995 \times 10^4$

No que toca à comparação entre os três índices de desempenho estudados, há uma diferença considerável entre o ISE e os demais, sendo este o mais otimizado por uma boa margem, como se verifica a partir da Tabela 11.

Tabela 11 Desempenho da Malha PSO – Recta

	Método	Índice Junta1	Índice Junta2	Índice Global
PSO	IAE	$1,657 \times 10^{-6}$	$8,520 \times 10^{-7}$	$2,509 \times 10^{-6}$
	ITAE	$2,401 \times 10^{-6}$	$1,251 \times 10^{-6}$	$3,652 \times 10^{-6}$
	ISE	$1,200 \times 10^{-12}$	$2,746 \times 10^{-13}$	$1,475 \times 10^{-12}$

Assim, as trajetórias nos três cenários foram delineadas com sucesso a partir do braço do robô, como se pode conferir pela Figura 46. As rectas teóricas e reais estão praticamente sobrepostas uma sobre a outra.

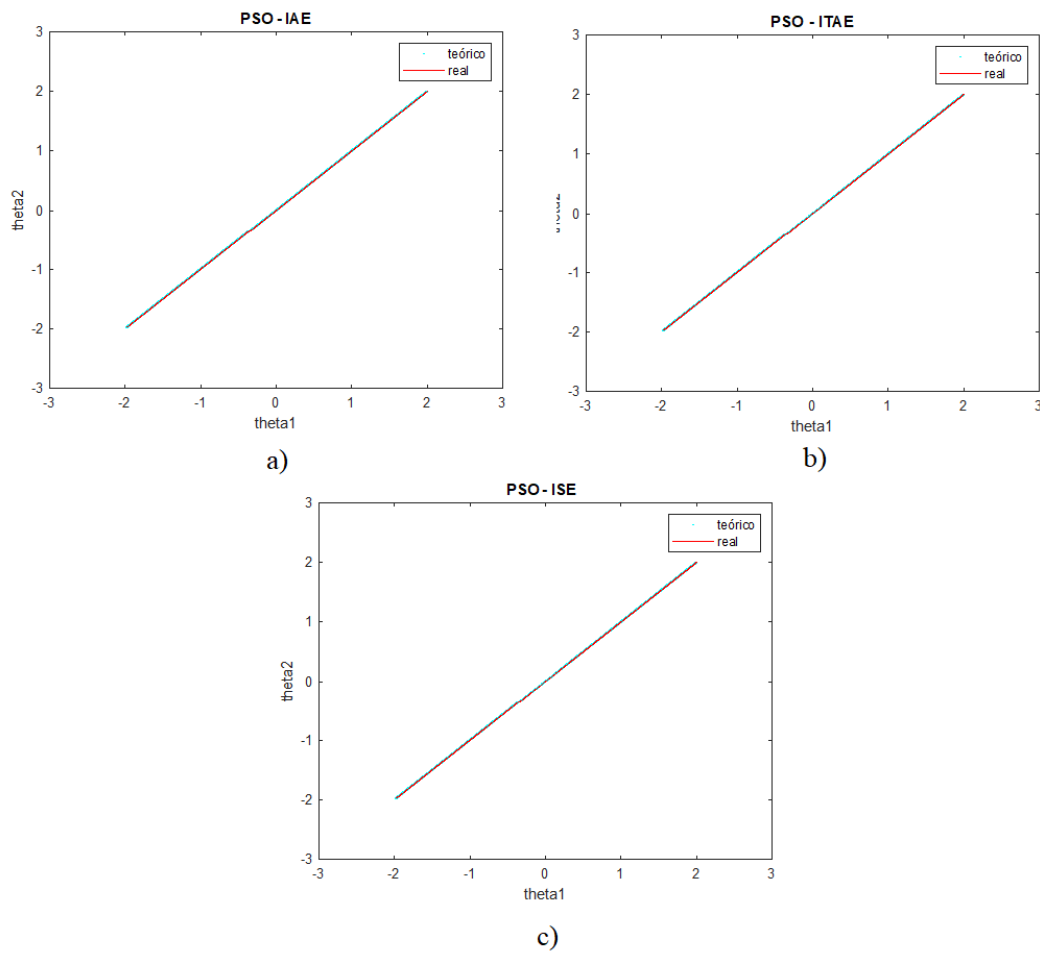


Figura 46 Trajetória linear descrita e esperada: a) PSO – IAE; b) PSO – ITAE; c) PSO – ISE

Quanto ao comportamento de cada uma das juntas, é possível concluir que o intervalo de valores encontra-se enquadrado e ajustado com o pretendido, pelo que é possível replicar e simular o movimento teórico, como é demonstrado a partir da Figura 47.

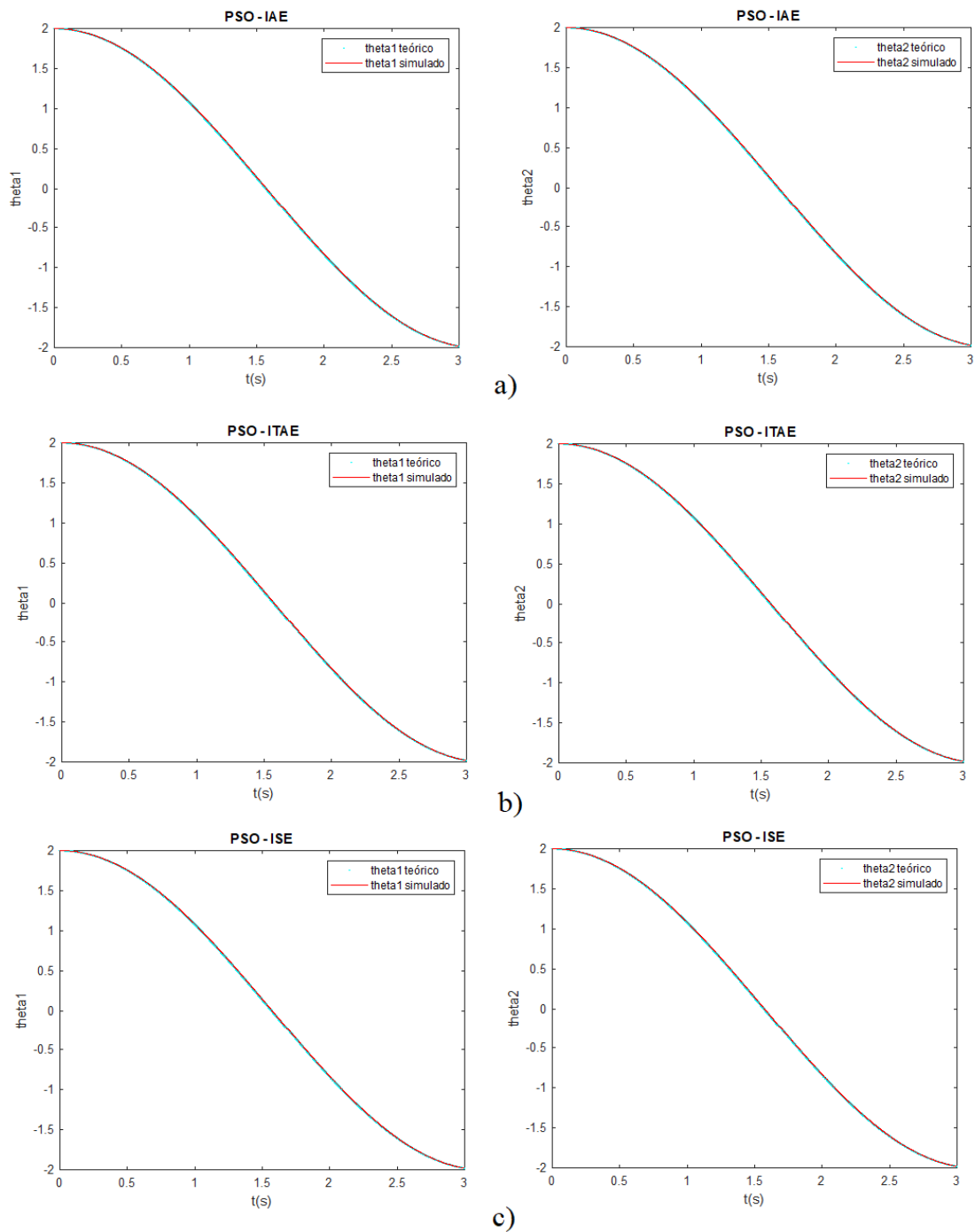


Figura 47 Comportamento descrito e esperado de cada junta: a) PSO – IAE; b) PSO – ITAE; c) PSO – ISE

Por fim, as funções objetivo encontram-se com valores muito próximos do zero, o que se tende a retirar ilações positivas, pois não há grandes discrepâncias ao longo do tempo da trajetória, como se constata pela Figura 48.

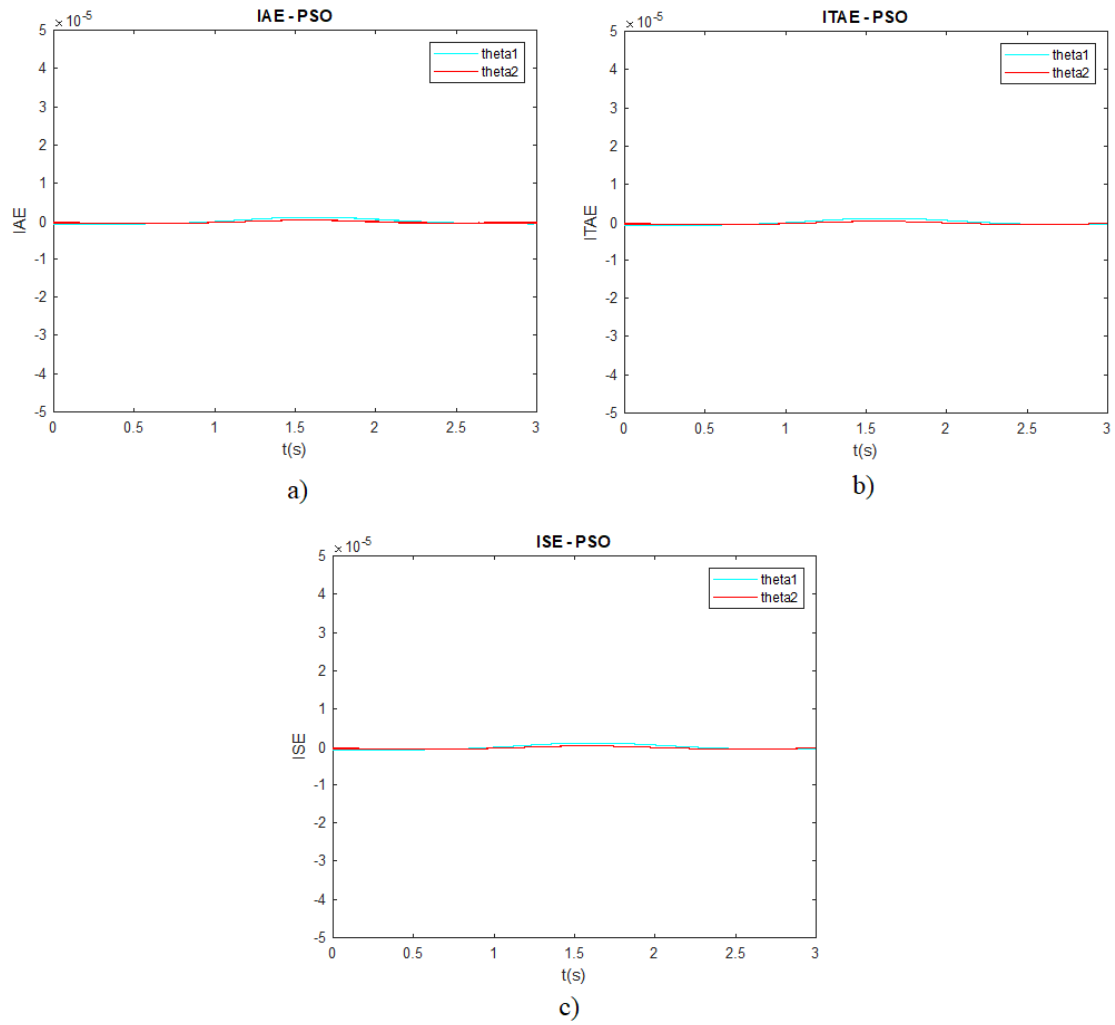


Figura 48 Funções objetivo: a) IAE - PSO; b) ITAE - PSO; c) ISE - PSO

6.3. GLOBAL PATTERN SEARCH (GPS)

6.3.1. TRAJETÓRIA CIRCULAR

Relativamente ao estudo dos ganhos de ambos os controladores existentes no braço robótico, conclui-se na sua generalidade que os valores se encontram num intervalo adequado, em termo de comparação com todos os cenários anteriores.

No entanto, é preciso referir que o índice de desempenho IAE apresenta valores de K_{i1} e K_{i2} nulos, como demonstrado pela Tabela 12.

Tabela 12 Ganhos do Controlador GPS - Círculo

	Método	Ganhos do controlador 1			Ganhos do controlador 2		
		K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
GPS	IAE	$9,741 \times 10^3$	0	$5,500 \times 10^2$	$6,054 \times 10^4$	0	$1,152 \times 10^3$
	ITAE	$3,532 \times 10^5$	0	$1,994 \times 10^4$	$3,274 \times 10^4$	$1,280 \times 10^2$	$5,120 \times 10^2$
	ISE	$4,815 \times 10^3$	$5,570 \times 10^0$	$1,780 \times 10^2$	$4,100 \times 10^1$	$8,038 \times 10^1$	$1,000 \times 10^0$

No que toca à tabela de desempenho dos vários índices estudados, rapidamente conclui-se que todos eles apresentam valores muito similares e que a diferença é muito pouco relevante. O índice global de desempenho com melhores resultados, passa pelo ITAE, como se vê na Tabela 13.

Tabela 13 Desempenho da Malha GPS – Círculo

	Método	Índice	Índice	Índice
		Junta1	Junta2	Global
GPS	IAE	$6,077 \times 10^{-3}$	$1,525 \times 10^{-4}$	$6,230 \times 10^{-3}$
	ITAE	$5,899 \times 10^{-4}$	$9,575 \times 10^{-4}$	$1,574 \times 10^{-3}$
	ISE	$1,273 \times 10^{-4}$	$4,261 \times 10^{-3}$	$4,388 \times 10^{-3}$

Assim, uma vez que os valores da Tabela 12 e da Tabela 13 encontram-se dentro do padrão normal de estudo, é também possível concluir, pela Figura 49, que as três trajetórias são bem efetuadas pelo braço robótico.

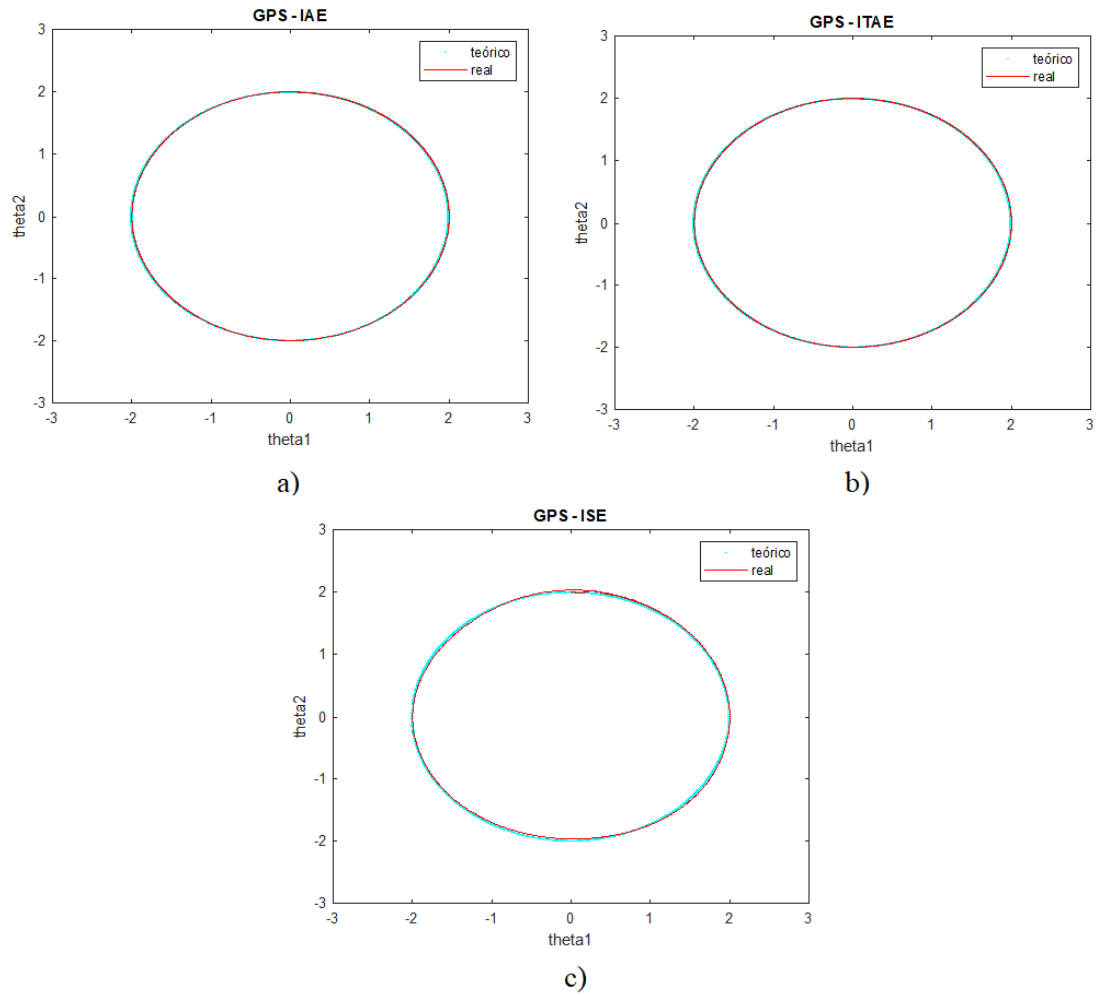


Figura 49 Trajetória circular descrita e esperada: a) GPS – IAE; b) GPS – ITAE; c) GPS – ISE

Complementando as trajetórias, o comportamento de cada junta estudada está igualmente bem configurado, de modo que qualquer discrepância seja praticamente invisível ao olho humano, como demonstrado na Figura 50.

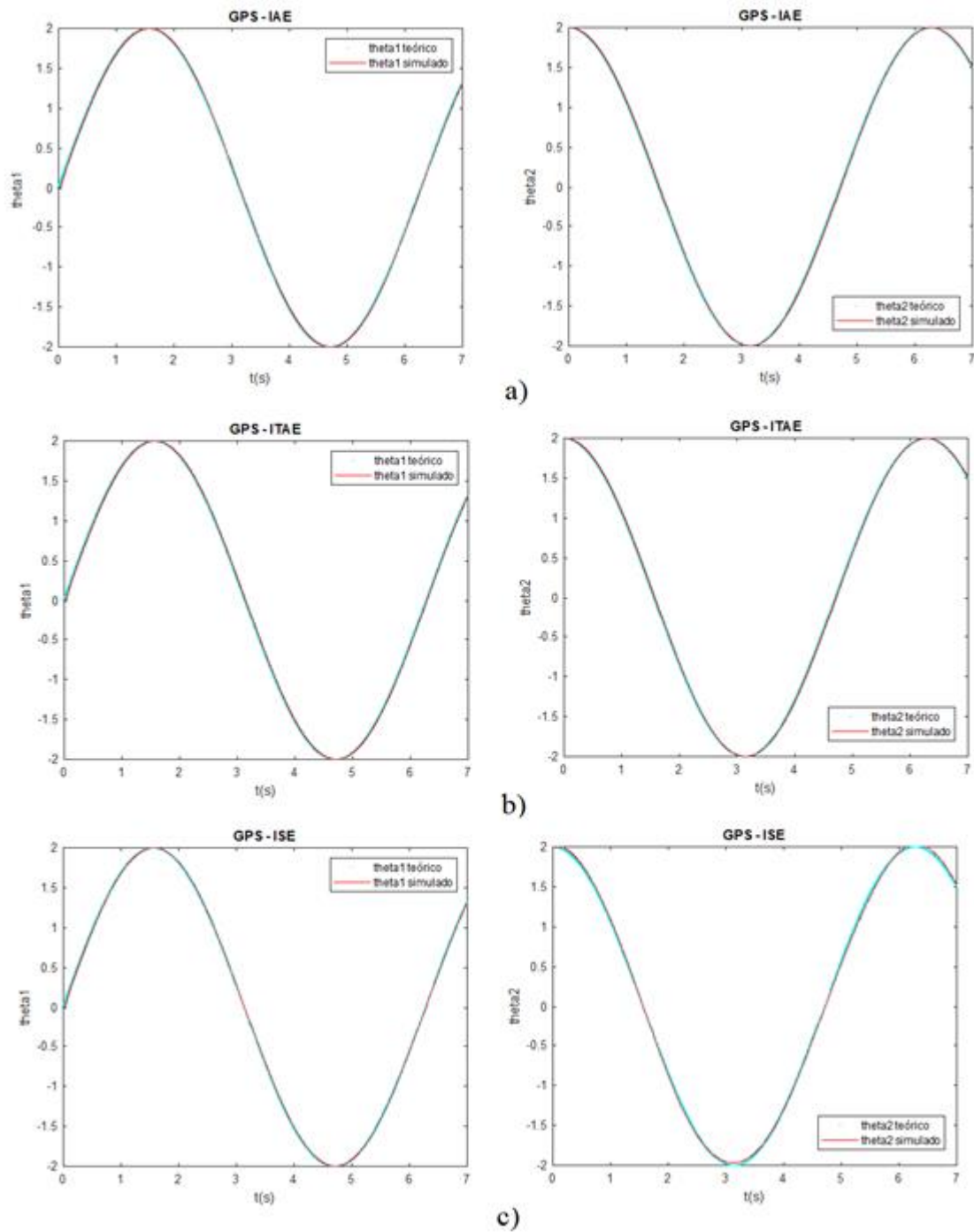


Figura 50 Comportamento descrito e esperado de cada junta: a) GPS – IAE; b) GPS – ITAE; c) GPS – ISE

No que toca às funções objetivo da Figura 51, os índices de desempenho IAE e ITAE apresentam comportamentos já expectados e vistos em cenários anteriores. No entanto, o índice ISE apresenta uma maior oscilação evidenciando a existência de um maior erro.

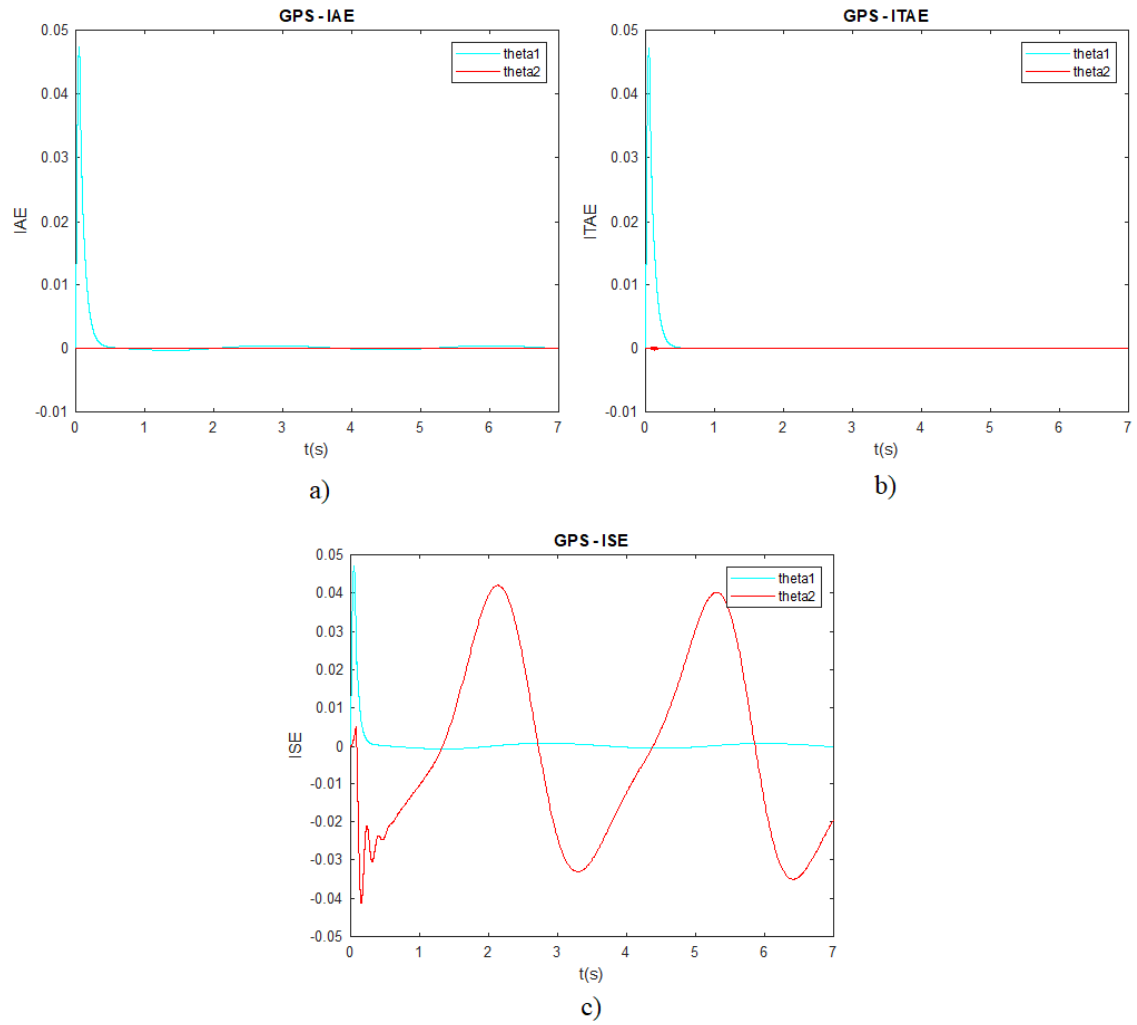


Figura 51 Funções objetivo: a) IAE - GPS; b) ITAE - GPS; c) ISE - GPS

6.3.2. TRAJETÓRIA LINEAR

É possível concluir que utilizando o índice de desempenho ISE, este apresenta K_{i1} e K_{i2} nulos, como demonstrado na Tabela 14.

No entanto, na sua generalidade, é possível concluir que os valores dos ganhos nos respectivos índices de desempenho, vão de acordo com o padrão de valores vistos em todos os outros cenários que já foram estudados.

Tabela 14 Ganhos do Controlador GPS - Recta

	Método	Ganhos do controlador 1			Ganhos do controlador 2		
		K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
GPS	IAE	$4,999 \times 10^6$	0	$4,996 \times 10^4$	$4,999 \times 10^6$	$1,280 \times 10^2$	$4,992 \times 10^4$
	ITAE	$4,997 \times 10^6$	$1,000 \times 10^0$	$4,915 \times 10^4$	$4,981 \times 10^6$	$7,750 \times 10^0$	$4,915 \times 10^4$
	ISE	$5,000 \times 10^6$	0	$4,978 \times 10^4$	$5,000 \times 10^6$	0	$4,992 \times 10^4$

Olhando para os diversos índices de desempenho da malha, é possível concluir que todos os valores são bastante baixos, indicando uma taxa de optimização do sistema consideravelmente alta. O índice com valor menor e, obrigatoriamente, mais otimizado, é o ISE, que apresenta um valor bem mais baixo que os outros dois, neste cenário em questão.

Tabela 15 Desempenho da Malha GPS – Recta

	Método	Índice Junta1	Índice Junta2	Índice Global
GPS	IAE	$1,660 \times 10^{-6}$	$8,659 \times 10^{-7}$	$2,526 \times 10^{-6}$
	ITAE	$2,405 \times 10^{-6}$	$1,318 \times 10^{-6}$	$3,723 \times 10^{-6}$
	ISE	$1,214 \times 10^{-12}$	$3,479 \times 10^{-13}$	$1,562 \times 10^{-12}$

Através da Figura 52 é possível ver e concluir que todas as trajetórias foram delineadas com uma precisão alta e muito próximas do pretendido. Assim, o comportamento do braço robótico e a sua configuração, estão acertadas para o tipo de movimento que é pretendido.

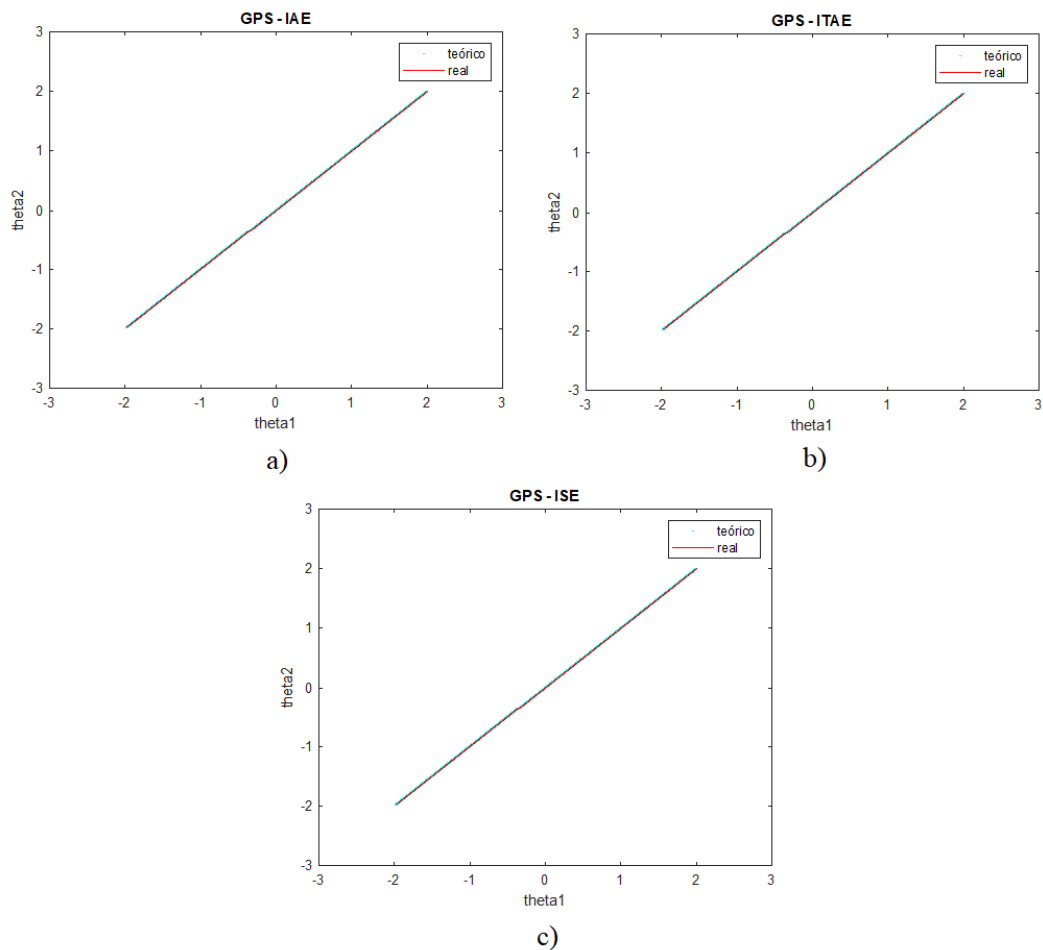


Figura 52 Trajetória linear descrita e esperada: a) GPS – IAE; b) GPS – ITAE; c) GPS – ISE

De modo a complementar o que já foi dito a partir da Figura 52, também o comportamento das juntas vai de acordo com o pretendido, confirmando o bom desempenho do robô e a facilidade em efetuar a trajetória que é previamente pedida, como se pode verificar pela Figura 53.

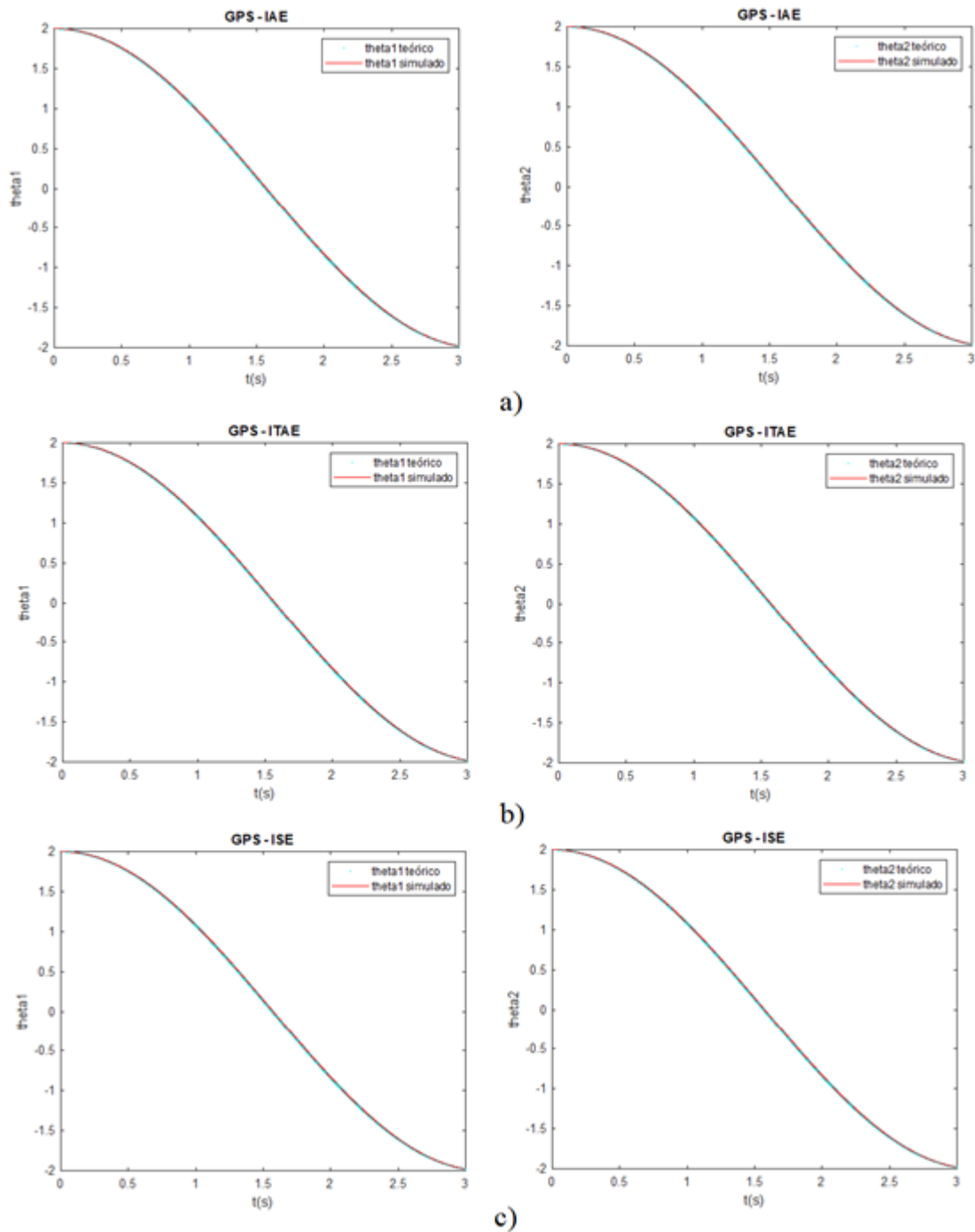


Figura 53 Comportamento descrito e esperado de cada junta: a) GPS – IAE; b) GPS – ITAE; c) GPS - ISE

Por fim, os três índices de desempenho da Figura 54, vão de acordo com os valores mostrados na Tabela 15, permitindo concluir e cimentar o bom desempenho e otimização do braço robótico ao efetuar as trajetórias descritas na Figura 52.

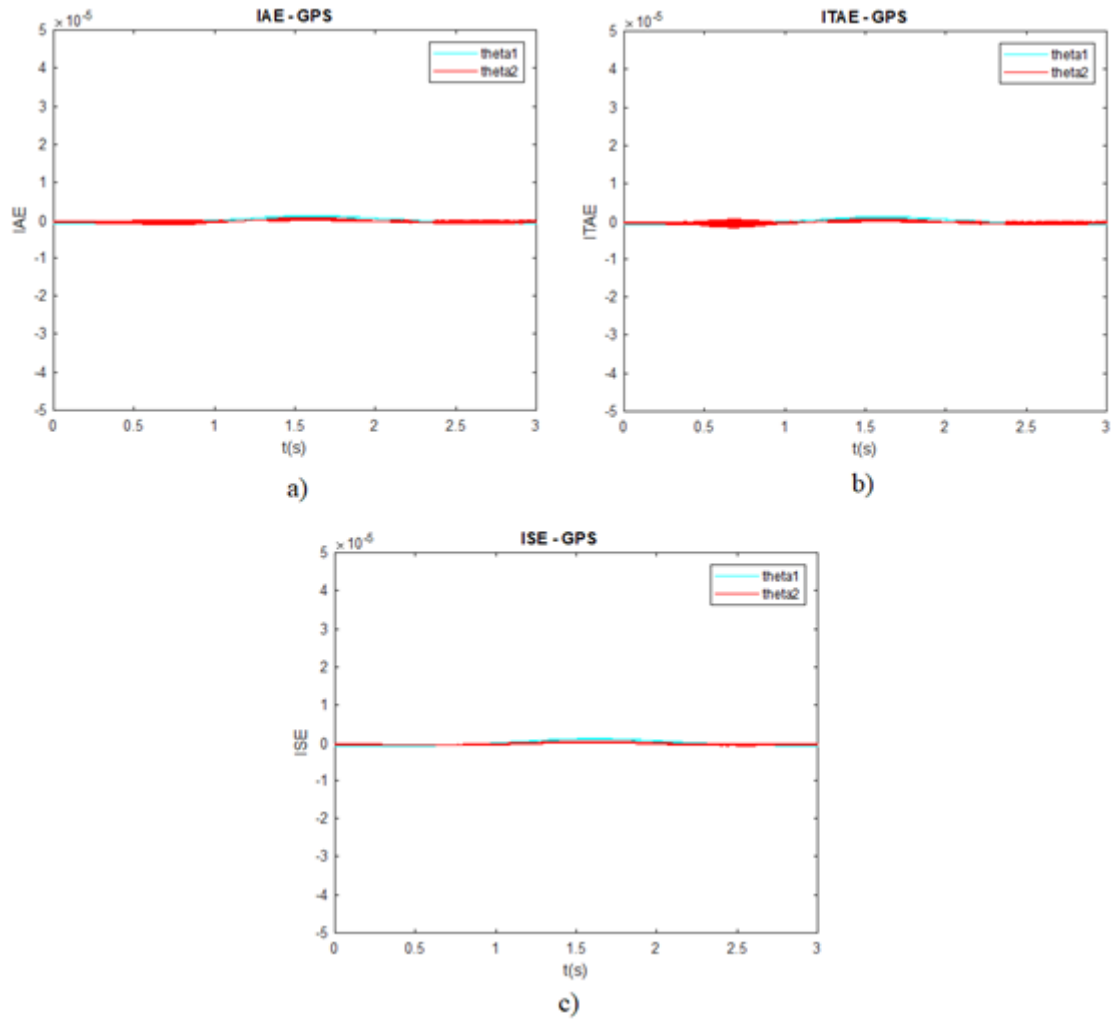


Figura 54 Funções objetivo: a) IAE - GPS; b) ITAE - GPS; c) ISE - GPS

6.4. ANÁLISE COMPARATIVA

Após todas as simulações estarem de acordo com ambas as trajetórias e os três índices de desempenho utilizados, considera-se crucial calcular o erro para cada simulação, de modo a retirar conclusões sobre o impacto dos algoritmos no braço robótico.

Assim, aplicando a cinemática direta, é possível calcular a posição real do braço robótico e comparar com a posição desejada a cada instante de tempo. O erro é calculado a partir da seguinte expressão:

$$Erro = \sqrt{\sum_{i=1}^N \Delta x_i^2 + \Delta y_i^2} \quad (38)$$

$$\Delta x_i = x_{simulado_i} - x_{teórico_i} \quad (39)$$

$$\Delta y_i = y_{simulado_i} - y_{teórico_i} \quad (40)$$

Onde N é o número de amostras correspondente ao tempo de simulação.

Assim, a partir da Tabela 16, é possível retirar múltiplas conclusões. A primeira consiste na discrepância entre os valores do erro de ambas trajetórias, sendo que estes são consideravelmente maiores quando o manipulador percorre a trajetória circular, o que seria de esperar, uma vez que se trata de uma trajetória mais complexa.

Neste cenário, é possível também ver que o menor erro é representado no algoritmo GPS com o índice IAE, sendo que o algoritmo PSO é o que apresenta um valor de erro maior.

Já no que toca à trajetória linear, todos os valores são bastante constantes e similares, havendo ligeiras variações, provando que se trata de uma trajetória simples para o braço do robô. O menor valor do erro é representado a partir do algoritmo PSO com o índice ITAE, sendo que este algoritmo apresenta os valores mais otimizados para qualquer um dos três índices de desempenho testados.

Tabela 16 Erro do sistema

Erro	Trajetória Circular			Trajetória Linear		
	IAE	ITAE	ISE	IAE	ITAE	ISE
GA	$8,370 \times 10^{-2}$	$4,250 \times 10^{-2}$	$8,260 \times 10^{-2}$	$7,500 \times 10^{-5}$	$5,428 \times 10^{-5}$	$5,184 \times 10^{-5}$
PSO	$1,568 \times 10^{-1}$	$1,329 \times 10^{-1}$	$1,434 \times 10^{-1}$	$3,108 \times 10^{-5}$	$1,738 \times 10^{-5}$	$1,700 \times 10^{-5}$
GPS	$8,070 \times 10^{-2}$	$6,850 \times 10^{-2}$	$8,410 \times 10^{-2}$	$6,903 \times 10^{-5}$	$7,602 \times 10^{-5}$	$2,565 \times 10^{-5}$

7. CONCLUSÕES

Em linhas gerais pode-se concluir que foi implementado um braço robótico com dois graus de liberdade, capaz de percorrer trajetórias previamente delineadas com sucesso.

Numa primeira instância, a implementação do modelo robótico em ambiente Simulink, pertencente à ferramenta MatLab, tratou-se de um desafio com bastantes obstáculos, uma vez que se recorreu a várias tentativas de implementação das equações da cinemática direta e inversa do robô sem sucesso.

O objetivo do controlador PID passa por minimizar a diferença entre a posição angular desejada e a posição angular real da junta, fornecendo o binário necessário de modo a mover as juntas para as posições desejadas. Assim, foi utilizada a *toolbox* de optimização para algoritmos genéticos, que, uma vez que otimizava seis variáveis de controlo, fazia com que o tempo de simulação de optimização demorasse várias horas. No entanto, os resultados obtidos foram satisfatórios, assim como as trajetórias delineadas pelo braço robótico, uma vez que estão bastante próximas das trajetórias pretendidas.

7.1. TRABALHO FUTURO

No que toca a futuros melhoramentos do trabalho, realçam-se dois pontos – integração de motores DC a alimentar o robô e implementação de um ambiente com obstáculos.

Ambos trazem mais realidade ao ambiente onde o braço robótico interage e movimenta-se. O primeiro iria servir como alimentação a cada uma das juntas do robô, tal como no funcionamento dos robôs industriais nos dias de hoje. Cada motor seria limitado a um binário máximo, de acordo já com o limitador que foi implementado neste trabalho, no entanto seria visto como uma melhoria, pois o modelo acabaria por ser mais completo.

Já o ambiente com obstáculos, iria trazer uma maior dificuldade às trajetórias que o braço robótico percorre e, por sua vez, obrigaria a implementar um sistema de sensorização para evitar qualquer tipo de colisões com os vários obstáculos. Tal como escrito acima, este ambiente traria mais realidade e também iria obrigar a um algoritmo mais complexo, de modo que o manipulador robótico consiga delinear a trajetória a percorrer, mas com os devidos desvios sobre os obstáculos que se encontram no caminho.

Referências Documentais

- [1] “An Introduction to Genetic Algorithms | The MIT Press.” <https://mitpress.mit.edu/books/introduction-genetic-algorithms> (accessed Nov. 10, 2020).
- [2] “Rodrigo Evangelista de Castro, Alvaro L. G. A. Coutinho, Helio J. C. Barbosa, *Optimização de Estruturas com Multi-objetivos Via Algoritmos Genéticos de Pareto* (tese de doutoramento), (2001), COOPE / UFRJ.” <http://www.coc.ufrj.br/pt/documents2/doutorado/2001-2/686-rodrigo-evangelista-de-castro-doutorado/file> (accessed Nov. 10, 2020).
- [3] “Banzhaf, Wolfgang et. al., ‘Genetic Programming (an introduction)’, Morgan Kaufmann Publishers, Inc, 1998.” <ftp://ftp.micronet-rostov.ru/linux-support/books/computer-science/Artificial-Intelligence/Evolutionary-computation/Genetic-Programming-An-Introduction-On-the-Automatic-Evolution-of-Computer-Programs-and-its-Applications-Morgan-Kaufmann.pdf> (accessed Nov. 10, 2020).
- [4] J. McCall, “Genetic algorithms for modelling and optimisation,” *J. Comput. Appl. Math.*, vol. 184, no. 1, pp. 205–222, Dec. 2005, doi: 10.1016/j.cam.2004.07.034.
- [5] “J. Welles Wilder, *New concepts in technical trading systems*, Trend Research, 1978.” <https://books.mec.biz/tmp/books/218XOTBWY3FEW2CT3EVR.PDF> (accessed Nov. 10, 2020).
- [6] D. Blanchard, “WCCI’94: It’s a fuzzy world out there,” *Expert Syst.*, vol. 11, no. 3, pp. 179–181, Aug. 1994, doi: 10.1111/j.1468-0394.1994.tb00325.x.
- [7] A. Petrovski and J. McCall, “Multi-objective optimisation of cancer chemotherapy using evolutionary algorithms,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2001, vol. 1993, pp. 531–545, doi: 10.1007/3-540-44719-9_37.
- [8] A. Petrovski, “An application of genetic algorithms to chemotherapy treatment,” Dec. 1998. Accessed: Nov. 10, 2020. [Online]. Available: <https://rgu->

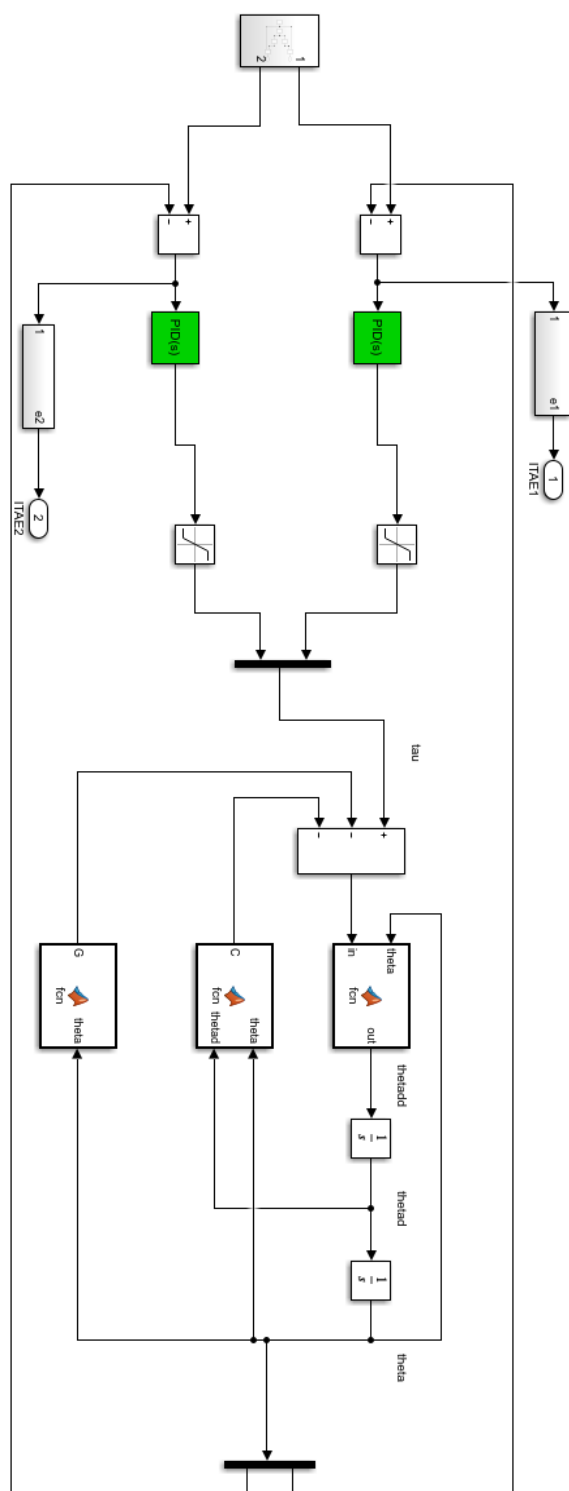
repository.worktribe.com/output/248737/an-application-of-genetic-algorithms-to-chemotherapy-treatment.

- [9] “Ant colony optimization.” https://www.slideshare.net/MeenakshiDevi/ant-colony-optimization-11696728?next_slideshow=1 (accessed Nov. 10, 2020).
- [10] C. Blum, “Ant colony optimization: Introduction and recent trends,” *Phys. Life Rev.*, vol. 2, pp. 353–373, 2005, doi: 10.1016/j.plrev.2005.10.001.
- [11] “An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report - TR06 | Request PDF.” https://www.researchgate.net/publication/255638348_An_Idea_Based_on_Honey_Bee_Swarm_for_Numerical_Optimization_Technical_Report_-_TR06 (accessed Nov. 10, 2020).
- [12] B. Akay and D. Karaboga, “Artificial bee colony algorithm for large-scale problems and engineering design optimization,” *J. Intell. Manuf.*, vol. 23, no. 4, pp. 1001–1014, Aug. 2012, doi: 10.1007/s10845-010-0393-4.
- [13] X.-S. Yang and S. Deb, “Multiobjective cuckoo search for design optimization,” *Comput. & Oper. Res.*, vol. 40, no. 6, pp. 1616–1624, Accessed: Nov. 10, 2020. [Online]. Available: https://www.academia.edu/7395352/Multiobjective_cuckoo_search_for_design_optimization.
- [14] X.-S. Yang and S. Deb, “Cuckoo Search: Recent Advances and Applications,” *Neural Comput. Appl.*, vol. 24, no. 1, pp. 169–174, Aug. 2014, doi: 10.1007/s00521-013-1367-1.
- [15] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proc. ICNN’95 - Int. Conf. Neural Networks*, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
- [16] J. Kennedy and Morgan Kaufmann Swarm intelligence, 2001[S.1]. .
- [17] R. J. Crouser *et al.*, “Global Pattern Search at Scale,” *2015 IEEE Int. Symp. Technol. Homel. Secur. HST 2015*, Aug. 2015, doi: 10.1109/THS.2015.7225293.
- [18] I. Popa, “Robot’s Apperance.” p. 47, 2016.

- [19] “History of Robotics: Timeline.” Accessed: Nov. 10, 2020. [Online]. Available: www.robotshop.ca.
- [20] A. Gasparetto and L. Scalera, “A Brief History of Industrial Robotics in the 20th Century,” *Adv. Hist. Stud.*, vol. 08, no. 01, pp. 24–35, 2019, doi: 10.4236/ahs.2019.81002.
- [21] “Robotics in Manufacturing: The History Outline.” <https://cerasis.com/robotics-in-manufacturing/> (accessed Nov. 10, 2020).
- [22] “Automated & Robotic Systems | Cisco-Eagle | Virginia | North Carolina.” <https://www.cisco-eagle.com/virginia/robotics-automation-va> (accessed Nov. 10, 2020).
- [23] J. Shah, S. S. Rattan, and B. C. Nakra, “DYNAMIC ANALYSIS OF TWO LINK ROBOT MANIPULATOR FOR CONTROL DESIGN USING COMPUTED TORQUE CONTROL,” 2015, Accessed: Nov. 10, 2020. [Online]
- [24] M. I. Ribeiro and P. Lima, “KINEMATICS MODELS OF KINEMATICS MODELS OF MOBILE ROBOTS MOBILE ROBOTS MOBILE ROBOTICS course,” 2002.
- [25] J. S. Sequeira, “Introdução à Robótica caps 1 a 6.” <http://docplayer.com.br/56204532-Robotica-joao-silva-sequeira.html> (accessed Jul. 18, 2021)
- [26] N. M. Ghaleb and A. A. Aly, “Modeling and Control of 2-DOF Robot Arm,” 2018.
- [27] A. De Luca Robotics, “Robotics 1 Direct kinematics.” <https://fdocuments.in/reader/full/robotics-1-direct-kinematics> (accessed Jun. 28, 2021)
- [28] N. A. Elkhateeb and R. I. Badr, “Novel PID Tracking Controller for 2DOF Robotic Manipulator System Based on Artificial Bee Colony Algorithm,” *Electr. Control Commun. Eng.*, vol. 13, no. 1, pp. 55–62, Jan. 2018, doi: 10.1515/ecce-2017-0008.
- [29] W. Liuping, “Basics of PID Control,” in *Practical PID Control*, 2006, pp. 1–18.
- [30] J. P. P., J. P. Perez, R. Soto, A. Flores, F. Rodriguez, and J. L. Meza, “Trajectory Tracking Error Using PID Control Law for Two-Link Robot Manipulator via

Adaptive Neural Networks,” *Procedia Technol.*, vol. 3, pp. 139–146, Jan. 2012, doi: 10.1016/j.protcy.2012.03.015.

Anexo A. Diagrama Simulink



Anexo B. Código MatLab

```
MATLAB Function*  X  +
1  function out = fcn(theta,in)
2  % parâmetros
3  m1 = 1;
4  m2 = 1;
5  l1 = 0.2;
6  l2 = 0.2;
7  g = 9.81;
8
9  M11 = m1*l1^2+m2*(l1^2+l2^2+2*l1*l2*cos(theta(2)));
10 M12 = m2*l1*l2*cos(theta(2))+m2*l2^2;
11 M21 = m2*l1*l2*cos(theta(2))+m2*l2^2;
12 M22 = m2*l2^2;
13 M = [M11 M12; M21 M22];
14 out = M\in;
```

```
MATLAB Function1  X  +
1  function C = fcn(theta,thetad)
2  % parâmetros
3  m1 = 1;
4  m2 = 1;
5  l1 = 0.2;
6  l2 = 0.2;
7  g = 9.81;
8
9  C11 = -m2*l1*l2*sin(theta(2))*thetad(2)^2-...
10      2*m2*l1*l2*sin(theta(2))*thetad(1)*thetad(2);
11 C21 = m2*l1*l2*sin(theta(2))*thetad(1)^2;
12 C = [C11; C21];
```

```
MATLAB Function2  X  +
1  function G = fcn(theta)
2
3  % parâmetros
4  m1 = 1;
5  m2 = 1;
6  l1 = 0.2;
7  l2 = 0.2;
8  g = 9.81;
9
10 G11 = (m1*l1*cos(theta(1))+m2*(l1*cos(theta(1))+l2*cos(theta(1)+theta(2))))*g;
11 G21 = (m2*l2*cos(theta(1)+theta(2)))*g;
12 G = [G11; G21];
```