

JOAQUIM SALUFUNDA MACUACUA

Licenciado em Engenharia de Sistemas e Tecnologias de Informação

AMBIENTE COLABORATIVO BASEADO NO ALTRUÍSMO: SIMULAÇÃO E ANÁLISE BASEADA EM ROS

MESTRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Universidade NOVA de Lisboa
Novembro, 2024

AMBIENTE COLABORATIVO BASEADO NO ALTRUÍSMO: SIMULAÇÃO E ANÁLISE BASEADA EM ROS

JOAQUIM SALUFUNDA MACUACUA

Licenciado em Engenharia de Sistemas e Tecnologias de Informação

Orientador: José Barata,

Professor Catedrático, Universidade NOVA de Lisboa

Coorientador: Luis Alberto Estrada Jimenez,

Estudante do Programa Doutoral em Engenharia Electrotécnica e de Computadores

Júri:

Presidente: João Pedro Abreu de Oliveira,

Doutor e Professor Associado com Agregação, FCT-NOVA de Lisboa

Arguente: André Dionísio Bettencourt da Silva Rocha,

Doutor e Professor Auxiliar, FCT-NOVA de Lisboa

Orientador: José António Barata de Oliveira,

Doutor e Professor Catedrático, FCT-NOVA de Lisboa

Membro: Luis Alberto Estrada Jimenez,

Doutorando, Dep. Eng. Electrotécnica e de Computadores, FCT-NOVA de Lisboa

MESTRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Universidade NOVA de Lisboa

Novembro, 2024

Ambiente colaborativo baseado no altruísmo: Simulação e análise baseada em ROS

Copyright © Joaquim Salufunda Macuacua, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

DEDICATÓRIA

Dedico este trabalho aos meus pais, pela educação, valores e pelo amor e apoio incondicional que me guiaram até aqui. Aos meus irmãos, pelo companheirismo e apoio ao longo de todos os momentos.

AGRADECIMENTOS

Inicialmente, começo por agradecer a Deus por permitir que esta tese de mestrado fosse realizada dentro do tempo estabelecido e por me guiar sempre que precisava.

A sua conclusão só foi possível graças ao apoio e orientação de duas personagens, a quem gostaria de expressar a minha profunda gratidão.

Em primeiro lugar, gostaria de agradecer ao meu orientador, Prof. José Barata, pela sua orientação contínua, paciência e expertise ao longo deste trabalho. Agradeço também ao meu coorientador Luis Alberto Estrada Jimenez por me guiar nas mais variadas opções em busca de melhores soluções, a sua sabedoria, sugestões e disponibilidade foram fundamentais para o desenvolvimento e sucesso desta tese.

No âmbito institucional, gostaria de manifestar o meu agradecimento ao Departamento de Engenharia Eletrotécnica e de Computadores, pelo ambiente de aprendizagem que proporciona, pelos recursos disponibilizados e pela oportunidade de desenvolver este projeto no seio de uma instituição de prestígio. Reconheço também o apoio e o conhecimento partilhado pelos professores e colegas que cruzaram o meu caminho ao longo desta jornada académica.

Por fim, mas não menos importante, gostaria de agradecer aos meus amigos, colegas e à minha família pelo apoio incondicional e encorajamento durante todo o mestrado.

A todos, o meu sincero agradecimento por tornarem esta conquista possível.

“You cannot teach a man anything; you can only help him
discover it in himself.” (Galileo).

RESUMO

Este trabalho de dissertação aborda a criação de um ambiente colaborativo no sistema de fabricação inteligente, explorando conceitos centrais da Indústria 4.0, produção flexível e a biologicalização. O foco principal incide no desenvolvimento de comportamentos altruístas entre robôs autônomos, permitindo que colaborem mutuamente para superar dificuldades, como falhas simples ou necessidades de energia, com o objetivo de manter o ritmo produtivo. Foram investigadas questões como a possibilidade de os robôs se ajudarem uns aos outros e como o altruísmo pode ser aplicado no contexto da fabricação colaborativa. A necessidade de aplicação do conceito de altruísmo entre os robôs baseia-se na redução ou eliminação do tempo de paragem das máquinas das indústrias, provocado pela falta de energia e pelo tempo de carregamento dos robôs autônomos. A solução apresentada inclui a implementação do comportamento altruísta por meio do ROS2, a criação de uma arquitectura para a aplicação do altruísmo, o desenvolvimento de um algoritmo específico para testar o comportamento do altruísmo entre os robôs, a apresentação de um caso de estudo para realçar a sua aplicação prática e a realização de simulações com o algoritmo para demonstrar a sua eficiência e medir o tempo necessário para realizar tarefas de partilha de energia. Os resultados deste trabalho demonstraram que robôs altruístas podem partilhar informações, como localização, energia e pedidos de ajuda, para superar desafios como falta de energia ou avarias, reduzindo interrupções na produção. A validação dos resultados destacou uma maior eficiência em relação aos processos manuais, com poupança financeira significativa e melhor aproveitamento dos períodos de inatividade dos robôs. Esta abordagem contribui para a evolução do conceito de fabricação inteligente (Smart Manufacturing), promovendo maior continuidade e eficiência nos processos produtivos.

Palavras-chave: Altruísmo, Auto-organização, Autoajuda, Fabricação inteligente, Robôs móveis, Ros2, Doador, Receptor, Turtlebot.

ABSTRACT

This dissertation addresses the creation of a collaborative environment in the intelligent manufacturing system, exploring core concepts of Industry 4.0, flexible manufacturing, and biologicalization. The main focus lies in the development of altruistic behaviors among autonomous robots, enabling them to collaborate mutually to overcome difficulties, such as simple failures or energy needs, with the aim of maintaining the production pace. Questions were investigated, such as the possibility of robots helping each other and how altruism can be applied in the context of collaborative manufacturing. The need for applying the concept of altruism among robots is based on reducing or eliminating downtime in industrial machines, caused by a lack of energy and the charging time of autonomous robots. The proposed solution includes the implementation of altruistic behavior using ROS2, the creation of an architecture for the application of altruism, the development of a specific algorithm for testing the behavior of altruism among robots, the presentation of a case study to highlight its practical application, and the execution of algorithm simulations to demonstrate its efficiency and measure the time required to perform energy-sharing tasks. The results of this project showed that altruistic robots can share information, such as location, energy and help requests, to overcome challenges like energy shortages or breakdowns, reducing production interruptions. The validation of the results highlighted greater efficiency compared to manual processes, with significant financial savings and better use of the robots' idle periods. This approach contributes to the evolution of the concept of intelligent manufacturing (Smart Manufacturing), promoting greater continuity and efficiency in production processes.

Keywords: Altruism, Self-organization, Self-help, Smart manufacturing, Mobile robots, Ros2, Doador, Receptor, Turtlebot.

ÍNDICE

1	INTRODUÇÃO.....	1
1.1	Contexto geral.....	1
1.1	Problema e objetivo.....	2
1.2	Organização do documento	2
2	ESTADO DA ARTE E CONCEITO.....	3
2.1	Conceitos gerais	3
2.1.1	A revolução industrial	3
2.1.2	Quarta revolução industrial.....	4
2.1.3	Auto-organização na fabricação	7
2.1.4	Biologicalização e bioinspiração na fabricação	9
2.1.5	Sistemas auto curativos na fabricação	10
2.1.6	Altruísmo e altruísmo recíproco	11
2.1.7	Sistema ROS	12
2.2	Trabalhos relacionadas	14
3	ARQUITECTURA	19
3.1	Altruísmo no processo de fabricação	19
3.2	Arquitectura geral.....	21
3.3	Processo altruístico entre os robôs	25
4	IMPLEMENTAÇÃO DO PROJETO.....	27
4.1	Sistema de simulação com ROS2.....	27
4.1.1	Plataforma de robô padrão TurtleBot	29
4.1.2	Tecnologia de navegação Nav2.....	30
4.1.3	Linguagem de programação Python	32
4.1.4	Simulador Gazebo	32
4.2	Cenário de aplicação do altruísmo entre os robôs	35

4.2.1	Parâmetros e condições para tornar-se altruísta.....	35
4.2.2	Cenários de altruísmos entre os robôs	38
4.3	Sistema de partilha de energia.....	47
5	RESULTADO E TESTES.....	49
5.1	Caso de estudo.....	49
5.2	Simulação dos AGV's no estudo de caso.....	57
5.3	Comparação entre o processo altruístico e o não- altruístico (manual) e métricas de avaliação	67
5.3.1	Testes em cenários diferentes	70
5.4	Alternativas de carregamento dos AGV's.....	75
6	CONCLUSÃO	79
6.1	Futuro trabalho	80
7	BIBLIOGRAFIA	81

ÍNDICE DE FIGURAS

Figura 2.1 - Etapas da revolução industrial.....	4
Figura 2.2 - Pilares da quarta revolução industrial.....	5
Figura 2.3 - Auto-organização e representação de emergente.....	8
Figura 2.4 - Principais elementos do ROS.....	13
Figura 3.1 -Representação do altruísmo entre os robôs.....	20
Figura 3.2 - Arquitectura do Ros2.....	21
Figura 3.3 - Arquitectura do altruísmo entre os robôs no ros2.....	23
Figura 3.4 – a) Definição do processo altruísta do projeto pela partilha de carga.....	25
Figura 4.1 - Estrutura do Worksapce do ROS2.	27
Figura 4.2 - Estrutura do Worksapce do projeto.....	28
Figura 4.3 - Representação do TurtleBot3 Waffle	30
Figura 4.4 - Construção de rota com o Nav2.....	31
Figura 4.5 - Arquitectura do Nav2.....	31
Figura 4.6 - TurbleBot3 World para o turtleBot3.	33
Figura 4.7 - Representação do TurtleBot3 no Gazebo.....	34
Figura 4.8 - Representação do TurtleBot3 dentro do Gazebo turtleBot3 world.....	35
Figura 4.9 - Cenário Um para Um / Single-mode do altruísmo entre os robôs.	39
Figura 4.10 - Algoritmo do altruísmo no cenário Um para Um / Single-mode.	40
Figura 4.11 - Cenário Um para Muitos / Single-multi mode do altruísmo entre os robôs.....	41
Figura 4.12 - Algoritmo do altruísmo no cenário Um para Muitos / Single-multi mode.....	42
Figura 4.13 - Cenário Muitos para Um / Multi-single mode do altruísmo entre os robôs.	43
Figura 4.14 - Algoritmo do altruísmo no cenário Muitos para Um/ Multi-single mode.....	43
Figura 4.15 - Cenário Muitos para Muitos / Multi-mode do altruísmo entre os robôs.....	44
Figura 4.16 - Algoritmo do altruísmo no cenário Muitos para Muitos/ Multi-mode.....	45
Figura 4.17 - Processo de partilha de carga por aproximação entre os robôs.	47
Figura 5.1 - Sensores fabricados pela Senseair.....	50
Figura 5.2 - Esquema de fabricação da Senseair.....	50
Figura 5.3 - Processo de fabricação manual da Senseair.	51
Figura 5.4 - Célula de produção All-in-one.....	51
Figura 5.5 - Ambiente colaborativo com altruísmo entre os robôs.....	52

Figura 5.6 - Ambiente colaborativo com robôs agindo de forma altruística	56
Figura 5.7 - Ambiente virtual de simulação	57
Figura 5.8 - Diferenciação entre robôs doadores e receptores.	58
Figura 5.9 - Robô de simulação na posição (0.0, 0.0).	60
Figura 5.10 - Cálculos das distâncias e verificação do tempo entre os robôs.	61
Figura 5.11 - Sistema de coordenadas do robô na posição (0, 0).	63
Figura 5.12 - Robô de simulação movimenta-se para posição do altruísta Robot6 (3.5, 0.5). ..	63
Figura 5.13 - Movimento do robô altruísta para a posição do Robot2 (0, 3.5).	64
Figura 5.14 - Robô de simulação movimenta-se para posição do altruísta Robot5 (-1.5, 3).	64
Figura 5.15 - Movimento do robô altruísta Robot5 para a posição do Robot1 (0.5, 2.5).	65
Figura 5.16 - Movimento do robô altruísta Robot5 voltando à sua posição inicial (-1.5, 3).	65
Figura 5.17 - Output do sistema após a finalização do altruísmo.	66
Figura 5.18 - Comparação entre o Downtime do altruísmo e o manual.	72
Figura 5.19 - Comparação entre a Perda financeira do altruísmo e o manual.	73
Figura 5.20 - Comparação entre a Perda financeira do altruísmo e o manual com tempo de partilha de 60s.	74
Figura 5.21 - Comparação entre a Perda financeira do altruísmo e o manual com tempo de partilha de 180s.	74

ÍNDICE DE TABELAS

Tabela 2.1 - Bioinspiração no sistema de fabricação	14
Tabela 3.1 - Descrição da arquitetura do Ros2	21
Tabela 5.1 - Critérios de um solicitante.	53
Tabela 5.2 - Critérios de um candidato à altruísta.....	53
Tabela 5.3 - Cálculo do tempo necessário para uma ação altruísta baseada na distância.	54
Tabela 5.4 - Dados dos robôs no ambiente colaborativo.	55
Tabela 5.5 - Dados dos robôs no ambiente colaborativo depois de várias etapas de trabalho	55
Tabela 5.6 - Distância entre os candidatos e os solicitantes.	55
Tabela 5.7 - Parâmetros dos robôs doadores e receptores.....	62
Tabela 5.8 - Cálculo do Downtime em cada um dos processos (altruísmo e manual) para o Robot1.	68
Tabela 5.9 - Cálculo do Downtime em cada um dos processos (altruísmo e manual) para o Robot2.	69
Tabela 5.10 - Duração em minutos de cada etapa de produção.	69
Tabela 5.11 - Preço dos produtos da Senseair.	69
Tabela 5.12 - Comparação da perda financeira entre processo manual e altruístico.	70
Tabela 5.13 - Testes do processo manual e altruístico em diferentes cenários.....	71
Tabela 5.14 - Comparação das médias do processo manual e altruístico.	73
Tabela 5.15 - Comparação das médias do processo manual e altruístico com tempo de partilha de 60s.....	74
Tabela 5.16 - Comparação das médias do processo manual e altruístico com tempo de partilha de 180s.....	75
Tabela 5.17 - Comparação das formas de carregamentos dos AGV's.	77

SIGLAS

ROS	Robot Operating System.
RA	Realidade Aumentada.
IoT	Internet of Things.
AGV	Automated Guided Vehicle.
IA	Inteligência Artificial.
SLAM	Simultaneous Localization and Mapping.
DDS	Data Distribution Service.
PF	Perda financeira.

INTRODUÇÃO

Este capítulo apresenta uma introdução ao trabalho incluindo o problema e o objetivo bem como a forma como este está organizado.

1.1 Contexto geral

Desde os tempos antigos que o ser humano vem procurando sempre uma forma de melhorar as suas tecnologias em diversas áreas, com o objetivo de explorar cada vez mais os recursos que temos à nossa disposição, bem como para tornar tudo mais automatizado. Desde o início da era industrial até à Indústria 4.0, a sociedade tem vindo a crescer e a modernizar-se, trazendo consigo muitas exigências e novas necessidades que as indústrias têm lutado para responder, o que obriga ao avanço de novas tecnologias industriais para o sistema de fabrico [1].

Neste contexto, emergem temas de alta relevância para a automação na fabricação inteligente envolvendo robôs autónomos, indústria inteligente e muitas outras tecnologias que podem alavancar a Indústria 4.0. Esses temas têm vindo a crescer cada vez mais à medida que avançamos para a nova era industrial. Com os robôs a tornar-se mais independentes de especialistas humanos, com a capacidade de realizar tarefas sem a constante intervenção, o que levanta a questão sobre a real necessidade de um ambiente de cooperação e organização para equilibrar e otimizar o trabalho coletivo [8]. A cooperação é a união entre elementos individuais ou organizações, com o propósito de alcançar um objetivo comum.

Outro fator muito importante neste contexto trata-se do conceito de bioinspiração, que tem sido a base de muitas tecnologias que utilizamos hoje e certamente de tecnologias que irão surgir no futuro. Muitas formas de vida que existem na natureza têm sido usadas como objeto de estudo e, como resultado, dão origem a conceitos e pesquisas robustas que têm vindo a ser aplicadas no mundo real e, em particular, no sistema de fabrico e na Indústria 4.0 [22]. Sendo assim, nesta fase da revolução industrial, os conceitos aprendidos com a natureza sobre a cooperação e a autoajuda desempenham um papel fundamental para criar um ambiente mais adaptado à nova realidade [30].

1.1 Problema e objetivo

Assim sendo, este trabalho aborda alguns dos principais tópicos do ambiente colaborativo do sistema de fabrico inteligente, como resultado da exploração de diversas pesquisas encontradas sobre temas relevantes da Indústria 4.0, sistema de fabrico inteligente, fabrico flexível e a biologicalização. Muitas das investigações que abordam o tema fornecem uma base de estrutura geral que pode ser aplicada em diferentes contextos para solucionar problemas genéricos. Contudo, no sistema de fabrico, a capacidade desses agentes autónomos de realizar tarefas complexas sem intervenção direta de especialistas representa um avanço significativo e exige uma abordagem bem mais específica. No contexto do fabrico inteligente, são considerados agentes autónomos robôs que podem controlar, de certa forma, as suas tarefas e o seu próprio destino de forma automática, possuindo equipamentos ou mecanismos que lhes permitem sentir o ambiente à sua volta para agir sobre ele sem a constante intervenção externa [41]. Entretanto, a busca por níveis mais elevados de autonomia desafia os paradigmas convencionais, levantando questões cruciais sobre a capacidade dos robôs se autoajudarem em situações adversas. Nessa ordem de ideias, duas questões são relevantes para este trabalho:

- Será que os robôs podem ajudar-se uns ao outro para superar dificuldades ou falhas simples com o objetivo de que a produção continue no mesmo ritmo?
- Como pode-se usar e aplicar o altruísmo para criar um ambiente colaborativo no sistema de fabricação?

1.2 Organização do documento

O Capítulo 1 apresenta uma visão introdutória sobre o trabalho. Em seguida, no Capítulo 2, são descritos os conceitos fundamentais sobre a revolução industrial, Indústria 4.0 e os seus pilares, a auto-organização no fabrico, biologicalização e bioinspiração, sistemas autocurativos no fabrico, altruísmo e o altruísmo recíproco, bem como o sistema ROS. Na parte final, apresenta-se o estado da arte, que denominamos de trabalhos relacionados. Segue-se o Capítulo 3, onde é apresentada a arquitetura do projeto, descrevendo a sua estrutura de construção, funcionamento e integração com o ROS2. No Capítulo 4, é feita a implementação do projeto, demonstrando as principais tecnologias, algoritmos e os cenários em que o altruísmo pode ocorrer no ambiente colaborativo. No Capítulo 5, são apresentados os resultados e testes do trabalho. Neste capítulo, é descrito um caso de estudo onde o altruísmo é aplicado e diversos testes são realizados em diferentes cenários, cujos resultados são mensurados e comparados com a solução manual para o mesmo problema. Por fim, temos o Capítulo 6, que denominamos de conclusão. Tal como o próprio nome indica, é neste capítulo que se faz o desfecho do trabalho, com as considerações finais baseadas nos testes e resultados obtidos, e são dadas algumas sugestões para um futuro trabalho na área.

ESTADO DA ARTE E CONCEITOS

Neste capítulo é apresentado os principais conceitos que fazem parte deste trabalho e as principais fontes de inspiração que levaram ao desenvolvimento deste tema.

2.1 Conceitos gerais

2.1.1 A revolução industrial

Tudo começou no século XVIII com James Watt, quando este melhorou o conceito e o funcionamento da máquina a vapor, permitindo que a indústria têxtil se desenvolvesse e se tornasse um símbolo da produção de alto nível nesse período, criando um modelo económico. Assim nasceu a Primeira Revolução Industrial. A Segunda Revolução Industrial teve início nos primórdios do século XX, quando Henry Ford criou uma linha de produção em massa, que apresentava grandes benefícios face à sua antecessora em termos de custos, permitindo que a classe média trabalhadora pudesse adquirir esses produtos, trazendo mais vantagens para a indústria e para a economia. Cerca de 60 anos depois, começou a surgir a automação como resultado do avanço da Informática, que incluía a eletrónica e os computadores, dando início à Terceira Revolução Industrial. A sua principal característica foi a implantação de dispositivos com a capacidade de gerir uma ampla quantidade de variáveis de produção. Estes dispositivos incluíam computadores, controladores eletrónicos, sensores e outros. [1]. E, finalmente, no século XXI, com o avanço da inteligência artificial, o crescimento exponencial da capacidade dos computadores e a quantidade imensurável de informação digitalizada, teve início uma nova era: a Quarta Revolução Industrial. Atualmente, assistimos ao mundo a fazer uma transição da Terceira para a Quarta Revolução Industrial, distinguindo-se esta revolução pela fusão de tecnologias digitais, físicas e biológicas. [2]. As fases da revolução industrial podem ser resumidas em uma única imagem como se vê na Fig. 2.1.

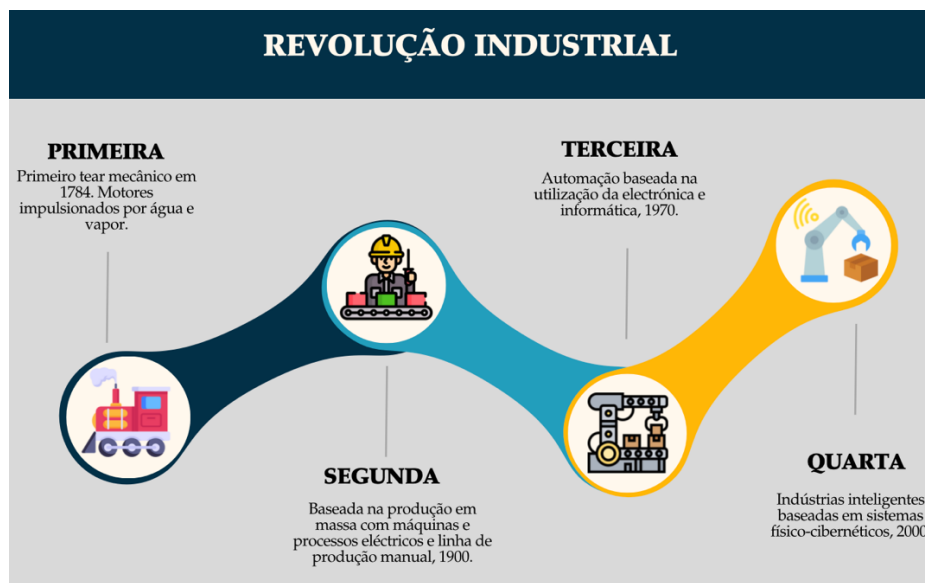


Figura 2.1 - Etapas da revolução industrial [23].

2.1.2 Quarta revolução industrial

O termo “Quarta Revolução Industrial” ou, ainda, “Indústria 4.0” tem-se tornado cada vez mais recorrente na atualidade, pois permite alterações na sua programação de acordo com os eventos que ocorram, tanto dentro como fora do ambiente industrial.

Esta Revolução Industrial pode também ser descrita como o surgimento de "sistemas ciberfísicos" que introduzem capacidades completamente novas para pessoas e máquinas. Embora a sua infraestrutura tecnológica dependa das tecnologias e infraestruturas da revolução anterior, a Quarta Revolução Industrial traz consigo novas formas de integração da tecnologia nas sociedades e, até mesmo, no corpo humano. Exemplos incluem a edição de genomas, novas formas de inteligência artificial, materiais inovadores e abordagens de governança que dependem de métodos criptográficos, como o blockchain [3].

Principais pilares da Quarta revolução industrial

A Quarta Revolução Industrial beneficiará da internet e de tecnologias digitais para se desenvolver cada vez mais em direção a sistemas ciberfísicos autônomos e inteligentes. Isto permitirá expandir ainda mais os seus principais pilares, de forma a garantir uma melhor adaptação ao mundo atual e às nossas necessidades insaciáveis.

A Fig. 2.2 ilustra de forma simples os principais pilares tecnológicos da Quarta revolução industrial, esses pilares são: cibersegurança, realidade aumentada, big data, robôs autônomos, fabricação aditiva, simulação, integração de sistemas, computação em nuvem e internet das coisas [4].



Figura 2.2 - Pilares da quarta revolução industrial [26]

Cibersegurança

A automatização da infraestrutura da indústria 4.0, tal como descrito acima, tem como resultado a relação de dependência entre grande parte dos processos industriais, o que gera uma enorme quantidade de dados gerados, recolhidos e processados pelos sistemas inteligentes de informação. Desta forma, a cibersegurança tem como principal objetivo proteger a infraestrutura da indústria 4.0 de invasões, combinando várias técnicas e esforços constantes devido ao crescente aprimoramento dos ataques cibernéticos [5].

Realidade aumentada

“Realidade Aumentada (RA) é a integração do ambiente real e virtual em tempo real, por meios tecnológicos, ou seja, através da utilização de um computador com webcam e outros dispositivos móveis que executam um programa, proporcionando a exibição de imagens virtuais no ambiente real. A principal característica da RA é a utilização e apresentação de objetos virtuais em situações reais(...)” [6].

Big Data

Segundo a gigante tecnológica, Google, “Big Data é a enorme quantidade de dados disponíveis para organizações que, por causa do volume e complexidade, não é facilmente gerenciada ou analisada por muitas ferramentas de Business Intelligence” [46].

Desta forma os atributos que definem Big Data são volume (petabytes, exabytes ou mais), variedade, velocidade e variabilidade.

Na indústria 4.0, a análise de Big Data desempenha um papel fundamental, sendo que essa nova geração de indústrias inteligentes (smart factories) produzem uma enorme quantidade de dados que se forem bem aproveitados podem contribuir bastante para o sucesso das organizações [7].

Robótica autônoma

A robótica industrial é um conceito introduzido na terceira revolução industrial, sendo que o que a diferencia da robótica na quarta revolução é a sua autonomia. Na indústria 4.0, os robôs possuem a capacidade de aprender através de tarefas realizadas utilizando recursos e técnicas de inteligência artificial. Esta nova geração de robôs tem a capacidade de sentir o ambiente à sua volta, permitindo-lhe realizar várias tarefas ou operações sem a supervisão direta de especialistas, tornando assim a indústria mais autônoma e produtiva [8].

Fabricação aditiva

A fabricação aditiva, comumente conhecida como impressão 3D, é uma técnica desenvolvida para a produção de uma enorme variedade de estruturas e geometrias complexas com base em dados de modelos tridimensionais que são construídos, ou seja, impressos por camadas sucessivas de materiais. A fabricação aditiva torna-se um ponto-chave para a indústria 4.0, pois possibilita a produção personalizada de uma ampla diversidade de formas de peças na escala desejada, reduzindo custos e tempo de fabrico, desempenhando assim um papel fundamental para a parte física das indústrias inteligentes, principalmente em termos de eficiência, capacidade e complexidade de produção [9].

Computação em nuvem

A Google define a computação em nuvem como sendo “a disponibilidade sob demanda dos recursos de computação como serviços na internet”. Ela elimina a necessidade de as empresas adquirirem, configurarem ou gerirem a infraestrutura, assim elas pagarão apenas pelo que utilizarem [10]. A principal característica da computação em nuvem é a virtualização de recursos e serviços de computação, permitindo que os recursos de computação sejam entregues com cinco características principais, ou seja, auto-serviço sob demanda, amplo acesso à rede, agrupamento de recursos, elasticidade rápida e serviço medido, tornando-a uma tecnologia chave para a implementação da indústria 4.0 [11].

Simulação

A simulação computacional na indústria 4.0 preocupa-se em utilizar de forma mais ampla informações obtidas desde o chão de fábrica ou planta até a etapa superior, analisando os seus dados em tempo real para aproximar o mundo físico do virtual, facilitando a visualização de erros, custo-benefício e tempo. Segundo Silva et al. (2007), a simulação é uma ferramenta que auxilia na prevenção de problemas, testes de configurações e busca de soluções antes que se implemente qualquer alteração no ambiente real [12].

Integração de sistemas

A integração de sistemas consiste na união dos sistemas de gestão e controlo, de modo a interligar diretamente o chão de fábrica com o nível corporativo. O seu principal objetivo é transformar em informação útil os dados gerados pelos sistemas e processos que envolvem a fabricação e a comercialização de produtos. A integração de sistemas pode ser feita de forma vertical e horizontal. A integração vertical permite obter uma visão detalhada do seu ambiente interno ao realizar integrações de sistemas de tecnologias de informação em vários níveis hierárquicos, desde dispositivos físicos do chão de fábrica, como: sistemas de controlo de máquinas, linha de produção, planeamento da produção, controlo da qualidade, até processos de negócios como vendas e marketing. Por outro lado, a integração horizontal é direcionada para a cadeia de suprimentos/valor, cujo objetivo primordial é a interface de dados entre os sistemas, tendo como base a integração e monitorização de toda a cadeia de valor, seja ela interna, como: planeamento, produtividade, lead time, satisfação dos funcionários, ou externa, como fornecedores e clientes [13].

Internet das coisas

A Internet das Coisas (IoT) pode ser definida como a expansão da internet atual, que surgiu como resultado dos avanços tecnológicos de várias áreas, permitindo que os objetos do nosso quotidiano estejam conectados à internet, desde que possuam capacidade computacional e de comunicação. Desta forma, é inegável que a IoT trará mudanças para a indústria 4.0, afetando principalmente os meios de produção para criar tendências e estimular a criação de novos negócios [14].

Com o surgimento da indústria 4.0, aumenta a necessidade de equipamentos com a capacidade de agir com maior autonomia, quer seja em termos de hardware, quer em termos de software, que possam adaptar-se e auto-organizar-se sem a interferência de um profissional, mesmo quando há mudanças no ambiente e em casos de falhas.

2.1.3 Auto-organização na fabricação

O sistema de produção flexível que surgiu nos últimos anos, fazendo uso de veículos guiados automaticamente (AGVs), propôs um novo conceito para o sistema de produção: a auto-organização na fabricação. Segundo Kubota, “A auto-organização na fabricação é capaz de reorganizar o seu hardware e também o seu software por si só”, tornando o controlo e o gerenciamento da fabricação descentralizados [21].

A auto-organização na fabricação pode ser definida com base em diferentes critérios, mas existirá sempre uma convergência entre elas tendo as seguintes principais características [25]:

- Espontâneo: capacidade de se adaptar reorganizando-se sem interferência externa a ambientes em constantes mudanças e em caso de falhas ou outros motivos.
- Distribuído: não existe um nó central, ao contrário disso, possui vários nós distribuídos de controlo autônomos que trocam informações entre si.
- Autonomia individual: os componentes podem resolver individualmente as tarefas simples, ao passo que tarefas ou problemas mais complexos podem ser negociados com base na autonomia.
- Autoaprendizagem: capacidade de se adaptar e se otimizar com base no que aprende com o ambiente em que se encontra ou pelas alterações que nela ocorrem.
- Reorganização: reorganizar e ajustar seu sistema para criar funções que podem lidar com diferentes necessidades.

Desta forma, a Fig. 2.3 ilustra o processo auto-organizado e emergente como resultado de interações locais.

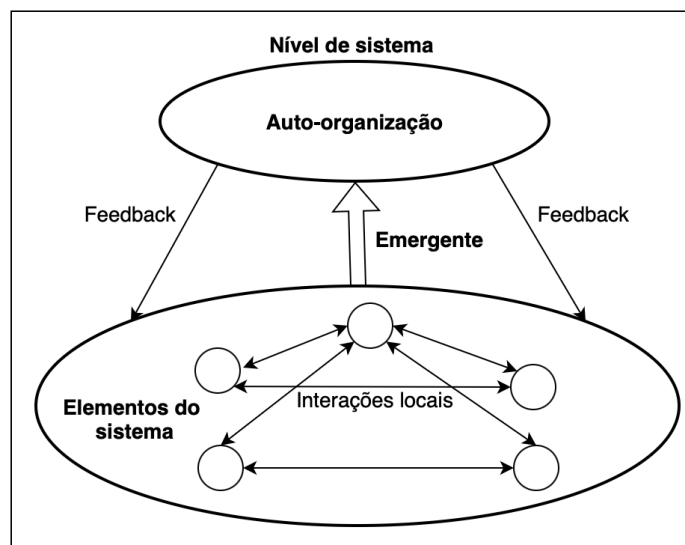


Figura 2.3 - Auto-organização e representação de emergente [28].

O conceito de auto-organização tem sido incansavelmente pesquisado na área de engenharia de software e para o uso em enxame de robôs bem como também pode ser um campo de aplicação para os sistemas de produção inteligentes, especificamente, a produção orientada ao produto [24].

Se olharmos para a natureza, veremos que existem muitos exemplos que poderiam ser mencionados para ilustrar a auto-organização, como, por exemplo, um cardume de peixes, uma colônia de formigas ou um bando de pássaros que se auto-organizam para sobreviver. Daí vem a nossa fonte de inspiração, pois muitos conceitos que utilizamos nas áreas da engenharia, medicina e em muitos outros campos provêm da vida animal, o que chamamos de bioinspiração.

2.1.4 Biologicalização e bioinspiração na fabricação

Biologicalização

Nestes últimos anos, surgiu na Alemanha uma área denominada “Biologicalização”, que está relacionada com princípios e recursos da natureza em sistemas técnicos com o potencial de trazer consigo novos mercados para indústrias ligadas a tecnologias e sistemas de produção [19].

A biologicalização na fabricação é definida como a “utilização e integração de princípios, materiais, funções, estruturas e recursos biológicos e bioinspirados para tecnologias e sistemas de produção inteligentes e sustentáveis, com o objetivo de atingir todo o seu potencial”. A biologicalização tem o potencial de afetar amplamente o ambiente industrial e empresarial, uma vez que existem evidências que mostram que as fases iniciais de uma transformação biológica estão a ocorrer com alguma interseção entre a biomimética, a biotecnologia e a bioeconomia [19]. Um outro conceito que já tem sido muito utilizado e seus resultados já são visíveis é a bioinspiração.

Bioinspiração

A nossa curiosidade em conhecer melhor a natureza, bem como a forma como os animais agem, se locomovem (nadam ou voam), tem resultado em muitas invenções tecnológicas que beneficiam a nossa sociedade. Desta forma, podemos entender o que é a bioinspiração: trata-se de criar ou recriar tecnologias que estão muito acima da capacidade daquela que é a sua fonte natural. As ideias que obtemos da natureza (biologia) têm sido uma fonte inspiradora para a criação de diversas técnicas e métodos que nos têm ajudado a solucionar problemas complexos de engenharia, como logística e otimização de tráfego, redes de telecomunicações, mercados económicos e sistemas de produção [22]. Os sistemas multiagentes utilizados na indústria já incorporam, de certa forma, ideias vindas da biologia e da natureza. Porém, ainda existem melhorias que podem ser feitas ao adicionar mais conhecimentos biológicos, como, por exemplo, a inteligência e a auto-organização de enxame, tornando-os mais responsivos e adequados para lidar e atender aos requisitos contemporâneos impostos aos sistemas de produção [22]. Para benefício da indústria, as técnicas bioinspiradas contribuem de forma significativa para a obtenção de melhores resultados no sistema de produção, em aspectos como a flexibilidade, robustez, reconfiguração e capacidade de resposta [20]. Requer mencionar que uma das técnicas inspiradas na natureza é a capacidade que os sistemas possuem de se autocurarem e autossustentarem, sem a necessidade frequente de um profissional humano.

2.1.5 Sistemas auto curativos na fabricação

Com o avanço da tecnologia e a transição para a nova era da indústria 4.0, o sistema de fabrico inteligente tem vindo a lidar com exigências como a resposta rápida a falhas no chão de fábrica, o equilíbrio de stocks para a produção personalizada e o ajuste flexível de cronogramas operacionais de acordo com o estado do produto e do chão de fábrica, exigências essas que são difíceis de satisfazer simultaneamente [29]. Como esses avanços vêm aumentando exponencialmente a complexidade do sistema de fabrico inteligente, uma vez que envolvem as questões mais recentes, como IoT, Inteligência Artificial (IA), robôs, máquinas, transportadores, sensores inovadores, eletrónica e muito mais [29].

Consequentemente, aumenta o número de problemas ou falhas que podem interromper e até parar abruptamente todo o ciclo produtivo. Desta forma, requerem-se soluções inovadoras auto-curativas ou autocorretivas para melhorar o desempenho do ciclo de produção e elevar a qualidade do processo em geral [31].

No processo de fabricação, os erros podem prejudicar diretamente o produto e gerar uma sequência de outros problemas, resultando em perdas financeiras para a empresa e danos ambientais no mundo. O sistema de produção deve possuir alguma capacidade de se auto-curar ou corrigir, considerando que os elementos que fazem parte do sistema podem fornecer “curas”, ou seja, possuir a capacidade de ajudar outros elementos do mesmo sistema.

Desta forma, para criar um ambiente de cura colaborativo, onde o objetivo principal de ter um chão de fábrica adaptável e flexível é fundamental, é necessário que haja uma pequena redução da aptidão individual. Assim sendo, o ambiente de cura colaborativo pode ter como foco as novas infraestruturas de produção que possuam os níveis mais elevados de escalabilidade, modularidade e mobilidade, visto que estas estão a tornar-se a tendência atual devido aos novos métodos e recursos de automação e modelos de decisão para o gerenciamento de robôs móveis autónomos [33].

Sistemas de auto cura e abordagens tolerantes a falhas

As abordagens de sistemas auto-curáveis ou auto-corrigíveis permitem que o sistema se mantenha funcional de forma eficiente, apesar de haver algumas falhas. Este conceito deriva da investigação de sistemas tolerantes a falhas e auto-estabilizadores.

Estes sistemas, que podemos chamar de sistemas tolerantes a falhas, enfrentam falhas transitórias e mascaram falhas permanentes para voltar a um estado válido. Por outro lado, os sistemas auto-estabilizadores, ao contrário dos sistemas tolerantes a falhas, não fazem mascaramento de falhas [34].

As abordagens de autocura e tolerância a falhas mencionadas acima, são divididas em 2 categorias ou métodos: passivo e ativo.

Controlo passivo tolerante a falhas

Os métodos passivos referem-se ao desenvolvimento de técnicas de controlo robustas, em que a lista de potenciais defeitos é previamente reconhecida como defeitos fundamentais de projecto e é incorporada durante a fase de desenvolvimento do sistema de controlo.

A técnica de controlo robusto exerce um papel crucial ao assegurar que o sistema em malha fechada seja resistente a um conjunto de falhas já conhecidas. O que significa que o termo "passivo" denota que, diante de maus funcionamentos, não é acionado nenhum procedimento pelo sistema de controlo. Em suma, este método é baseado na simples ideia de que, no início do projecto da lei de controlo, devem ser consideradas as falhas para que o sistema ou projecto seja insensível a elas [38].

Controlo ativo tolerante a falhas

Por outro lado, o método de controlo activo tolerante a falhas responde a falhas reconfigurando a lei de controlo em tempo real, a fim de manter a estabilidade e eficácia dos valores nominais do sistema. O seu foco é alcançar os objectivos de controlo com a menor degradação possível no desempenho do sistema, quer por meio do uso de uma lei de controlo pré-calculada, quer pela síntese e actualização de uma estratégia de controlo online. Em casos de defeitos no sistema, o método activo deve manter a estabilidade do sistema a um nível aceitável de desempenho, operando em tempo real [38].

Falando sobre a autocura, um dos conceitos relevantes que surge no meio disso é o altruísmo e o altruísmo recíproco.

2.1.6 Altruísmo e altruísmo recíproco

De acordo com o dicionário de língua portuguesa, Porto Editora, o altruísmo é definido como “sentimento de interesse e dedicação por outrem; orientação de quem procura garantir o bem de outro(s), mesmo se à custa dos próprios interesses” [47].

No meio animal, o conceito de altruísmo e altruísmo recíproco foi introduzido pela primeira vez por Trivers (1971), numa pesquisa sobre a biologia evolutiva, onde ele descreve o termo “altruísmo” como ações feitas para beneficiar outra pessoa que não seja aquela que age de forma altruísta. Ainda no âmbito da mesma pesquisa, ele apresentou os conceitos de custos e ganhos, com argumentos de que, geralmente, as ações de caridade tendem a retornar sob forma de favores, tornando-se benéficas para quem doou inicialmente. Isto pode ser vantajoso dentro de uma rede estratégica para o desempenho tanto a nível individual como a nível comunitário ou social [35].

Altruísmo na vida animal

Falando de uma situação mais real na natureza, o altruísmo recíproco pode ser observado em morcegos vampiros. Visto que se alimentam exclusivamente de sangue, o que não é fácil de obter, há momentos em que alguns morcegos vampiros não conseguem encontrar alimento, o que pode representar um risco de morte se a situação se repetir, uma vez que estes morcegos podem sobreviver sem comida durante até três noites antes de morrerem.

É neste ponto que o altruísmo recíproco se torna uma estratégia evolutivamente estável, porém a partilha de alimento com o vizinho só ocorre quando alguns conseguem obter mais alimentos do que o necessário. Este comportamento de partilha de comida entre morcegos vampiros é um processo que acontece naturalmente, seja entre morcegos parentes ou não parentes, e, além disso, também pode ser induzido artificialmente [35].

Certamente que os morcegos não são os únicos seres na natureza a ajudarem-se uns aos outros. Outro exemplo a ser mencionado é um estudo realizado com os papagaios cinzentos africanos, que foram ensinados a trocar moedas por comida. Neste estudo, alguns papagaios receberam moedas para trocar por comida, enquanto outros não receberam nenhuma moeda. No entanto, verificou-se que alguns dos papagaios que possuíam moedas as deram a outros papagaios que não as possuíam [36]. Ou seja, os papagaios cinzentos africanos agiram de forma espontânea e voluntária para ajudar o seu parceiro no compartimento ao lado, que não tinha como obter comida, fornecendo-lhe moedas para que pudessem ter o suficiente para se alimentar. No caso de altruísmo social ou comunitário, as formigas são um verdadeiro exemplo disso, pois são umas das criaturas mais altruístas que existem, devido ao seu elevado nível de ajuda mútua. Uma das suas principais características mais conhecidas é a capacidade de desenvolver colônias bem organizadas, em que cada elemento é responsável por uma tarefa na qual se dedica ao bem-estar e segurança da colônia [37].

O altruísmo dentro da colônia de formigas pode ser observado nas chamadas formigas operárias, que armazenam alimento e água no estômago para partilhar com as formigas famintas ou doentes. Para evitar que alguma doença se propague entre os outros membros da colônia, estas formigas operárias levam para fora da comunidade as formigas doentes ou mortas, demonstrando assim uma preocupação com o bem-estar das restantes. Além disso, também cuidam dos ovos da formiga rainha e podem movê-los para outros locais que considerem mais seguros, sempre que necessário [37].

De forma geral, este comportamento altruísta descrito acima pode ser aplicado no conceito de fabrico inteligente. No âmbito industrial, o conceito de altruísmo tem um grande potencial para se tornar um modelo que poderá abrir portas para um novo campo de aplicação da cooperação e interligação no domínio social e industrial, graças à sua elevada autonomia e colaboração entre os elementos que integram o sistema. Assim, muitas tecnologias e ferramentas terão de ser desenvolvidas, enquanto outras necessitarão de adaptação. Uma das ferramentas mais utilizadas é o ROS, cuja finalidade é o desenvolvimento de aplicações na área da Robótica.

2.1.7 Sistema ROS

O ROS (Robot Operating System) é um framework de código aberto baseado em Linux para operar robôs que se tornou globalmente popular para o desenvolvimento de aplicações em Robótica [16]. Foi criado como parte de um projeto para a Universidade de Stanford e ao mesmo tempo fazia parte do Programa de Robôs Pessoais na Willow Garage por volta de

2007 [17]. O ROS foi projetado para lidar com diversos tipos de desafios específicos no desenvolvimento de serviços em larga escala aumentando a produtividade, fornecendo conceitos de publish/subscribe, múltiplas bibliotecas e várias ferramentas para a criação de aplicações robóticas [16]. E atualmente se encontra na sua segunda versão, chamada ROS2.

Elementos fundamentais do ROS2

O ROS2 é baseado em 4 elementos fundamentais, nomeadamente: Nós, Mensagens, Tópicos e Serviços tal conforme vemos na Fig. 2.4 [18]. No contexto do ROS2, o termo “Nó” é usado para representar um módulo de software com o objetivo de realizar algum processo. Neste caso, um sistema desenvolvido em ROS2 é composto por vários nós. Os nós podem se comunicar entre si, ponto-a-ponto, essa comunicação é feita pela troca de mensagens, o que nos leva ao segundo elemento, Mensagem.

A Mensagem é um conceito relevante no ROS2, representando uma estrutura de dados caracterizada pela organização de dados digitais. Esses dados podem incluir valores numéricos, booleanos, caracteres e muitos outros tipos. Assim, as mensagens podem conter outras mensagens de forma aninhada. As mensagens são publicadas pelos nós num tópico específico e subscritas por outros nós interessados nas mensagens desse tópico. O nó que publica as mensagens é designado por Publicador (Publisher), enquanto o nó que subscrive é chamado de Subscritor (Subscriber). Desta forma, um nó pode publicar e/ou subscriver vários tópicos, o que significa que, num único tópico, podem existir múltiplos publicadores e subscritores [17].

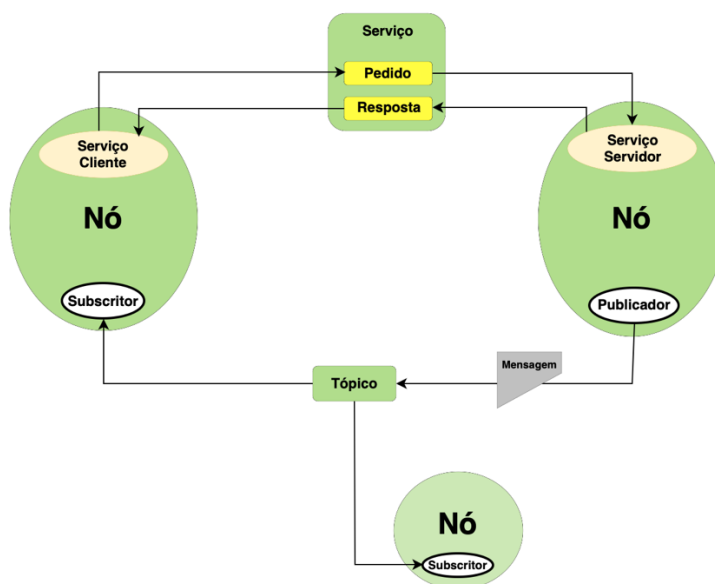


Figura 2.4 - Principais elementos do ROS [15].

Por último, temos os serviços, que operam de forma semelhante aos serviços web cliente/servidor, sendo a sua principal característica a comunicação síncrona. Neste modelo, o cliente envia uma solicitação e fica bloqueado até receber uma resposta. Nota-se que, ao contrário dos tópicos, apenas um nó pode anunciar um serviço com um nome específico [18].

A necessidade de criar robôs que respondam às exigências da sociedade moderna deu origem a uma nova versão do ROS, o **ROS2** (*Robot Operating System 2*). Introduzido em 2014, o ROS2 tem causado um impacto significativo na indústria de máquinas inteligentes, utilizadas em diversos sectores, desde o agrícola até ao espacial.

O conceito de aprender com a natureza está profundamente enraizado no espírito de cientistas e engenheiros, que agora começam a compreender a estrutura da natureza e a utilizá-la em produtos como implantes, na indústria aeroespacial, arquitectura e automóvel.

2.2 Trabalhos relacionadas

A intuição de aprender com a natureza está profundamente enraizada no espírito dos seres humanos. Como resultado, vemos ao longo de décadas cientistas e engenheiros a estudarem a natureza e a aplicarem os conceitos adquiridos dela nas áreas de tecnologia, produção industrial, indústria aeroespacial, arquitectura, automóvel e outros sectores [27]. Em particular, na fabricação industrial, muitos conceitos tiveram como base alguma forma de vida na natureza, ou seja, características únicas de determinados animais. Tal como se observa na Tabela 2.1, muitos destes conceitos resultaram em tecnologias que aceleraram o desenvolvimento industrial.

Tabela 2.1 - Bioinspiração no sistema de fabricação.

Referência	Bioinspiração	Conceito
[30] L. A. Estrada-Jimenez, S. Kalateh, S. Nikghadam-Hojjati, J. Barata, 2024.	Altruísmo dos morcegos	Baseado na capacidade que os morcegos têm para partilhar alimento com outros membros do grupo, esse trabalho aborda como esse conceito pode ser aplicado no contexto da fabricação inteligente.
[48] Dias-Ferreira, J., Ribeiro, L., Akillioglu, H., Neves, P., & Onori, M. 2018.	A arquitectura de auto-organização biológica.	Este artigo propõe uma arquitectura de referência bioinspirada para sistemas de produção, focada em ambientes altamente dinâmicos, denominada Arquitectura Auto-Organizável para fabricação inspirada em BIO (BIOSOARM).

[50] Rocha, A. D. Lima-Monteiro, P. Parreira-Rocha, M., & Barata, J. 2019	Baseada em sistemas imunológicos.	Este trabalho propõe uma arquitetura que realize diagnósticos distribuídos, utilizando sistemas imunológicos artificiais para realizar diagnósticos evolutivos.
[39] J. C. Chen, T. Chen, B. R. Pratama, and Q. Tu. 2016	Colônia de formigas.	Propõe-se que a inteligência das colônias de formigas seja combinada com a coordenação de agentes locais, de modo a tornar os agentes autônomos adaptáveis às novas circunstâncias e a dar origem a um desempenho global eficiente.
[40] Y. Song, Y. Guan, 2021.	Skeletal Muscle.	Este estudo é baseado num modelo de rigidez variável dos tendões e do ventre muscular humano, para criar um atuador bio mimético de rigidez variável, ou seja, a rigidez pode ser ajustada infinitamente regulando um cabo de acionamento.
[42] X. Yang and S. Deb, 2011.	Pesquisa da ave cuco.	Pode ser usada para otimizar uma linha de fabricação altamente automatizada permitindo maximizar a utilização das máquinas, enquanto minimiza o capital vinculado.
[43] M. K. Sayadi, R. Ramezani, and N. Ghafari-Nasab, 2010.	Algoritmo Vaga-lume.	É usado para resolver problemas de agendamento de “flow shop” de permutação, ou seja, o objetivo é minimizar o tempo de produção e o tempo total de fluxo e determinar o tamanho ideal do sub lote.

[49] R. Pannequin and A. Thomas, 2012.	Outra interpretação da estigmergia para arquitetura de sistemas orientada a produtos	Este artigo propõe uma arquitetura de sistema product-driven baseada numa interpretação particular do conceito de estigmergia, onde a cooperação entre atores da produção é alcançada graças a atributos (feromonas informacionais) anexados aos produtos.
--	--	--

Os diferentes artigos mencionados na Tabela 2.1 exploram a ideia da bioinspiração aplicadas em diversos contextos. Em todo o caso, temos a base da definição de bioinspiração, explorar a natureza para criar tecnologias [22] que lidam com desafios específicos na fabricação em particular.

Principais desafios e problemas em comuns

Como visto anteriormente, através da tabela e de outros conceitos apresentados, os sistemas de fabricação modernos aplicam claramente os conceitos bioinspirados para alcançar maior produtividade e eficiência na produção. Alguns desses conceitos revelam-se mais desejáveis no contexto da produção inteligente, como: adaptação, evolução, auto-organização e conceitos emergentes. Curiosamente, estes são também os conceitos fundamentais que sustentam o sucesso dos sistemas biológicos [44].

Embora cada abordagem explorada pelas investigações tenha um objetivo distinto, é possível identificar alguns pontos em comum relacionados aos desafios e problemas enfrentados até se alcançar uma conclusão plausível e aceitável, que permitisse a sua aplicação em termos práticos. Alguns desses problemas podem incluir o seguinte:

- Muitos dos conceitos estruturais bioinspirados devem ser detalhadamente gerenciados por uma estrutura técnica robusta, apesar do apoio tecnológico e distribuído obtido dos sistemas mecatrônicos modulares.
- Os requisitos de produção que geralmente ficam centralizados em entidades únicas, devem ser descentralizados e distribuídos entre os elementos que fazem parte do sistema de produção, com o objetivo de diminuir as especificações necessárias e, ao mesmo tempo, incentivar a otimização e a execução distribuída dos fluxos de trabalho de produção.
- A escalabilidade, o poder reconfigurável e a execução coerente e distribuída dos planos de produção devem derivar de padrões de interação eficientes e semânticos que permitam a implementação de princípios regulatórios de auto-organização bioinspirados. Apesar da viabilidade das transições de estado não seja tão crucial para a

execução normal dos algoritmos bioinspirados (contanto que a viabilidade das soluções seja assegurada), ela é crítica para sistemas que obedecem holisticamente esses mesmos princípios. Cada transição pode impactar diretamente o chão de fábrica físico e como resultado os princípios regulatórios concebidos necessitam garantir a viabilidade da transição. Assim, o sistema poderá ser capaz de emular de perto a robustez e a adaptação dos sistemas biológicos. Ao contrário dos sistemas naturais, os ciclos evolutivos dos sistemas de produção têm de ser proporcionais aos prazos de produção muito exigentes [45].

Os desafios e problemas em comuns explorados mostram tendências gerais que podem ser encontradas na busca por inspirações da vida para o bem e evolução das indústrias.

ARQUITECTURA

Neste capítulo é apresentado os aspetos mais direccionados ao altruísmo entre os robôs. É descrito a sua estrutura básica do seu funcionamento e a integração com o Ros2.

3.1 Altruísmo no processo de fabricação

No decorrer dos pontos mencionados no capítulo 2, tornou-se evidente que estamos efectivamente na transição da indústria 3.0 para a indústria 4.0. Como resultado, alguns dos temas mais recorrentes desta nova fase de industrialização incluem a auto-organização e a autoajuda entre os elementos que compõem o sistema de fabrico inteligente. Um dos pilares fundamentais da indústria 4.0 reforça o conceito da autonomia dos robôs, dotados da capacidade de perceber o ambiente e executar determinadas tarefas sem a supervisão direta de especialistas [8], é essencial que o nível de cooperação e organização aumente bastante.

De facto, existem diversas pesquisas que abordam esses temas e agregam um elevado valor e conhecimento para a nova era industrial. Estes estudos contribuem significativamente para o avanço das tecnologias e para um mundo cada vez mais moderno. Contudo, existe uma escassez de trabalhos que relatam a aplicação prática ou simulação destes temas, particularmente sobre altruísmo ou autoajuda entre os robôs dentro do ambiente colaborativo.

Sendo assim, neste capítulo é apresentado da forma mais clara possível como podemos aplicar o altruísmo dentro do ambiente colaborativo industrial nos robôs, de modo a alcançar maior eficiência no processo de produção ou serviços na fabricação. No contexto deste trabalho, o altruísmo aplicado entre os robôs constitui um conjunto simples de acções que visa ajudar o robô que estiver a necessitar de alguma ajuda, com base em alguns critérios apresentados posteriormente. Na Fig. 3.1 é apresentada a definição básica do que consideramos ser “altruísmo no ambiente colaborativo”.

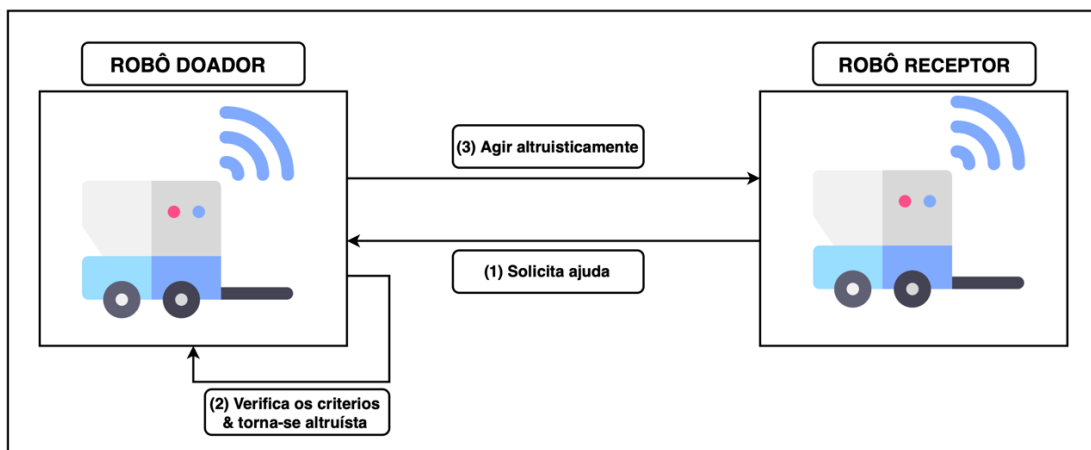


Figura 3.1 -Representação do altruísmo entre os robôs.

Conforme vê-se na imagem acima que representa a definição básica do altruísmo, temos dois elementos que fazem parte do sistema:

- **Robô Receptor:** trata-se do robô que precisa de alguma ajuda dentro do ambiente colaborativo.
- **Robô Doador:** trata-se do robô que possui os critérios para poder agir de forma altruísta e ajudar o outro robô.

A Fig. 3.1, apresenta três (3) ações necessárias para a realização do processo:

1. **Solicitação de ajuda:** é a ação aplicada pelo robô que precisa de alguma ajuda solicitando ao(s) robô(s) no ambiente colaborativo.
2. **Verificação dos critérios e tornar-se altruísta:** é uma ação realizada pelo(s) robô(s) que recebe(m) a solicitação de ajuda, verificando os critérios necessários para ajudar o outro robô, e caso sejam cumpridos os requisitos necessários, este robô torna-se o altruísta. Para o caso acima, considerou-se que os requisitos foram preenchidos por este robô.
3. **Agir altruisticamente:** após preencher os requisitos e tornar-se altruísta ou doador como descrito na figura acima, essa é a ação que o robô executa para poder ajudar o outro considerada como ação altruística.

3.2 Arquitectura geral

Descrito anteriormente, o ROS2 significa Sistema Operativo para Robôs versão 2 (Robot Operating System 2), e tal como o seu nome sugere, é um framework construído para o desenvolvimento de software para robôs, fornecendo maior segurança, middleware, drivers, ferramentas e uma variedade de algoritmos usados nos robôs. Na Fig. 3.2, podemos ver cada uma das camadas que formam a arquitectura do ROS2.

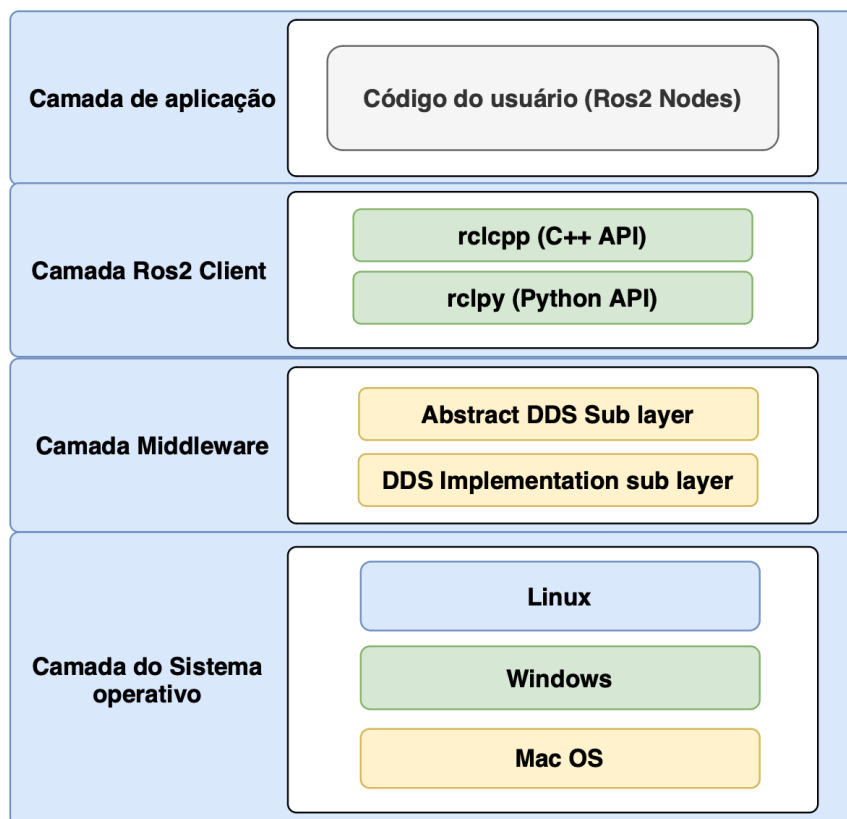


Figura 3.2 - Arquitectura do Ros2.

Cada uma das camadas da arquitectura do ROS2 desempenha um papel importante, o que o torna um framework essencial e fácil de usar para o desenvolvimento de software para robôs. Na Tabela 3.1, é descrita a função de cada um dos elementos desta arquitectura.

Tabela 3.1 - Descrição da arquitectura do Ros2

Camada	Descrição	Subcamadas e recursos
Aplicação	São códigos desenvolvidos pelo usuário para atingir um determinado objetivo dentro do Ros2, acessando recursos como câmera, sensores e outros. Esses são considerados Nós.	Sensores de movimento. Câmeras. Algoritmos etc.

Ros2 Client Library (Rcl)	Permite que as aplicações e Nós acessem os recursos do Ros2. O Rcl é escrito em linguagem C++, mas acima dele pode existir uma rcl para cada linguagem.	Rclcpp é uma biblioteca sobre a rcl para a linguagem C++. Rclpy é uma biblioteca sobre a rcl para a linguagem Python. Também existem bibliotecas para Java, Go e outras linguagens, porém não são muito populares.
Middleware	O Ros2 usa o como middleware o DDS que fornece uma comunicação do tipo “publicador-subscritor”, descentralizada e escalável entre as diferentes partes/nós do Ros2 sem a necessidade de um Ros master.	Abstract DDS sub layer é uma subcamada que torna invisível a complexidade do DDS aos olhos do usuário. DDS Implementation sub layer é a subcamada de implementação do DDS e geralmente é usada a Fast-DDS.
Sistema Operativo	Essa é a camada operacional por onde corre o Ros2. Geralmente para desenvolvimento são usadas S.Os como Ubuntu 22.04 Jammy Jellyfish e Windows 10.	Ubuntu 22.04. Windows 10. Mac OS.

Baseado na arquitetura conforme mostrada na Fig. 3.2, é apresentado a arquitetura do altruísmo entre os robôs com ROS2 considerando o funcionamento dos robôs em dois cenários diferentes na Fig. 3.3.

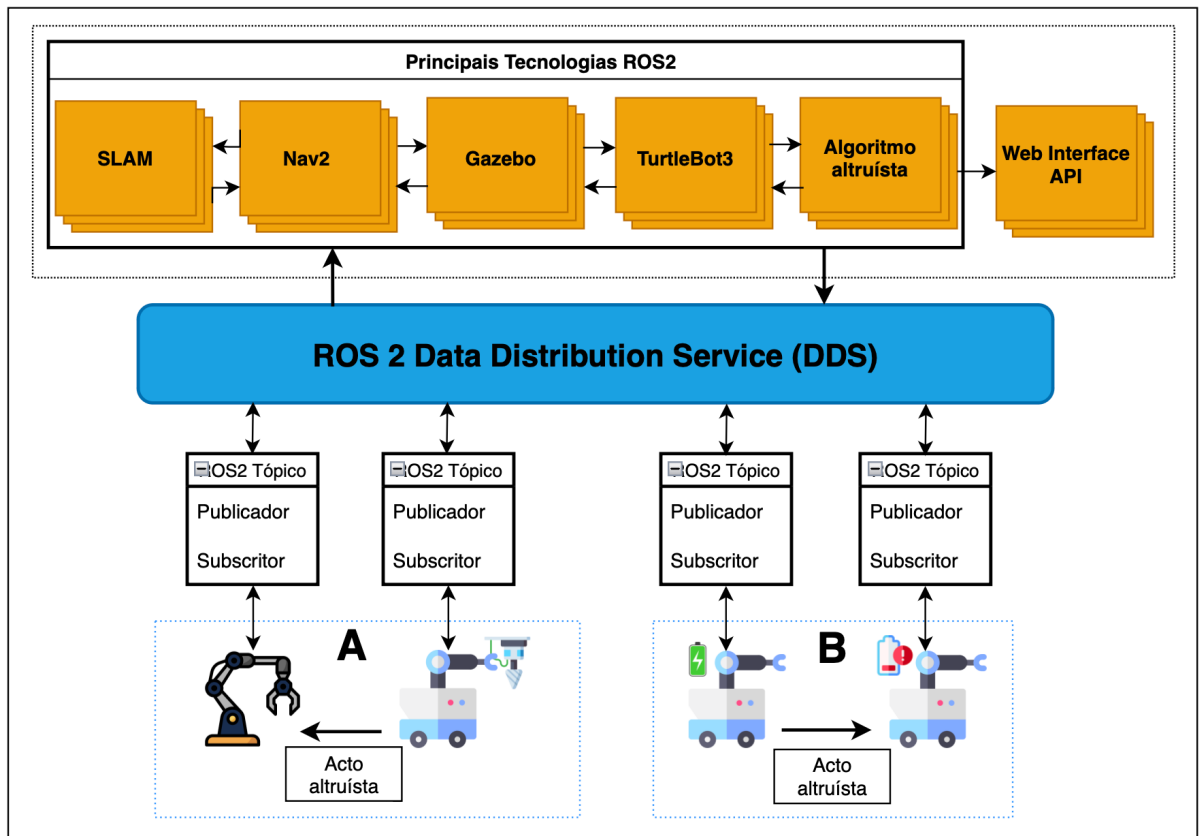


Figura 3.3 - Arquitetura do altruísmo entre os robôs no ros2.

A arquitetura foi separada em 4 partes diferentes para poder tornar mais simples a sua compreensão visto que cada uma delas tem um papel fundamental:

1. Camada física: essa é a camada mais abaixo onde os robôs interagem fisicamente.

Na Fig. 3.3, é possível identificar dois cenários possíveis:

- No cenário **A**, consideramos o altruísmo que pode existir entre um robô móvel autoguiado (AGV) e um manipulador. Nesse caso, sempre que o manipulador estiver a necessitar de uma nova ferramenta ou quando a ferramenta precisa de ser mudada, por exemplo, devido a alterações inesperadas na manutenção, o AGV irá se disponibilizar em achar a ferramenta que se precisa para aquela situação, se deslocar até o manipulador e finalmente trocar as ferramentas.
- No cenário **B**, consideramos o caso em que quando um AGV estiver com a bateria fraca, aquele que for adequado com base nos critérios escolhidos, irá se deslocar para partilhar a sua carga elétrica com o outro AGV.

Os cenários de altruísmo demonstrados acima podem tornar real a autoajuda e melhorar o desempenho do sistema de produção sem a intervenção direta de um profissional por um certo período aumentando a produtividade no sistema de fabricação.

Sendo assim, para a simulação do altruísmo no ambiente colaborativo baseado em ROS2, considerou-se o altruísmo de partilha de carga entre os robôs móveis autoguiados.

2. **Camada de Tópicos:** esta é a camada onde as mensagens são publicadas e sobrescritas pelos nós do ROS2. Tal como descrito anteriormente nos pontos acima, um tópico pode ter vários nós a publicar (Publicadores) e vários a subscrever (Subscritores).
3. **Camada do Middleware:** o ROS2 utiliza como middleware o DDS (Data Distribution Service) para permitir a comunicação entre as diferentes partes do sistema ROS2. É por meio dele que os nós conseguem subscrever e publicar em vários tópicos simultaneamente, permitindo uma comunicação descentralizada e escalável no sistema.
4. **Camada de Tecnologias principais:** os nós são partes do sistema construídas com ROS2 responsáveis por um único propósito ou tarefas específicas, tais como controlar a câmara do nosso robô, controlar as rodas e muito mais. Na arquitectura mostrada acima, vemos as seguintes tecnologias usadas no projecto:
 - **SLAM (Simultaneous Localization and Mapping):** é uma técnica ou tecnologia geralmente usada na robótica para mapear um espaço determinado, onde se encontra cada um dos objetos, as limitações e obstáculos do ambiente. Além de mapear o ambiente, também fornece a posição exata do robô dentro daquele meio. O mapeamento e a localização são feitos simultaneamente enquanto o robô se move, pois, o mapeamento só é realizado graças à atualização ou mudanças constantes da posição anterior do robô.
 - **Nav2:** é uma tecnologia de navegação poderosa usada por robôs móveis e não só. Ela permite a navegação dos robôs em ambientes complexos com o propósito de executar tarefas definidas por um utilizador ou programa. A navegação é o cálculo do caminho ou das rotas que devem ser seguidas para se atingir um destino.
 - **Gazebo:** é uma aplicação de simulação que permite a criação de gráficos 3D, simulações dinâmicas e muito mais. No nosso projeto, o Gazebo será integrado ao ROS2 para permitir a simulação do nosso ambiente virtual, onde o nosso robô irá mover-se para executar a sua tarefa.
 - **TurtleBot3:** Sendo o TurtleBot o robô padrão do ROS(2), que existe tanto em formato físico como de software, o TurtleBot3 é uma versão de robô móvel programável que o ROS2 oferece para a realização de testes, simulações e diversos estudos com robôs, sem a necessidade de gastar muitos recursos e capacidade computacional.
 - **Algoritmo altruísta:** É o algoritmo desenvolvido contendo todas as classes para a implementação do conceito de altruísmo apresentado no presente trabalho.

Web Interface (FastAPI): Usada dentro do módulo das tecnologias do ROS2, a Web Interface é o meio pelo qual o utilizador pode ler as informações que acontecem por trás do altruísmo de forma simples e compreensível. Esta interface é apresentada através de um

navegador web, que por sua vez obtém os dados gerados pelo sistema através da framework web FastAPI.

Visto que o este trabalho apresenta o conceito de altruísmo dentro do ambiente de cooperação, considerou-se que a forma de “Ajuda entre os robôs” pode ser possível em duas aplicações (Fig. 3.3):

1. Permitir a partilha de carga entre os robôs de forma autónoma: A partilha de carga entre os robôs móveis pode ocorrer dentro de um ambiente colaborativo, quando um deles estiver com pouca energia na bateria. Nessa situação, será acionado um dos robôs que possua carga suficiente para partilhar com o outro.
2. Permitir que um robô partilhe uma ferramenta (end effector) entre os robôs: neste contexto, ocorrerá o mesmo processo descrito acima, porém a partilha será acionada quando um dos robôs precisar, por um momento, de uma ferramenta que outro robô possua. Nesse caso, será seleccionado um dos robôs que possua a ferramenta, com base nos critérios mínimos descritos posteriormente.

Em ambas as opções, os robôs agirão de forma altruísta sem a intervenção de um especialista para garantir a continuidade da produção num ritmo eficiente.

3.3 Processo altruístico entre os robôs

No decorrer deste trabalho, o processo altruísta será considerado para o cenário em que um robô se disponibiliza para ajudar outro robô. Essa ajuda consiste em partilhar uma porção da sua energia da bateria (carga) com o outro, tal como é ilustrado na Fig. 3.4 (a). Por outro lado, o mesmo conceito de altruísmo também será demonstrado para a partilha de ferramentas entre os robôs, tal como se observa na Fig. 3.4 (b).

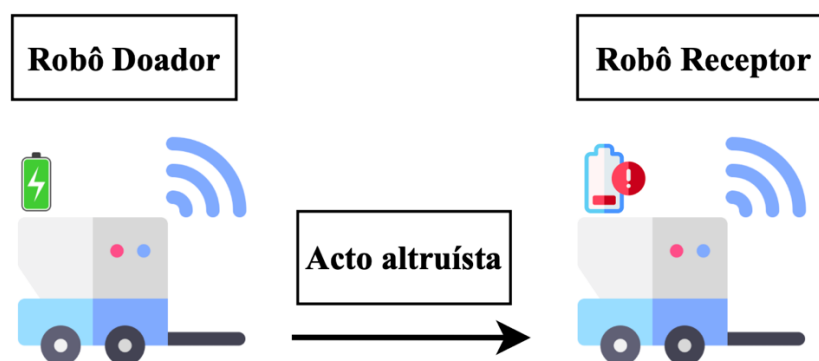


Figura 3.4 – a) Definição do processo altruísta do projeto pela partilha de carga

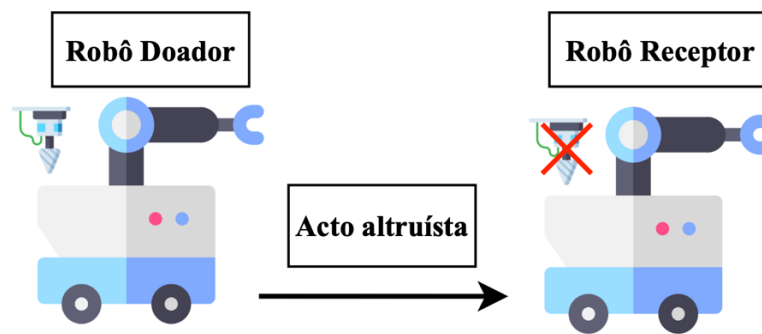


Figura 3.4 – b) Definição do processo altruísta pela partilha de ferramenta.

O processo de altruísmo que acontece entre dois robôs é basicamente constituído por três elementos fundamentais, Fig. 3.4 (a) e 3.4 (b):

1. **Robô Doador:** Considera-se Doador aquele robô que possui carga suficiente ou a ferramenta necessária para partilhar com outro robô. Apenas este robô pode tornar-se altruísta, mas nem todos os Doadores têm a capacidade de o fazer, uma vez que alguns requisitos podem depender do Receptor.
2. **Robô Receptor:** Considera-se Receptor ou solicitante aquele robô que possui um nível baixo de bateria ou necessita de uma ferramenta que outro robô possui, sendo esta a razão pela qual o robô solicita ajuda.
3. **Acto Altruísta:** É a ação de partilha de energia/carga ou ferramenta de um robô para o outro.

IMPLEMENTAÇÃO DO PROJETO

Este capítulo descreve como o projeto foi implementado, explorando cada uma das principais tecnologias usadas, algoritmos e diferentes cenários pelo qual o altruísmo pode ocorrer dentro do ambiente colaborativo.

4.1 Sistema de simulação com ROS2

O ROS2 é o framework adequado para operar e criar aplicações robóticas, sendo capaz de lidar com diversos tipos de desafios específicos no desenvolvimento de serviços em grande escala. Um projeto em ROS2 é composto por elementos que tornam o projeto mais organizado e simples para o processo da sua execução, como se vê na Fig. 4.1. Estes elementos incluem: workspace, diretórios, pacotes, nós e outros ficheiros.

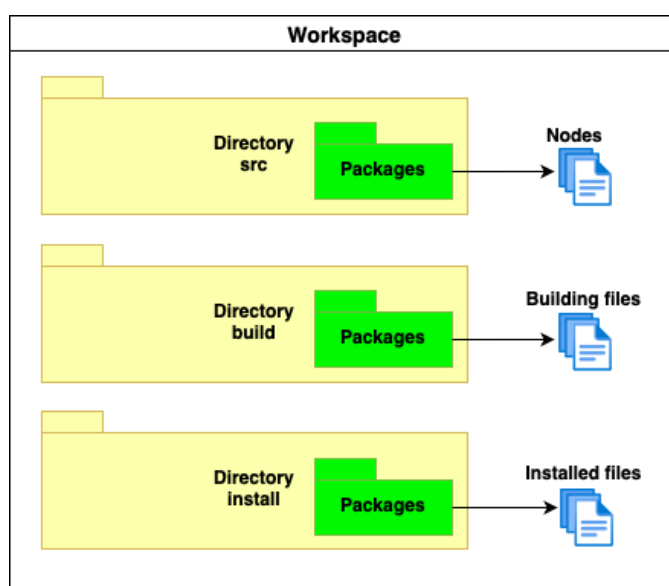


Figura 4.1 - Estrutura do Workspace do ROS2.

Os projetos em ROS2 devem estar separados por workspaces, ou seja, cada um dos projetos deve ter seu próprio workspace.

Sendo assim, o workspace é um diretório que armazena todos os pacotes e ficheiros relacionados a um único projeto de ROS2, permitindo organizar e desenvolver o projeto com a melhor gestão possível. Da mesma forma, foi construído o projeto atual.

Dentro do workspace encontram-se 3 principais diretórios:

- **Src:** o *source*, normalmente escrito em minúsculas como *src*, é o diretório que armazena os pacotes que contêm o código fonte e todos os nós (nodes) necessários para o desenvolvimento completo do projeto. As funcionalidades do robô que são acrescentadas ou desenvolvidas são geralmente projetadas dentro do *source*.
- **Build:** é o diretório utilizado para compilar os pacotes que se encontram no *src*. Os ficheiros deste diretório não são usados durante a execução do projeto, mas sim para gerar bibliotecas e ficheiros executáveis com base no que foi adicionado no *src*.
- **Install:** este é o diretório que armazena o resultado do processo de compilação do *build*, ou seja, todos os pacotes e ficheiros que o *build* gera como resultado da compilação são armazenados no *install*. São esses ficheiros com os quais os nós interagem durante a execução do projeto, armazenando todas as versões finais das bibliotecas, ficheiros de configuração, *launch* e muitos outros.

Desta forma, pode-se afirmar que o *src* contém os "source files" (ficheiros fonte), os quais são compilados no *build*, e o resultado do processo de *build* é armazenado no *install*, que será utilizado durante o *runtime*. Na Fig. 4.2 é apresentada a estrutura real do *workspace* do nosso projeto construído em ROS2.

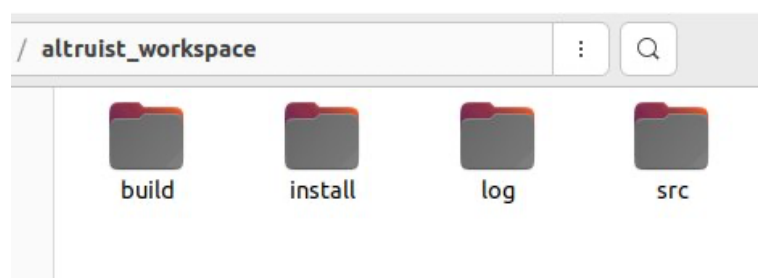


Figura 4.2 - Estrutura do Workspace do projeto.

O nosso *workspace* chama-se "altruist_workspace" e, dentro dele, podemos claramente ver os diretórios mencionados anteriormente, com exceção do *log*.

O diretório *log* armazena os registos gerados durante a execução dos nós e do projeto em si. Estes *logs* podem conter erros, mensagens informativas, avisos e outras mensagens que podem ser úteis para compreender melhor o comportamento do projeto no ROS2

Conforme descrito na Tabela 3.1, o ROS2 para desenvolvimento é geralmente instalado sobre um sistema operativo, podendo ser o Ubuntu 22.04 Jammy Jellyfish, Windows 10 ou Mac OS. Neste projeto, foi utilizado o Ubuntu 22.04.4 LTS, codename Jammy, para a implementação do conceito de altruísmo dentro do ambiente colaborativo.

A construção do sistema para concretizar a simulação do projeto envolveu várias tecnologias incorporadas no ROS2 ou com capacidade de compatibilidade com ele.

4.1.1 Plataforma de robô padrão TurtleBot

O ROS(2) oferece uma plataforma de robô padrão denominada TurtleBot. Atualmente, existem 4 versões/modelos do TurtleBot, sendo que, para o desenvolvimento deste projeto, foi utilizada a versão 3, ou seja, o TurtleBot3.

O TurtleBot3 é um robô móvel baseado em ROS de tamanho reduzido e programável, utilizado para fins educacionais, pesquisa, entre outros. Pode ser utilizado em diferentes projetos, permitindo a adição de elementos extras, como por exemplo, sensores.

A tecnologia principal do TurtleBot3 inclui SLAM, Navegação e Manipulação, o que o torna adequado para robôs que operam em ambientes domésticos. Além disso, o TurtleBot3 apresenta diferentes variantes de robôs móveis, nomeadamente:

- TurtleBot3 Burger.
- TurtleBot3 Waffle.
- TurtleBot3 Waffle Pi.

O TurtleBot3 Waffle, apresentado na Fig. 4.3, é a variante utilizada para a simulação do projeto no ambiente virtual. O Waffle foi projetado para aprendizado e desenvolvimento de aplicações de robótica utilizando ROS1 e 2. O TurtleBot3 é popular por ser acessível, compacto e por fornecer uma plataforma robusta que funciona com o ROS2, tanto em ambientes reais como simulados [52].

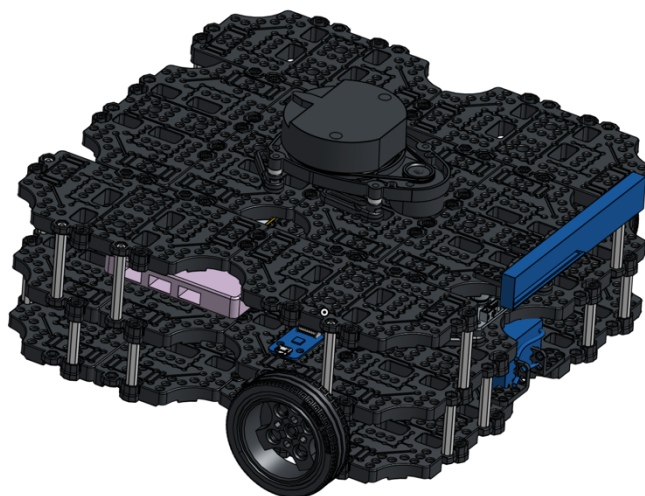


Figura 4.3 - Representação do TurtleBot3 Waffle [52].

O TurtleBot3 pode executar algoritmos SLAM (Simultaneous Localization and Mapping), ou seja, algoritmo de localização e mapeamento simultâneo para construir ou fazer um mapeamento de uma área e pode dirigir sobre ela.

4.1.2 Tecnologia de navegação Nav2

Falando da navegação do robô, que, como apresentado acima, utilizou o modelo Waffle do TurtleBot3, neste projeto foi usado o Nav2 para fornecer ao robô a capacidade de percepção e visualização do ambiente, planejamento de rotas, controle e localização dos pontos de coordenadas no ambiente em que se encontra, tornando-o assim mais autônomo e altamente fiável.

O Nav2 é uma tecnologia de navegação para sistemas ou robôs autônomos que pode ser utilizada com o ROS2. Através dos sensores do robô, o Nav2 permite criar um modelo ambiental, construir rotas dinamicamente para alcançar um determinado ponto-alvo ou posição, mesmo em ambientes dinâmicos. Além disso, possibilita calcular os comandos motores viáveis, evitar obstáculos e estruturar comportamentos de robôs de nível superior, Fig. 4.4.

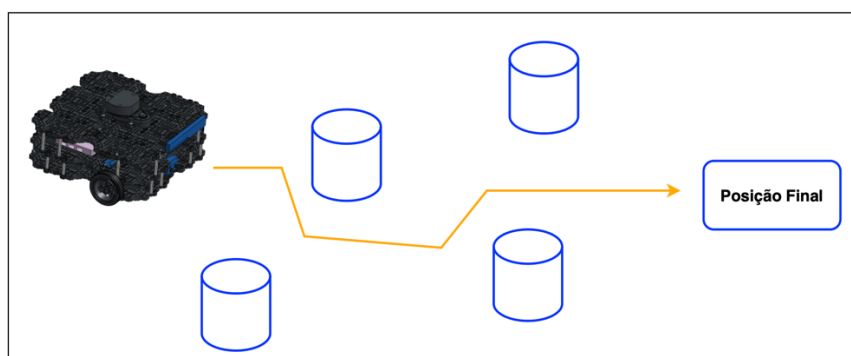


Figura 4.4 - Construção de rota com o Nav2.

O Nav2 usa árvores de comportamento (BT) para criar movimento de navegação personalizado e inteligente por meio da orquestração de muitos servidores modulares independentes. Tal como mostra a Fig. 4.5, esses servidores independentes como a camada de *costmap*, planejador (*planner*), controlador (*controller*), BT e outros permitem que se crie aplicativos próprios ou algoritmos personalizados.

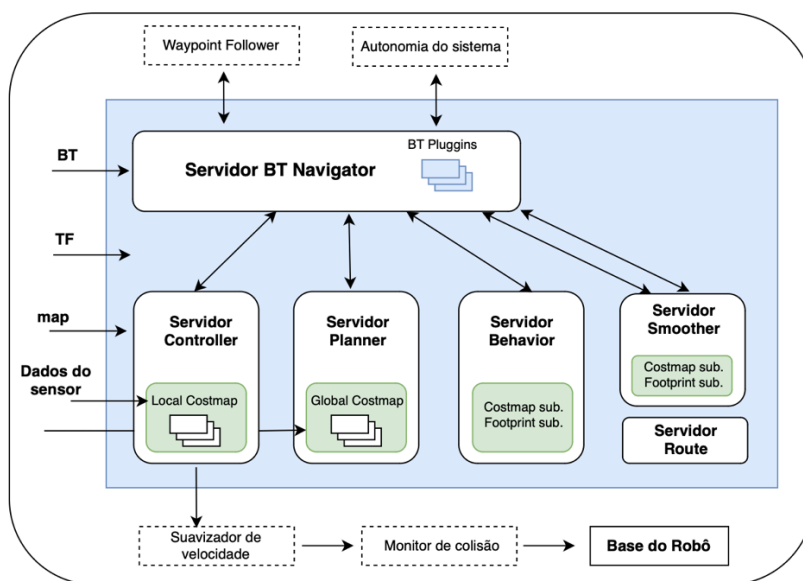


Figura 4.5 - Arquitectura do Nav2 [53].

O Servidor BT Navigator é o ponto de entrada de mais alto nível, integrando a Árvore de Comportamento (BT) para implementar comportamentos de navegação. Baseado no Waypoint Follower, que contém o destino ou as posições a serem alcançadas, ele orquestra a tarefa de navegação com sua Árvore de Comportamento para atingir ou alcançar essas metas.

A Árvore de Comportamento pode utilizar servidores de ação no controlador, planejador, comportamento e no servidor de suavização (smoother) para lidar com o controle de esforços, planos, recuperações e muitos outros aspetos. Todos os servidores representados na

Fig. 4.5 são geridos de forma a utilizarem o ciclo de vida (Lifecycle). O Lifecycle Manager permite coordenar o ciclo de vida do programa do BT Navigator e dos nós desses servidores, que, conforme o seu estado ou ciclo de vida, pode estar: ativo, inativo ou finalizado.

Esses servidores separados se comunicam com a árvore de comportamento (BT) por meio da interface do ROS2. Um robô pode utilizar potencialmente muitas árvores de comportamento diferentes para permitir que um robô execute muitos tipos de tarefas únicas e complexas.

4.1.3 Linguagem de programação Python

No desenvolvimento deste trabalho, utilizou-se a linguagem Python para a criação de todos os nós (nodes) necessários para a implementação do altruísmo entre os robôs. O Python é uma das linguagens mais utilizadas atualmente e oferece um vasto conjunto de bibliotecas, o que torna a linguagem mais acessível e aplicável em diferentes tipos de projetos. Além de possibilitar a criação das classes para a execução do altruísmo, também permitiu o desenvolvimento de uma interface baseada no framework FastAPI.

FastAPI

Trata-se de um framework web moderno, rápido e eficiente para construir APIs em Python. Foi projetado para facilitar a criação de APIs RESTful de forma simples e intuitiva, oferecendo alto desempenho. O seu papel neste projeto é permitir que os dados sejam enviados para uma interface criada com tecnologias como HTML, CSS e JavaScript.

4.1.4 Simulador Gazebo

O presente projeto foca-se na simulação da implementação do conceito de altruísmo dentro do ambiente colaborativo, como já foi descrito no capítulo anterior. Sendo assim, tratando-se de uma simulação, o framework utilizado para simular o projeto foi o Gazebo.

O Gazebo é uma aplicação que permite simular ambientes reais e é geralmente utilizado em conjunto com o ROS ou ROS2 para dar forma (virtual) às necessidades de muitos desenvolvedores durante a implementação ou construção de um projeto. Ele oferece uma caixa de ferramentas completa, com bibliotecas necessárias para o desenvolvimento e serviços em nuvem para facilitar a simulação. A sua integração com ambas as versões do ROS é feita através de um conjunto de pacotes chamado `gazebo_ros_pkgs`, que fornece uma ponte entre a API C++ e o sistema de transporte do Gazebo, incluindo mensagens e serviços do ROS2. O ROS 2 `gazebo_ros_pkgs` é um conjunto de pacotes (metapacote) que oferece uma variedade de configurações e serviços agrupados pelos seguintes pacotes:

- **gazebo_msgs:** fornece as estruturas de dados de mensagens e serviços para interagir com o Gazebo do ROS2.
- **gazebo_dev:** fornece uma configuração cmake para a versão padrão do Gazebo para a distribuição ROS. Isso significa que os pacotes *downstream* podem depender apenas do gazebo_dev em vez de precisar encontrar o Gazebo sozinhos.
- **gazebo_ros:** fornece classes e funções C++ convenientes que podem ser usadas por outros plug-ins, como gazebo_ros::Node, utilitários de conversão e teste.
- **gazebo_plugins:** é uma série de plug-ins Gazebo que expõem sensores e outros recursos ao ROS2. Por exemplo, gazebo_ros_camera publica imagens ROS 2 e gazebo_ros_diff_drive fornece uma interface para controlar e inspecionar robôs de acionamento diferencial por meio do ROS2.

O Gazebo é uma das peças fundamentais para a realização da simulação do presente projeto devido à sua capacidade de integração com o ROS2.

Gazebo e turtleBot3

O Gazebo e o TurtleBot3 estão unidos por um conjunto de pacotes e ficheiros que podem ser executados em tempo real. No presente projeto, foi utilizado o ambiente do Gazebo TurtleBot3 World, apresentado na Fig. 4.6, onde o robô (TurtleBot3) estará localizado. Este é um ambiente pré-configurado do TurtleBot e do Gazebo, cuja integração com o ROS2 permite uma implementação rápida e eficiente.

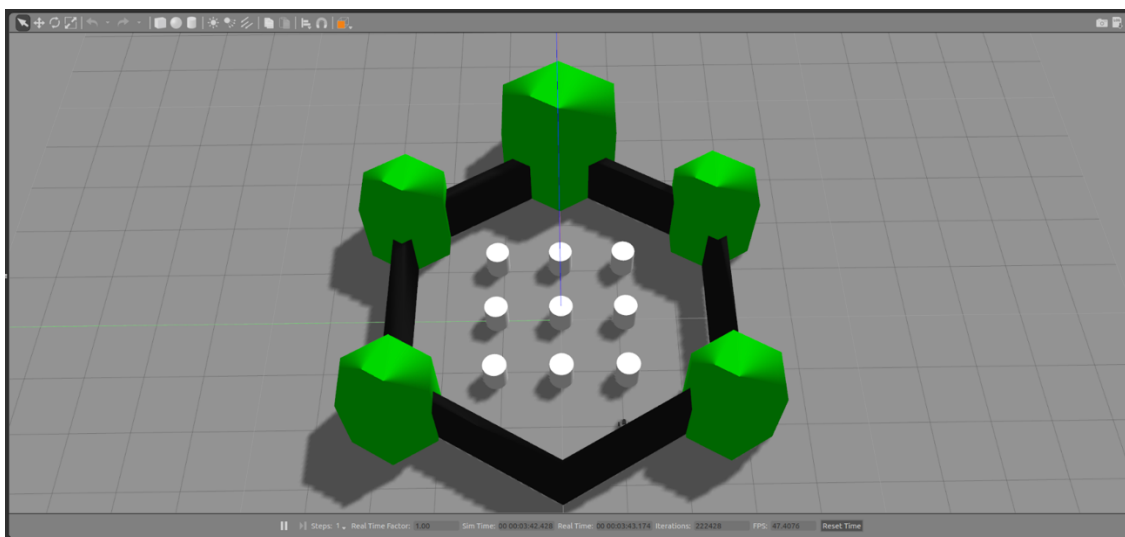


Figura 4.6 - TurtleBot3 World para o turtleBot3.

O Gazebo e o TurtleBot3 estão unidos por um conjunto de pacotes e ficheiros que podem ser executados em tempo real. No presente projeto, foi utilizado o ambiente do Gazebo TurtleBot3 World, apresentado na Fig. 4.6, onde o robô (TurtleBot3) estará localizado. Este é um ambiente pré-configurado do TurtleBot e do Gazebo, cuja integração com o ROS2 permite uma implementação rápida e eficiente.

Por outro lado, temos a representação do robô TurtleBot3 modelo Waffle no Gazebo, apresentada na Fig. 4.7. O robô está equipado com sensores a laser, que lhe permitem mapear a área e realizar a auto-localização à medida que se move, facilitando a navegação e o reconhecimento de obstáculos no ambiente.

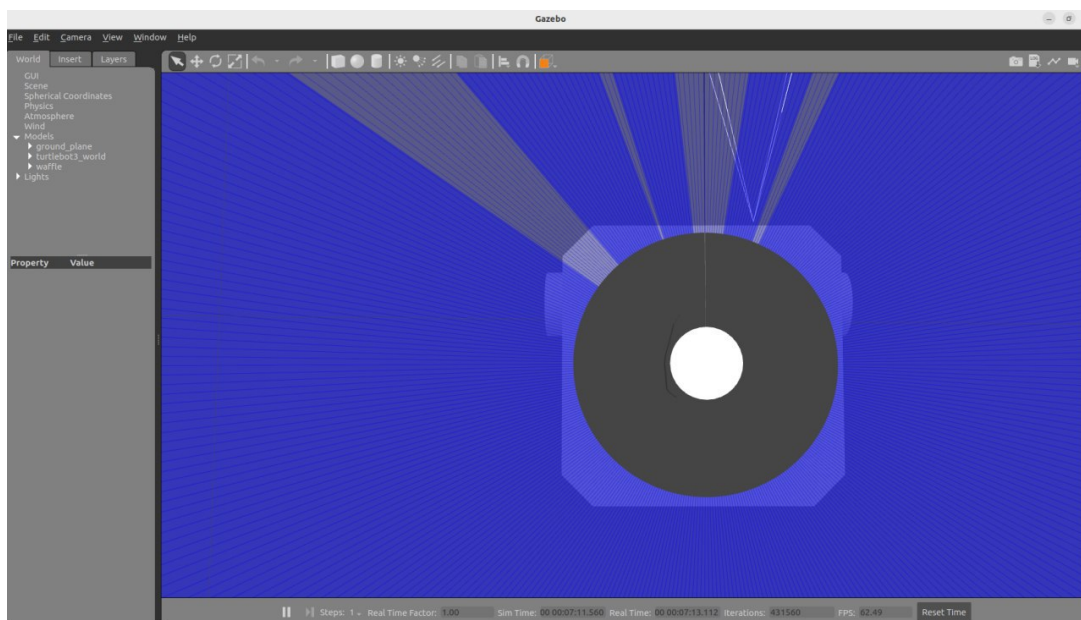


Figura 4.7 - Representação do TurtleBot3 no Gazebo.

Nessa linha de pensamento, na Fig. 4.8, temos o TurtleBot3 dentro do ambiente TurtleBot3 World. Os campos preenchidos com cores diferentes indicam o campo de visão do robô, onde o azul escuro indica que o sensor ou laser do robô atingiu o limite do espaço, como uma parede ou obstáculo. Já a cor azul-claro indica que o robô ainda não reconheceu o limite, ou seja, ele desconhece o que está à sua frente. Nesse caso, o robô precisaria se aproximar até que consiga identificar a parede ou o obstáculo próximo, determinando assim o limite do seu campo de visão.

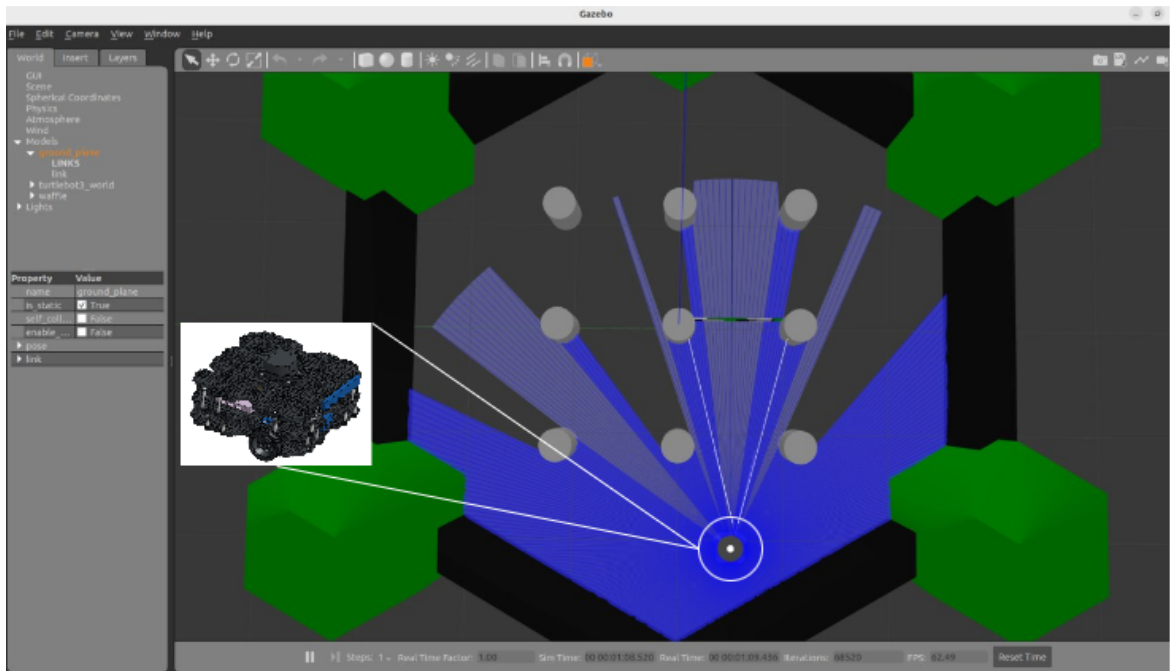


Figura 4.8 - Representação do TurtleBot3 dentro do Gazebo turtleBot3 world.

4.2 Cenário de aplicação do altruísmo entre os robôs

4.2.1 Parâmetros e condições para tornar-se altruísta

Considera-se que todos os robôs móveis dentro do ambiente colaborativo possuem os seguintes parâmetros: *carga ou energia* (p - power), *tempo de espera para executar sua tarefa* (t) e *distância que se encontra em relação a um determinado robô solicitante* (d).

Diante destes parâmetros, um Robô Doador se torna altruísta se:

- **Ter carga suficiente ($p > 20\%$):** o factor p é o primeiro critério para ser considerado doador, pois permite separar os robôs doadores dos receptores. Com isso, estamos a considerar que a carga real (p) do robô é superior à carga a ser partilhada. Fez-se a suposição de que a carga da bateria de 20% é o valor mínimo aceitável para que um robô possa partilhar com outro; abaixo de 20%, a carga é considerada de urgência e, desta forma, o robô não pode partilhar, apenas solicitar ajuda. Neste caso, temos a equação (eq.1) que diz,

$$\text{Se, } p > 20\%: \quad (1)$$

$$\text{robot} = \text{Doador}$$

onde,

p : é a carga energética da bateria do robô;

robot : é qualquer robô móvel disponível no ambiente colaborativo;

Doador : é a denominação para os robôs que podem ajudar.

- **Estar em estado de espera (t):** embora os robôs estejam constantemente em movimento, em algumas situações podem não executar nenhuma tarefa, o que os faz estar em estado de espera. Assim sendo, no momento que surge uma solicitação, o robô não deve estar a executar nenhuma tarefa e o *tempo de espera (t) deve ser superior ao tempo (t_{ajuda}) que leva para ir agir altruisticamente até voltar a sua posição inicial somando o tempo que leva para partilhar a carga ($t_{partilha}$)*. Tal como se vê na equação 2, temos que,

$$\text{Se, } t > 2 * t_{ajuda} + t_{partilha} \quad (2)$$

Robot = Doador_Altruista

onde,

t : tempo em estado de espera;

t_{ajuda} : tempo que leva até o robô solicitante. Multiplicando por “2” significa que estamos a considerar o tempo de ida e volta e está diretamente relacionada à distância entre os robôs.

$T_{partilha}$: tempo que leva para partilhar a carga com o outro robô.

Robot: é qualquer robô móvel disponível no ambiente colaborativo;

Doador_altruista: é a denominação para os robôs que cumprem a primeira e segunda condição apresentadas acima.

- **Estar próximo do solicitante:** esta condição permite tornar ou escolher como altruísta aquele robô que estiver mais próximo do solicitante, caso haja mais de 1 candidato. Ela influencia diretamente o tempo de ajuda (t_{ajuda}) (eq.3), visto que a velocidade dos AGV's pode variar entre 0.8m/s a 2m/s, **vamos considerar 1m/s**. Neste caso teremos,

$$d = t * v \text{ (Lei do espaço do movimento retilíneo uniforme),}$$

$$v = 1m/s, t = t_{ajuda}$$

$$d = t_{ajuda} * 1,$$

Logo,

$$d = t_{ajuda} \quad (3)$$

Onde, v : velocidade do robô móvel; d : distância até o robô solicitante.

Desta forma, para que um robô se torne finalmente altruísta deve cumprir as três condições acima apresentadas que matematicamente seria a união da eq.1, eq.2 e eq.3:

$$\text{Se, } p > 20\% \text{ e } t > 2 * d + t_{ajuda} \quad (4)$$

Robot = Altruista

Caso haja mais de um robô disponível para ajudar um mesmo solicitante, aquele que tiver o menor “ d (eq.3)” será selecionado como altruísta, visto que quanto menor o “ d ”, mais rápido será realizada a ação altruística de modo a impedir que uma estação de produção pare por muito tempo.

Uma vez selecionado o altruísta, considerou-se que pode partilhar a carga da bateria com base no seu próprio nível de energia (bateria) disponível:

- robô altruísta com carga entre 20 e 30% doa 10%;
- robô altruísta com carga entre 30 e 40% doa 15%;
- robô altruísta com carga entre 40 e 50% doa 20%;
- robô altruísta com carga superior e 50% doa 25%.

Nota: é importante realçar que, embora não tenha sido considerado na simulação, o robô também pode partilhar uma certa quantidade de carga de forma dinâmica. Neste caso, o solicitante pode pedir ajuda, indicando a quantidade de energia de que precisa, com base no tempo que a sua tarefa irá durar. Quanto maior for o tempo da tarefa, maior será a energia necessária. Assim, apenas responderão os doadores que possuam uma quantidade de energia superior à solicitada.

Os critérios acima são aplicados para a condição de haver altruísmo baseado na partilha de carga da bateria do entre os robôs dentro do ambiente colaborativo.

A aplicação destas condições dentro do projeto é feita através da função denominada *calculate_time_from_dist*. A função recebe a lista de robôs *doadores* (*robotList*), a posição que se deseja alcançar (*target_pose*) e o robô receptor. Tal como se mostra o algoritmo 1 abaixo, na alínea 3 é feita uma subtração entre o tempo que o robô possui (*robotList[i].time_*) e o tempo que leva para percorrer a distância quer na ida e na volta (*dist* multiplicado por 2) (*Eq.8*) e o resultado é armazenado na variável *help_time* que servirá para como ponto de verificação para o cumprimento das condições básicas do altruísmo nesse contexto.

Algorithm 1: Function - calculate_time_from_dist

Input: *robotList, target_pose, robot*
Output: *robots_with_time, robots_hasPower_noTime, messages*

```

...
1  for i --> 1 to robotList do
2      ...
3      help_time = robotList[i].time_ - 2 * dist
4      ...
5      if (help_time > 0) do
6          robots_with_time.append(robotList[i])
7          ...
8      end if
9      if (not help_time > 0) and (robotList[i].power > 20) do
10         ...
11         robots_hasPower_noTime.append(robotList[i])
12     end if
13 end for

```

A função retorna uma lista contendo os robôs que possuem tempo suficiente para ajudar (*robots_with_time*), robôs *doadors* sem tempo suficiente (*robots_hasPower_noTime*) e mensagens de informação do seu funcionamento (*messages*).

Considerando que o altruísmo seja baseado na partilha de ferramenta, o critério de verificação da carga seria substituído por uma verificação da posse da ferramenta solicitada:

1. **Ter a ferramenta solicitada ($f==1$):** para ser considerado um robô Doador, a condição inicial é possuir a ferramenta solicitada por um dos robôs, tal como se vê na Eq.5.

$$\begin{aligned} \text{Se, } f == 1: \\ \text{robot} = \text{Doador} \end{aligned} \quad (5)$$

onde

f : é a variável que representa a ferramenta, se for “1”, o AGV possui, se for “0”, então o AGV não possui a ferramenta solicitada.

2. **Estar em estado de espera (t):** nesta condição, mantém-se os parâmetros que na Eq.2, porém o tempo de partilha vai se referir ao tempo que vai levar até ir ao outro robô e esperar até que a ferramenta seja devolvida.
3. **Estar próximo do solicitante:** E finalmente teríamos uma adaptação da última condição, Eq.4, que resultaria da Eq.6 abaixo:

$$\begin{aligned} \text{Se, } f == 1 \text{ e } t > 2 * d + t_{partilha}: \\ \text{Robot} = \text{Altruista} \end{aligned} \quad (6)$$

4.2.2 Cenários de altruísmos entre os robôs

Na Fig. 3.3, vemos cada uma das camadas que formam a arquitetura do ROS2. Desta forma, para a concretização do objetivo do projeto “Altruísmo dentro do ambiente colaborativo com o foco na partilha de bateria ou carga elétrica entre os robôs móveis autoguiados (AGV)”, a principal tarefa a ser realizada é:

- **Processo de seleção do robô altruísta.**

Os AGVs dentro do ambiente colaborativo podem executar diversas funções por longos períodos e, em alguns casos, podem ficar parados por curtos períodos, dependendo da tarefa a ser executada. Durante esse processo, é possível que um ou mais AGVs fiquem com pouca carga na bateria. Nesse caso, pode ser selecionado um AGV capaz de agir de forma altruísta, com o objetivo de partilhar a sua carga com os outros sempre que possível, garantindo assim a continuidade da produção de forma eficiente. Essa seleção é feita com base em critérios como carga, distância e o tempo necessário para a realização da ação altruísta.

O processo altruísta pode ocorrer em cenários diferentes, dentre eles pode-se mencionar de “Um para Um” na Fig. 4.9, “Um para Muitos” na Fig. 4.11, “Muitos para Um” na Fig. 4.13 e “Muitos para Muitos” na Fig. 4.10, abaixo apresento cada uma das situações ou cenários:

- **Cenário 1 - Um para Um / Single-mode:** neste cenário, a ação altruísta é realizada nas condições em que existe um robô solicitando ajuda e um único robô disponível para fornecer apoio. Devido à sua simplicidade, os robôs comunicam-se diretamente entre si. Em todos os cenários, o solicitante envia os seus dados, como a posição e a carga, para o robô doador, que então decide se pode ou não oferecer assistência com base nas informações recebidas.

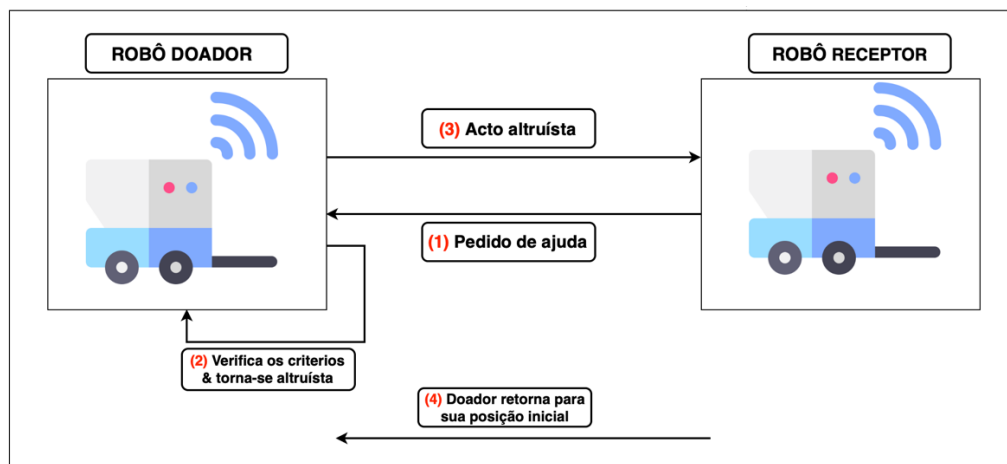


Figura 4.9 - Cenário Um para Um / Single-mode do altruísmo entre os robôs.

O cenário 1 representado na Fig. 4.9 possui um robô solicitador de ajuda e um robô disponível para ajudar, isso torna-o mais simples sem a necessidade de um terceiro elemento para ajudar na tomada de decisão. Na Fig. 4.10 vemos o algoritmo de seleção do robô altruísta.

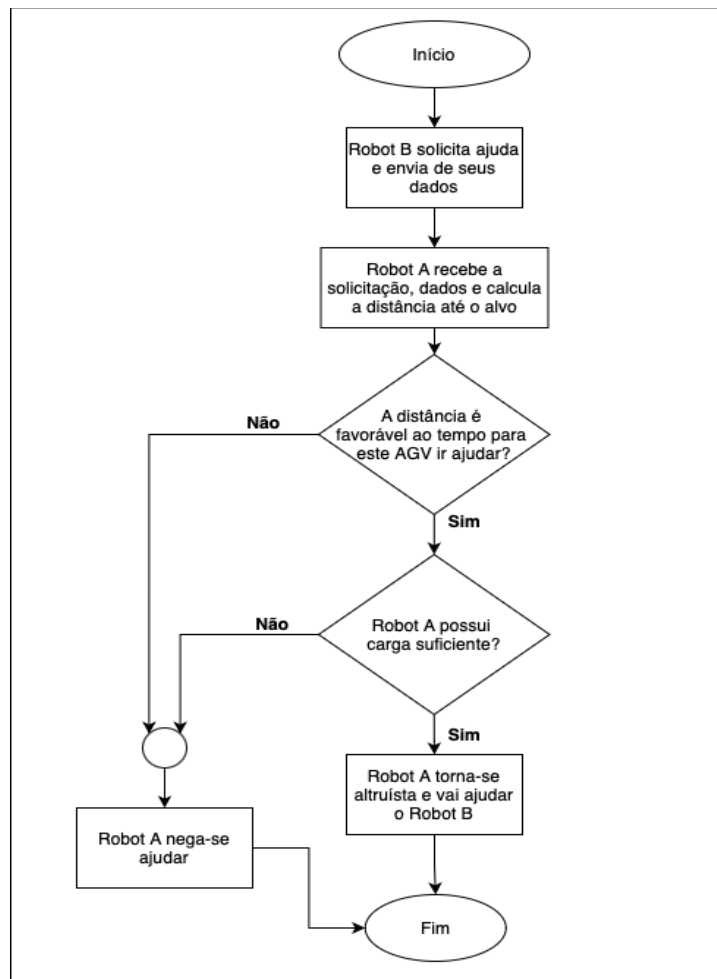


Figura 4.10 - Algoritmo do altruísmo no cenário Um para Um / Single-mode.

- **Cenário 2 - Um para Muitos / Single-multi mode:** neste cenário, a ação altruísta é realizada nas condições em que existem vários robôs a solicitar ajuda e um único robô disponível para fornecer apoio. Os critérios de selecção que permitem a um robô escolher qual vai ajudar são: distância, tempo e carga.

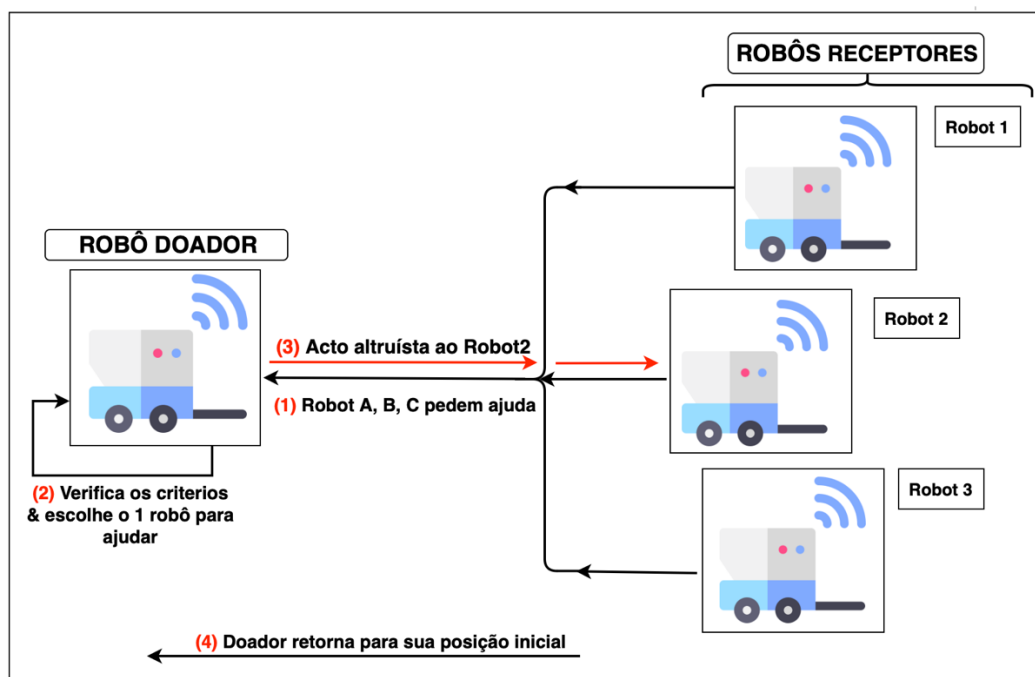


Figura 4.11 - Cenário Um para Muitos / Single-multi mode do altruísmo entre os robôs.

Nesse cenário representado na Fig. 4.11, o facto de seleção é mais criterioso em relação ao cenário 1 uma vez que temos vários robôs solicitantes e apenas um robô disponível para ajudar, o que torna o algoritmo mais complexo. Na Fig. 4.12, vemos a representação do algoritmo que permite que o altruísmo aconteça no cenário acima.

Este algoritmo, por permitir a recepção de múltiplas solicitações, permite que o robô que se torna altruísta tenha a possibilidade de ajudar mais de um robô. Isso pode acontecer caso o robô possua carga suficiente para apoiar os outros robôs solicitantes após realizar a sua primeira acção altruísta. Reforça-se que, depois da primeira acção altruísta, esse robô já terá uma nova carga (decrementada por partilhar com o outro robô) e o seu tempo de disponibilidade também será reduzido. São essas novas informações que poderão ser usadas para fazer uma segunda verificação com a possibilidade de agir novamente de forma altruísta com outro robô na sua área, tal como vemos na Fig. 4.12.

Visto que há mais de um robô solicitante, o robô doador irá seleccionar aquele que estiver mais próximo de si para economizar tempo ou aquele solicitante cuja tarefa possui um custo maior em relação aos outros.

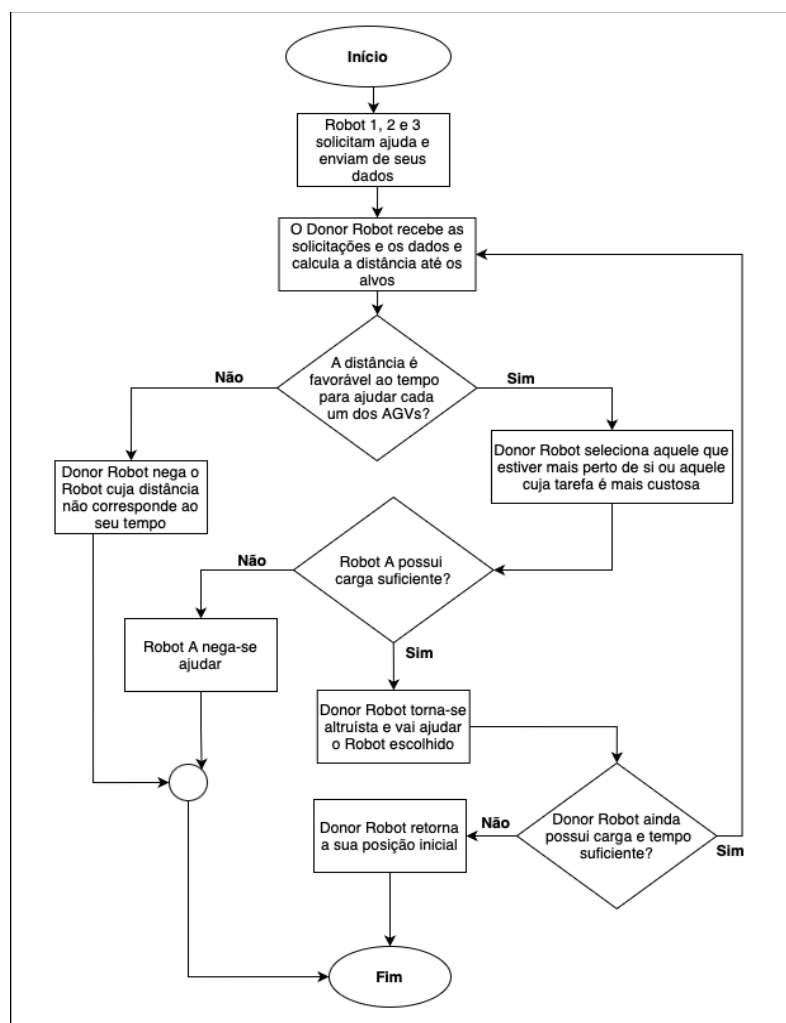


Figura 4.12 - Algoritmo do altruísmo no cenário Um para Muitos / Single-multi mode.

- Cenário 3 - Muitos para Um / Multi-single mode:** neste cenário, conforme mostra a Fig. 4.12, a ação altruísta é realizada nas condições em que existem vários robôs disponíveis para fornecer ajuda e um único robô solicitando apoio. Nesta situação, será seleccionado aquele que se encontra mais próximo (menor distância), o que fará com que o robô leve menos tempo de espera e tenha carga suficiente. Por existirem vários robôs disponíveis para ajudar, entra em cena um elemento intermediário (servidor) que irá comparar as distâncias de cada um dos robôs e seleccionar aquele que tiver a menor. O algoritmo deste cenário pode ser visualizado na Fig. 4.13, onde vemos a sequência de cada uma das acções que são executadas para se atingir o objectivo altruísta. Seu algoritmo encontra-se representado no Fig. 14.

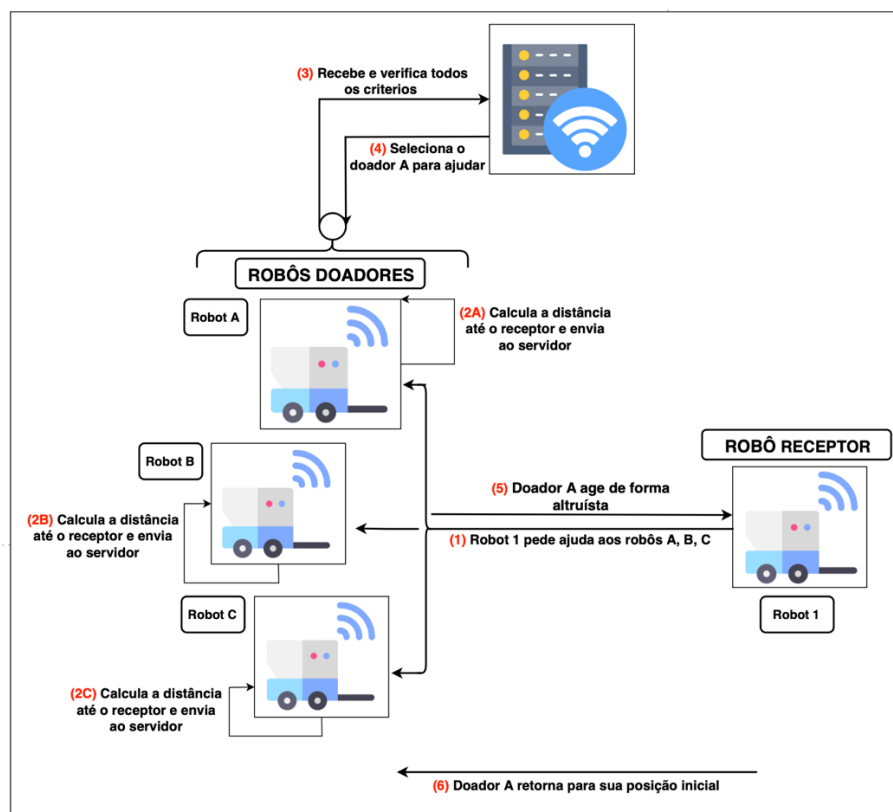


Figura 4.13 - Cenário Muitos para Um / Multi-single mode do altruísmo entre os robôs.

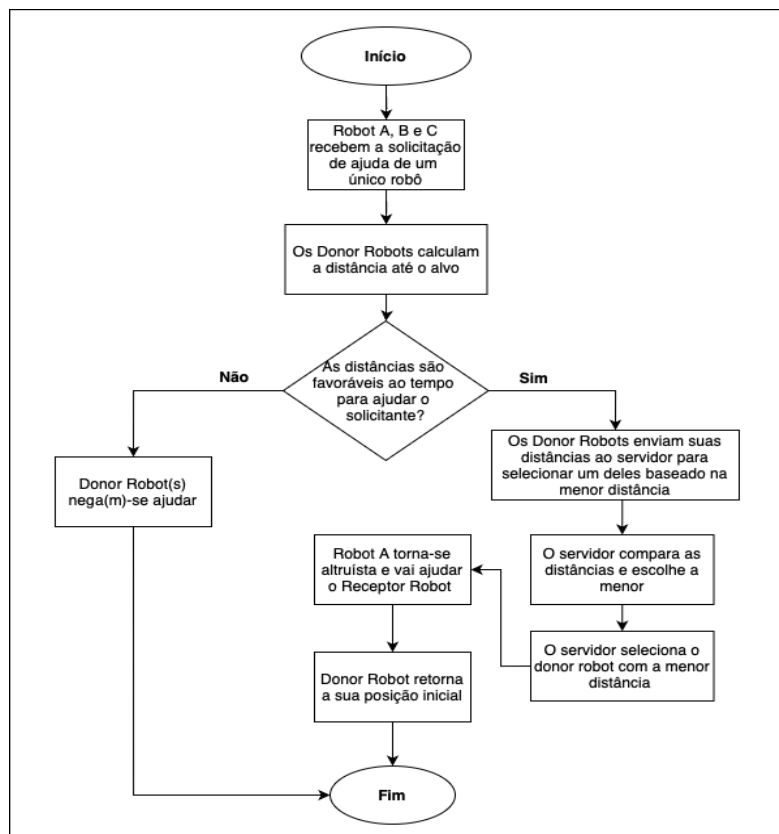


Figura 4.14 - Algoritmo do altruísmo no cenário Muitos para Um/ Multi-single mode

Cenário 4 - Muitos para muitos / Multi-mode: neste cenário, conforme mostrado na Fig. 4.15, a ação altruísta é realizada nas condições em que existem vários robôs disponíveis para fornecer ajuda e um grupo de robôs solicitando apoio. Nesta situação, será seleccionado um robô para cada outro que solicitar ajuda, baseando-se nos mesmos critérios anteriormente mencionados. O seu algoritmo pode ser visualizado na Fig. 4.16.

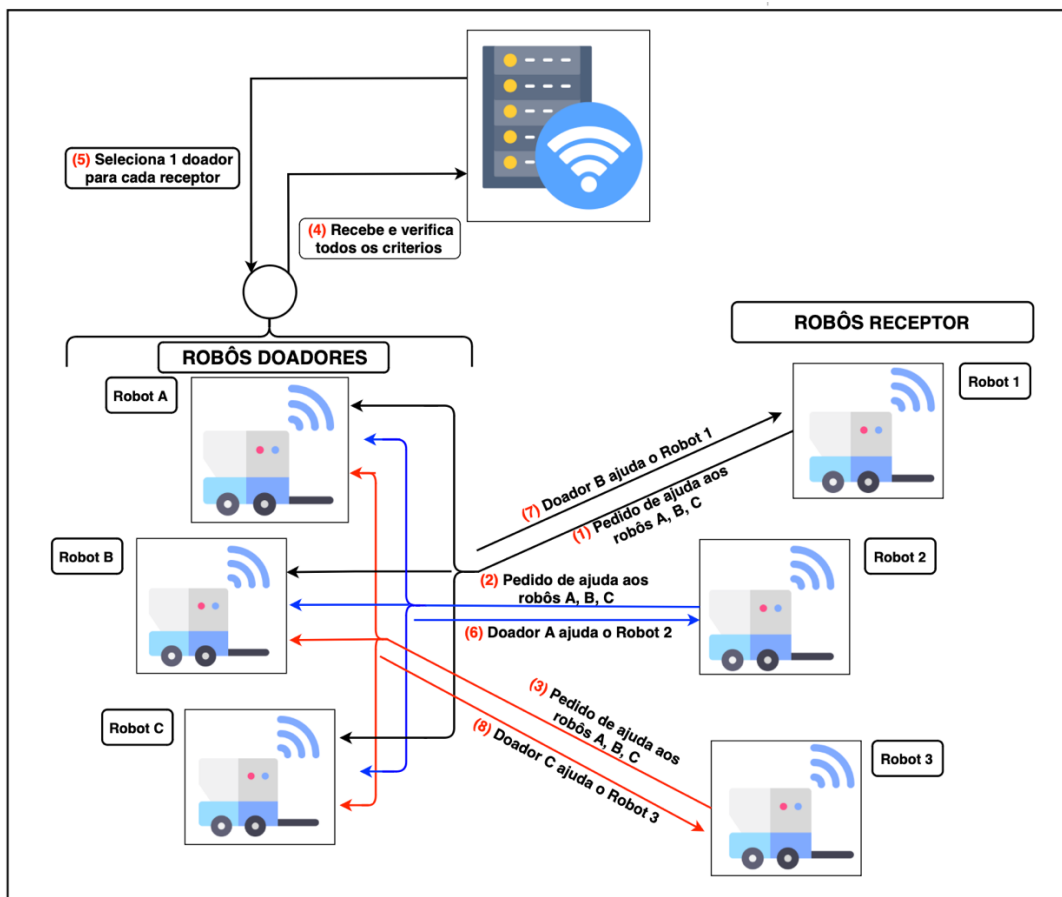


Figura 4.15 - Cenário Muitos para Muitos / Multi-mode do altruísmo entre os robôs.

Cada robô (A, B e C) pode tornar-se altruísta com base na distância e no tempo que tem disponível para realizar a ação de "apoio" e na sua própria tarefa. Todos os robôs doadores recebem as solicitações de cada um dos receptores (1, 2 e 3) que enviam os seus pedidos em intervalos de tempo diferentes (alguns milissegundos). O critério para seleccionar a ordem é baseado no custo da tarefa que o robô solicitante possui ou, caso todos tenham os mesmos custos de tarefa, é baseado na ordem de chegada dos pedidos. Neste caso, estamos a considerar que todos os custos são iguais, sendo assim, selecciona-se o primeiro solicitante para que todos os robôs doadores verifiquem a possibilidade de o ajudar, calculando a distância e o tempo necessário para essa ação. Assim, aqueles que têm uma distância e tempo aceitáveis enviam para o servidor, que os ajuda a seleccionar apenas um dos robôs com a menor distância e o menor tempo necessário, conforme mostrado no algoritmo da Fig. 4.16.

Após isso, é seleccionado o segundo solicitante de acordo com a ordem de chegada dos pedidos, e o processo é repetido apenas com os robôs doadores que ainda não foram seleccionados para ajudar. Este processo continua até que todos os robôs participantes deste cenário no sistema tenham sido verificados ou seleccionados.

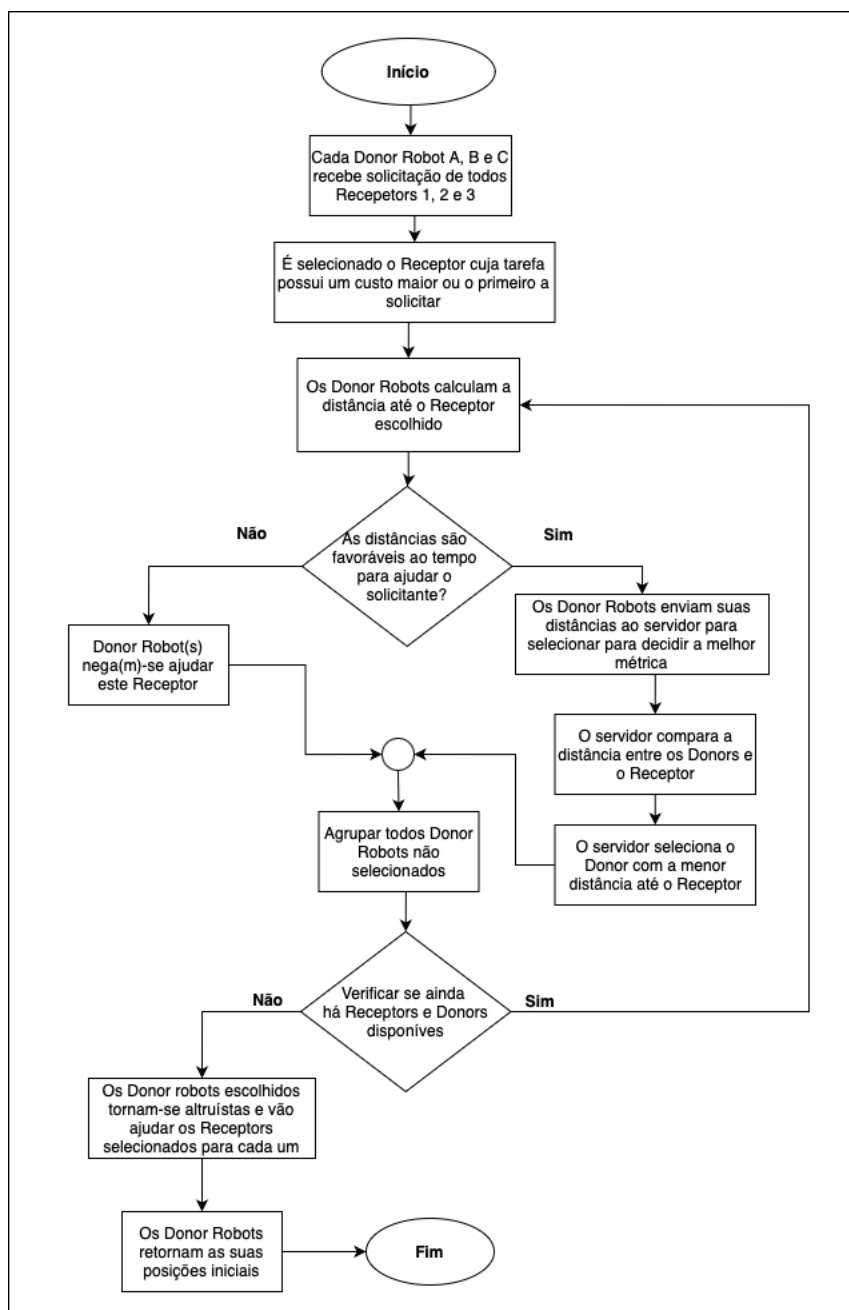


Figura 4.16 - Algoritmo do altruísmo no cenário Muitos para Muitos/ Multi-mode.

Cada um dos cenários apresentados acima, representam claramente o processo de selecção dos ou do robô altruísta levando em consideração a carga, tempo e distância até o robô solicitante. Esse processo decorre de forma automática desde o momento que um robô solicita ajuda.

Os algoritmos apresentados nas Fig. 4.10 ao 4.16, mostram os diversos cenários em que os altruístas podem selecionar-se dependendo do número de solicitantes e na forma de código, todo esse processo é realizado por uma função chamada *altruistic_robot* apresentada abaixo.

Algorithm 2: Function - altruistic_robot

```

Input: rbList, target_pose, robot
Output: return different values depending on the robs_with_time_power condition
1  if len(robs_with_time_power) > 0 do
2      dist = distance_between_points(robs_with_time_power[0].pose, target_pose)
3  else
4      dist = 0
5  end if
6  if len(robs_with_time_power) == 1 do
7      ...
8      return robs_with_time_power[0], 0, dist, robs_with_time_power, messages
9  else if len(robs_with_time_power) == 0 do
10     return 0, 0, 0, 0, messages
11 else
12     ...
13     for i --> 1 to robs_with_time_power do
14         ...
15         dist = distance_between_points(robs_with_time_power[i].pose, target_pose)
16         dists.append((dist, robs_with_time_power[i].name))
17         min_distance, closest_robot = min(dists, key=lambda x: x[0])
18         ...
19     end for
20     for i --> 1 to len(dists) do
21         if dists[i][1] == closest_robot do
22             return robs_with_time_power[i], i, min_distance, robs_with_time_power, messages
23         end if
24     end for
25 end if
26 return 0, 0, 0, 0

```

Esta função recebe como parâmetros a lista dos robôs doadores selecionados após o cálculo da distância (*rbList*), a posição e o nome do robô receptor. E após uma minuciosa verificação de todos os parâmetros, ela seleciona o robô altruísta para um determinado receptor analisado e em resposta a função retorna o robô altruísta selecionado (*robs_with_time_power[i]*), sua posição na lista do sistema (*i*), a distância (*min_distance*), os robôs que serão reavaliados para o próximo receptor (*robs_with_time_power*) e as mensagens de informação (*messages*).

Oposto ao altruísmo

É esperado que no ambiente colaborativo, como o próprio nome sugere, haja sempre a cooperação altruística entre os robôs, mas na realidade, o altruísmo pode não acontecer se:

1. Todos os robôs tiverem $p < 20\%$ (oposto da eq.1): nesta situação não teremos robôs doadores, pois a carga de todos é inferior a 40%.
2. Todos os robôs doadores tiverem tempo de espera menor ao tempo necessário para ajudar ou estiverem distantes demais, $t < 2 * d + t_{partilha}$ (oposto a eq.2).

Todos os robôs possuirão 3 estados que influenciarão diretamente com o processo altruístico, visto que cada um deles pode definir o acontecimento ou não do altruísmo:

- **Working** (Trabalhando): é o estado em que o robô se encontra no meio de uma tarefa e geralmente não é interrompido e não responde as solicitações de ajuda de outros robôs.
- **Waiting** (Aguardando): é o estado em que o robô se encontra quando não tem nenhuma tarefa por realizar e está aguardando pela próxima tarefa. Todos os robôs que se encontram nessa condição recebem as solicitações de ajuda e podem tornar-se altruístas se possuírem todos os critérios para tal.
- **Stopped** (Parado): nessa condição ou estado, o robô para por causa de fraca bateria enquanto aguarda por um altruísta.

4.3 Sistema de partilha de energia

A partilha de carga elétrica entre os robôs é um processo essencial que define o altruísmo, bem como a prova real de que a ação foi concluída com sucesso no ambiente colaborativo. Conforme é apresentado na Fig. 4.17, considerou-se que a partilha da carga elétrica será realizada por aproximação física (sistema de carregamento sem fios) entre os robôs que participam no processo altruísta, ou seja, o robô altruísta e o receptor.

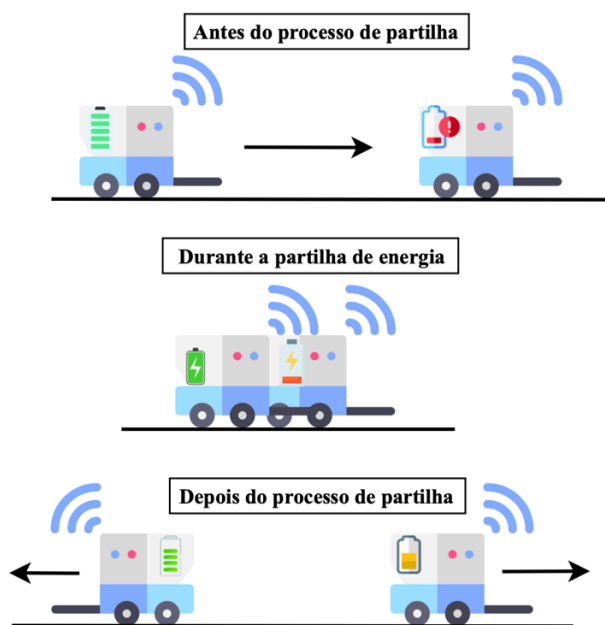


Figura 4.17 - Processo de partilha de carga por aproximação entre os robôs.

Embora o projeto não se foque propriamente na tecnologia que os robôs usariam para lhes permitir a partilha da carga, conforme ilustrado acima, foram identificadas algumas empresas e tecnologias que desenvolvem projetos permitindo que os AGVs carreguem as suas

baterias por contacto ou sem fio, em estações de carregamento ou sistemas de carregamento adaptados. Algumas dessas empresas e tecnologias incluem: DAIHEN, Wiferion e Wewo.

A DAIHEN, possui uma tecnologia D-Broad que oferece uma das mais eficientes soluções de carregamento sem fio com aplicações industriais, tornando-se o primeiro sistema de carregamento sem fio por ressonância magnética. Ela proporciona uma eficiência de 86% e uma folga máxima de 40mm, uma taxa de carregamento de 100% e uma corrente de 60A. Com o seu uso é possível uma redução na perda de energia de 26% e houve também casos em que se foi possível ter uma redução anual dos custos de eletricidade e das emissões de CO₂ até 60% [51].

Podemos observar que há uma rápida evolução no campo das baterias e na sua forma de carregamento, o que nos leva a considerar que o sistema de partilha de carga sem fio é realista. Assim, podemos acreditar que, com o decorrer do tempo, poderão ser desenvolvidos sistemas muito mais eficientes, com tempos de carregamento quase instantâneos para completar 20-30% da capacidade da bateria do AGV. Esta é a hipótese que será utilizada para a simulação dos resultados do nosso projeto.

RESULTADOS E TESTES

Neste capítulo é apresentado um caso de estudo onde é aplicado o altruísmo e seus resultados são mensurados em comparação com outras soluções ao altruísmo. Além disso, o sistema é testado em diferentes cenários para fortalecer seus benefícios.

5.1 Caso de estudo

Antes de apresentar as métricas de avaliação do projeto, será apresentado um caso de estudo, no qual as métricas serão calculadas com base nele.

Este caso de estudo tem como objetivo demonstrar a necessidade da aplicação do altruísmo entre os robôs dentro de um ambiente colaborativo. O conceito de altruísmo aplicado baseia-se na partilha de carga, sendo que o objetivo principal é dinamizar os processos industriais, aumentar a produtividade e evitar que as estações de fabricação fiquem paradas por períodos prolongados devido à falta de materiais. Este estudo será baseado na empresa Senseair.

Senseair é uma empresa localizada em Delsbo, na Suécia, fornecedora global de tecnologia para detecção da qualidade do ar e gás. Seu objetivo é dar sentido ao ar, fornecendo as melhores soluções de medição, serviços e inteligência possíveis.

A empresa produz vários tipos de sensores para atender seus clientes em todo mundo, porém para esse estudo, serão mencionados os sensores apresentados na Fig. 5.1 nomeadamente, S88, Sunrise, Sunlight e RDS R290.

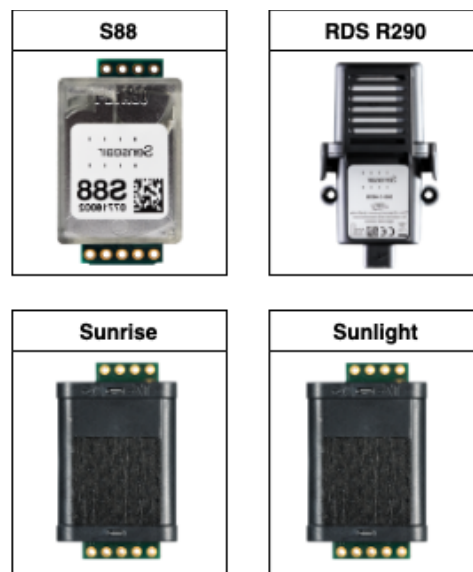


Figura 5.1 - Sensores fabricados pela Senseair [56].

Por motivos de confidencialidade, não serão descritos a função exata de cada um dos sensores e não serão aprofundados os dados sobre a produção, sendo que não faz parte do objetivo do presente trabalho.

Poucos anos atrás, a Senseair produzia seus sensores com base no esquema industrial apresentado na Fig. 5.2, que representa a produção do sensor Sunrise. Desde a etapa 1 (Recolha de Matérias) até a etapa 4 (Empacotar & Entregas), a produção passava por vários processos que envolviam humanos em mais de uma fase de produção, tornando o processo pouco automático.

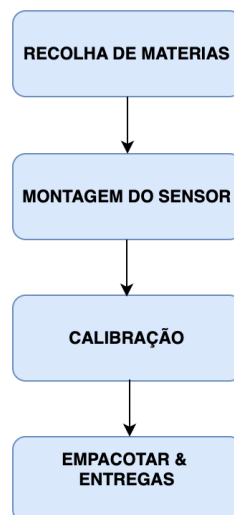


Figura 5.2 - Esquema de fabricação da Senseair.

Conforme se vê na Fig. 5.3, a mão de obra humana na fabricação dos sensores era de extrema importância de tal maneira que era preciso que alguém se deslocasse de um ponto para outro com o objetivo de transportar os sensores parcialmente produzidos para outra estação a fim de prosseguir com o acabamento.



Figura 5.3 - Processo de fabricação manual da Senseair.

A empresa tem acompanhado a evolução industrial de tal forma que tem transformado completamente a célula de produção para um modelo mais automatizado, eliminando assim a necessidade constante de especialistas humanos em trabalhos ou operações repetitivas, operações de alta precisão, bem como o transporte de equipamentos e acessórios. Dessa forma, todos os processos executados na célula apresentada na Fig. 5.2 foram integrados numa única estação de produção, que chamaremos de célula "All-in-one", apresentada na Fig. 5.4.

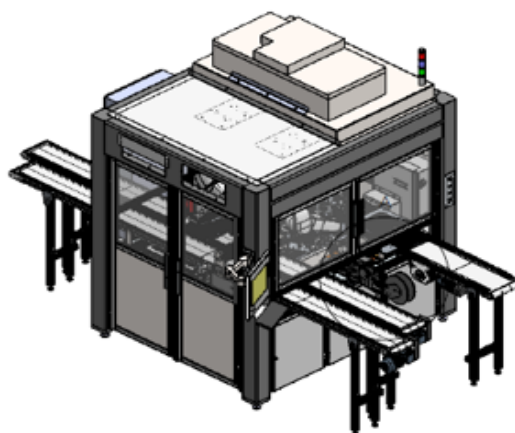


Figura 5.4 - Célula de produção All-in-one.

Com esta célula, pretende-se minimizar a presença humana nas principais operações de produção de cada um dos produtos da empresa. Para cada um dos sensores da Senseair, foi implementada uma célula All-in-one, e, conforme ilustrado na Fig. 5.1, que mostra a quantidade de alguns dos sensores da empresa considerados neste trabalho, surgiu a necessidade de implementar 4 células para a produção dos 4 tipos de sensores da empresa (caso hipotético para validação da arquitetura apresentada).

Um dos aspectos importantes quando se trata de um ambiente de produção com várias células de produção é o transporte de equipamentos ou acessórios para a fabricação do produto final. Assim, surgiu a necessidade de adicionar robôs móveis (AGV) no processo, de forma a transportar os equipamentos até as estações de produção (All-in-one), recolher os sensores produzidos e realizar muitas outras funções (caso hipotético).

Assumindo uma nova estrutura para o ambiente de produção da Senseair, vemos na Fig. 5.5 uma representação hipotética simples de como seria esta nova organização. Nela, estão incluídas as novas estações de produção, juntamente com os robôs de apoio, divididos em 4 robôs de transporte e 2 robôs de apoio geral, que têm funções como, por exemplo, retirar os sensores da célula para verificação ou transporte.

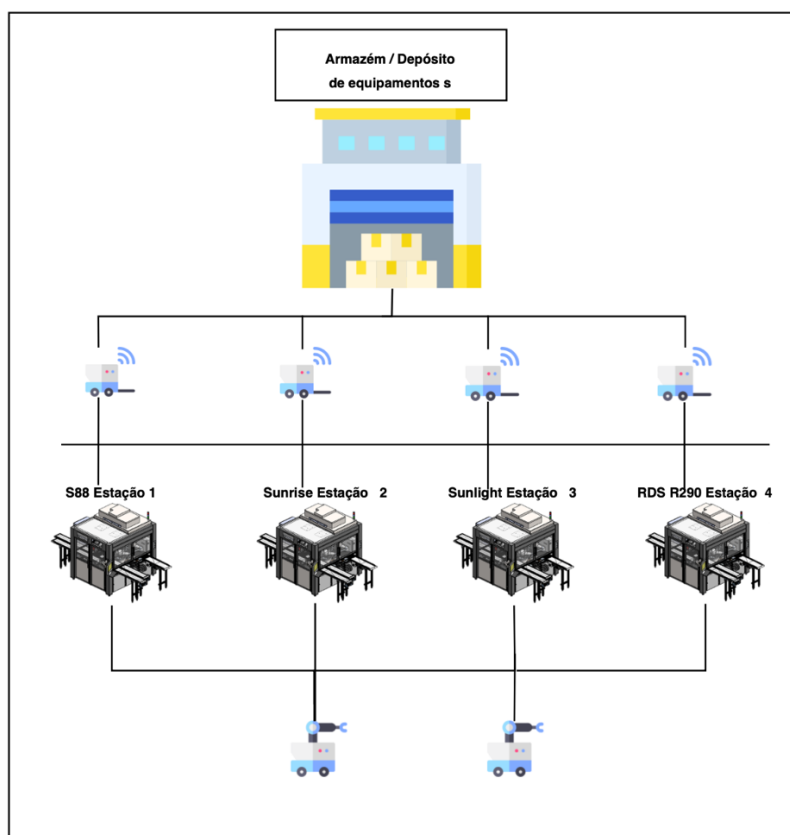


Figura 5.5 - Ambiente colaborativo com altruísmo entre os robôs.

O transporte de equipamentos do armazém até as estações de produção deve ser realizado dentro de um intervalo de tempo determinado, de acordo com a necessidade de cada célula. Da mesma forma, a recolha dos sensores das estações até o próximo nível de acabamento do produto, como por exemplo a calibração, exige que certas tarefas sejam realizadas com rapidez, de modo a não afetar a velocidade de fabrico nem a quantidade prevista de materiais produzidos.

Portanto, a Senseair depara-se com a seguinte situação: em alguns momentos, quando um ou mais AGVs estão no meio de uma tarefa, acabam por ficar sem carga de bateria. Dependendo do tempo de reação do especialista responsável pelo carregamento dos AGVs e o tempo necessário para substituir as baterias, isso pode afetar negativamente o processo de produção da empresa, uma vez que a rapidez e a continuidade são fatores cruciais. Por outro lado, alguns AGVs podem estar parados, aguardando o momento para executar as suas funções, enquanto a célula opera com as peças recebidas. Já outros AGVs podem estar em movimento, a caminho de outras estações para executar uma determinada função.

É exatamente neste cenário que pode ocorrer o altruísmo entre os robôs. Se uma célula de produção ficar sem peças ou se a esteira de saída de equipamentos pré-fabricados estiver cheia, isso causará a sua paragem, o que resultará em perdas financeiras para a empresa, uma vez que nenhum equipamento será produzido durante o tempo de inatividade da célula.

Desta forma, a Senseair poderia aplicar o conceito de altruísmo entre os robôs dentro do seu processo de fabrico de sensores, impedindo que nenhuma das células de produção fique parada devido a atrasos ou falta de AGVs por muito tempo. Os robôs que necessitem de ajuda devem cumprir os critérios apresentados na Tabela 5.1, enquanto um candidato a altruísta (Doador) deve preencher os critérios da Tabela 5.2.

Tabela 5.1 - Critérios de um solicitante.

CRITÉRIOS DO SOLICITANTE	
Estado	Stopped
Bateria	< 20%

Tabela 5.2 - Critérios de um candidato à altruísta.

CRITÉRIOS DO CANDIDATO À ALTRUÍSTA	
Estado	Waiting
Bateria	> 20%
Tempo de espera	>2*x

Em ambas as tabelas acima, vemos os estados que cada um dos robôs se encontra baseado no papel que executa (*Stopped* e *Waiting*), a “Bateria” que se refere à percentagem da bateria e o “Tempo de espera” que se refere ao tempo que um robô poderá ficar em estado de

espera enquanto aguarda pela próxima tarefa. O x está diretamente relacionado à distância, e essa relação é demonstrada nas Eq.7 e Eq.8 abaixo.

A distância e o tempo estão relacionados pela fórmula

$$v = x/t \quad (7)$$

onde, v : velocidade(m/s);

x : distância(m);

t : tempo(s).

De acordo com os dados obtidos, a velocidade máxima do AGV é de $2m/s$, mas na prática varia entre 1 e $1.8m/s$ [54]. Desta forma, assumiu-se que a velocidade média é de $1m/s$, logo teremos,

$$v = x/t \Rightarrow 1 = x/t \quad e \quad x=t \quad (8)$$

Considerando o tempo de ida e volta que o robô pode fazer, então a distância será percorrida duas vezes, o que levará o dobro do tempo, por isso na Tabela 5.2 temos que o “tempo de espera deve ser o dobro da distância a ser percorrida”. Essa fórmula pode variar consoante a velocidade que o AGV aplicará ao dirigir-se à posição de outro robô, na Tabela 5.3 vemos as variações da relação destas 3 variáveis (v , x e t).

Tabela 5.3 - Cálculo do tempo necessário para uma ação altruísta baseada na distância.

Velocidade: 1m/s	
Distância	X
Tempo	> 2*X
Velocidade: 1.8m/s	
Distância	X
Tempo	>2*1.8*X
Velocidade: 2m/s	
Distância	X
Tempo	>2*2*X

Baseado na sua necessidade, assumimos que a Senseair implementou, dentro do seu ambiente de produção, 6 robôs para auxiliar na realização das tarefas em 4 células da sua matriz. Esses robôs serão chamados de Robot1, Robot2, Robot3, Robot4, Robot5 e Robot6, conforme mostrado na Fig. 5.6. Cada um possui uma quantidade de carga da bateria e uma posição inicial que varia de acordo com o local em que o robô se encontra a realizar uma tarefa ou em espera. Inicialmente, temos os seguintes dados de cada um dos robôs, representados na Tabela 5.4.

Tabela 5.4 - Dados dos robôs no ambiente colaborativo.

Robot1			Robot2			Robot3		
Energia	Pose	Estado	Energia	Pose	Estado	Energia	Pose	Estado
80%	(7, 3)	Working	90%	(2, 1)	Working	75%	(11, 7)	Waiting
Robot4			Robot5			Robot6		
82%	(3, 10)	Working	58%	(0, 6)	Waiting	40%	(2, 9)	Working

Depois de algum tempo de trabalho auxiliando na produção, se movendo de um lado para o outro buscando e levando as ferramentas e peças necessárias para cada uma das células, os robôs apresentam os seguintes dados abaixo como se vê na Tabela 5.5.

Tabela 5.5 - Dados dos robôs no ambiente colaborativo depois de várias etapas de trabalho.

Robot1			Robot2			Robot3		
Energia	Pose	Estado	Energia	Pose	Estado	Energia	Pose	Estado
10%	(2.5, 0.5)	Stop.	19%	(3.5, 0)	Stop.	22%	(1, -1)	Wait.
Robot4			Robot5			Robot6		
20%	(1.5, 1)	Wait.	23%	(3, -1.5)	Wait.	21%	(3.5, 0.5)	Wait.

Nesse momento, os robôs 1 e 2, estão a solicitar ajuda visto que a sua carga da bateria está abaixo dos 20% e por isso entram no estado “Stop”. Os restantes robôs recebem as solicitações e começam a negociar avaliando seus critérios antes de tornarem-se altruístas.

De forma simples, esse processo obedece aos seguintes passos:

1. Identificação dos robôs solicitantes: para o nosso caso temos os robôs 1 e 2 por possuírem carga da bateria inferior a 20% e o estado “Stop”, eles tornam-se solicitantes.
2. Pré-selecionar os robôs que possuam carga suficiente para ajudar: neste ponto temos os restantes robôs, pois possuem carga superior a 20% e estão em estado “Waiting”.
3. Calcular a distância em relação ao tempo: neste ponto, é feito o cálculo da distância que cada um dos robôs candidatos está de ambos os robôs solicitantes, estas distâncias são apresentadas na Tabela 5.6.

Tabela 5.6 - Distância entre os candidatos e os solicitantes.

	To Robot2	To Robot1
Robot3	2.69m	2.12m
Robot4	2.23m	1.11m
Robot5	1.58m	2.06m
Robot6	0.50m	----

4. Verificar o tempo de espera de cada um e calcular o tempo a ser gasto com a ação altruísta para cada robô: aqui é comparado o tempo que o robô candidato tem disponível no estado “Waiting” com o tempo previsto que pode levar para executar a ação altruísta.
5. Selecionar o candidato com os melhores critérios para se tornar altruísta: neste passo, o(s) candidato(s) com os melhores parâmetros podem se tornar altruístas. Os altruístas são o Robot6 para auxiliar o Robot2 por possuir a menor distância e tempo suficiente, ao passo que para o solicitante Robot1 foi escolhido o Robot5 embora a menor distância verificada foi do Robot4, ele não possui o tempo necessário para a localização do Robot1.
6. Agir altruisticamente: após a conclusão de todos os pontos, cada um dos robôs altruístas (Robot5 e 6) começa a se mover à posição do solicitante que vai ajudar (Robot5 → Robot1 e Robot6 → Robot2) tal como vê na Fig.5.6.

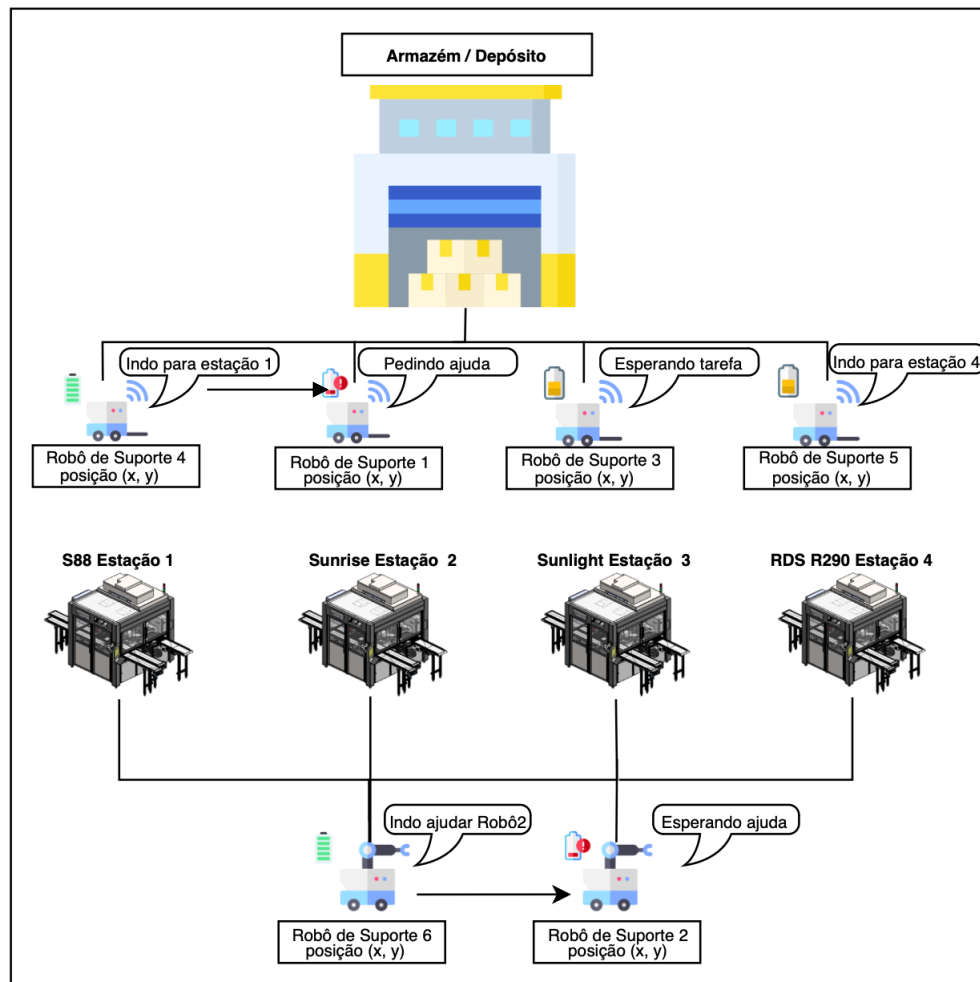


Figura 5.6 - Ambiente colaborativo com robôs agindo de forma altruística

Desta forma, vejamos a simulação do projeto usando as tecnologias mencionadas no capítulo anterior.

5.2 Simulação dos AGV's no estudo de caso

A simulação do projeto focou-se na ação altruística, em que um ou mais robôs podem ser selecionados para ajudar outros, compartilhando a sua carga/energia da bateria. Ela segue os mesmos padrões apresentados no estudo de caso da Senseair. Este processo envolveu as tecnologias mencionadas anteriormente. Devido a algumas limitações de recursos, a simulação não se foca em representar o ambiente do caso de estudo com as estações nem a representação dos robôs dentro deste ambiente. O objetivo da simulação é demonstrar o funcionamento do algoritmo altruísta e calcular o tempo que cada robô leva para executar um ato de altruísmo. Desta forma, o projeto limitou-se a introduzir um único robô num ambiente virtual, capaz de imitar e realizar todos os passos para a ação altruísta, de acordo com os parâmetros de cada um dos robôs virtuais, sendo que o algoritmo pode funcionar para vários robôs.

A simulação obedece a um conjunto de passos que os robôs devem seguir, desde a separação entre robôs doadores e receptores até à ação altruísta. Tratando-se da Senseair, que concluiu a instalação de 4 células All-in-one, assumimos que seriam necessários 4 AGVs para atender a cada uma das células e 2 inicialmente para dar suporte na saída das células, totalizando 6 robôs. Na Fig. 5.7 é apresentado o ambiente virtual de simulação. Nela, temos uma disposição das estações representadas por cilindros cinzentos numeradas de 1 a 4, um ambiente fechado por paredes pretas e os restantes cilindros representam outras estações com finalidades não relevantes para o projeto. No entanto, por questão de espaço, o TurtleBot3 poderá circular em toda a região marcada com a posição de um solicitante. De realçar que o robô inicia o seu movimento na posição (0, 0).

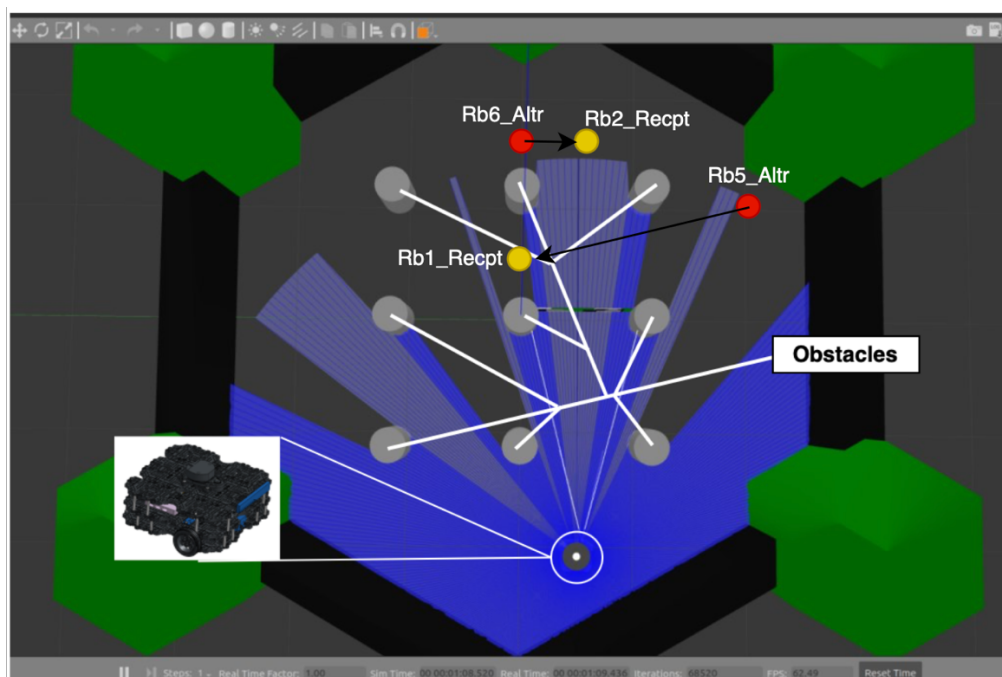


Figura 5.7 - Ambiente virtual de simulação.

Separação de robôs doadores e receptores

Reforça-se que os robôs doadores são aqueles que possuem carga suficiente para partilhar, enquanto os receptores são aqueles que solicitam ajuda após o nível de carga descer para 20%. Para fins de testes, as atribuições das cargas e localizações são feitas aleatoriamente para cada um dos robôs, a fim de formar os parâmetros necessários para o início da simulação. Na Fig. 5.8, temos um output onde são apresentados os robôs do sistema e as condições de cada um, com base nos critérios como: Nome, posição (a, b), carga (%) e tempo parado que o robô perdeu enquanto aguardava ser ajudado (X segundos).

De acordo com a Fig. 5.8, foi considerado 6 robôs, dentre eles, 4 doadores e 2 receptores (com borda vermelha). Convém realçar que as posições nas Fig. 5.8 estão trocas em (y, x) ao invés de (x, y).

Os robôs doadores são: *Robot6*, *Robot4*, *Robot3* e *Robot2*.

Os robôs receptores são: *Robot2* e *Robot1*

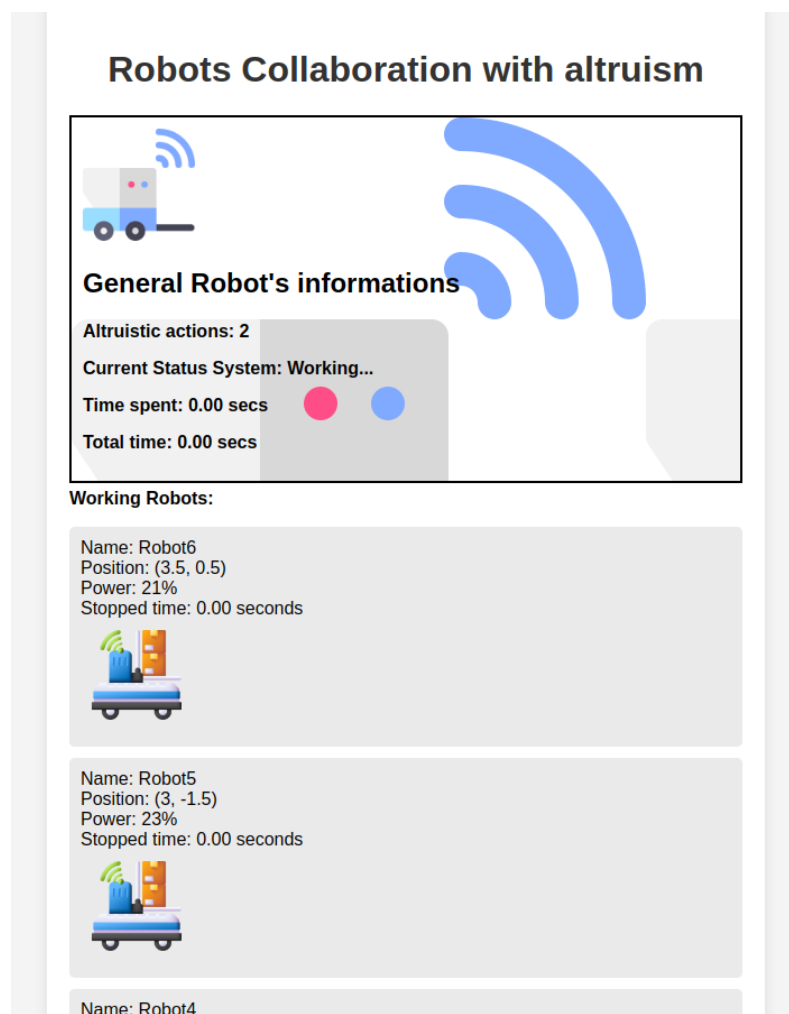


Figura 5.8 - Diferenciação entre robôs doadores e receptores.

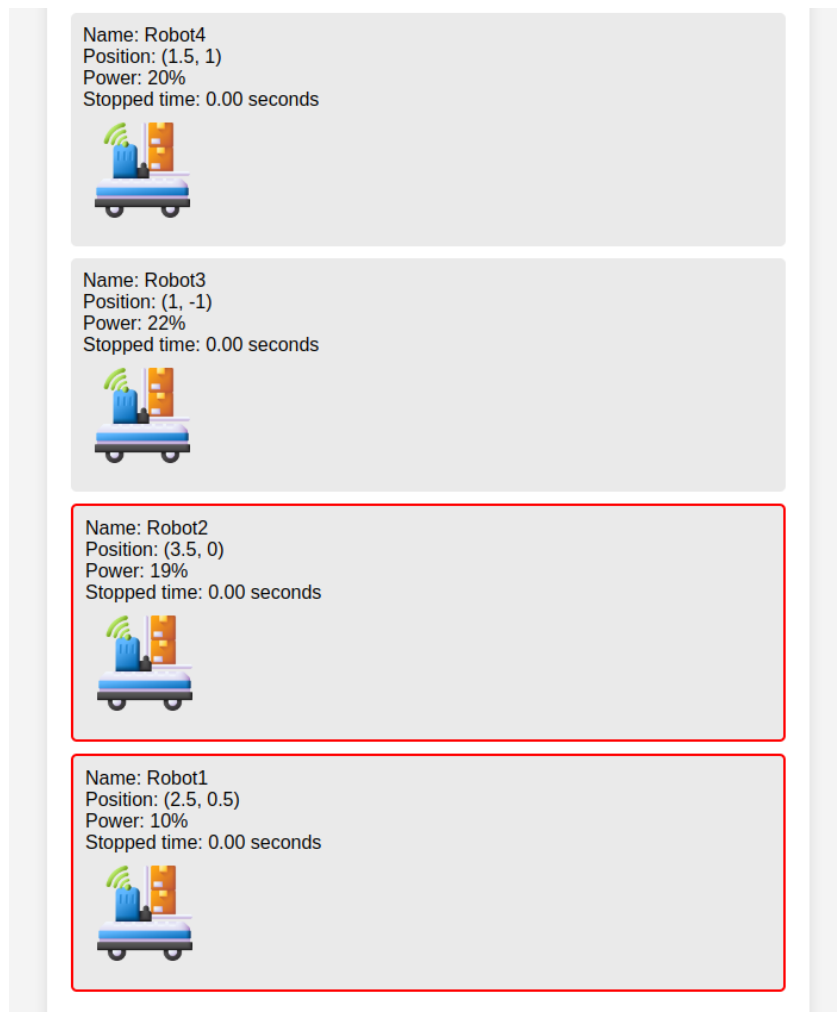


Figura 5.8 – Diferenciação entre robôs doadores e receptores.

Vê-se claramente que os robôs cuja carga da bateria (f) é inferior a 19% possuem uma borda vermelha. Desta forma, todos aqueles que possuem essa borda vermelha são considerados robôs receptores, enquanto aqueles que não possuem a borda vermelha são considerados doadores, por possuírem carga superior a 19%.

Na Fig. 5.9, vemos que o robô que irá simular o comportamento de cada um dos robôs do sistema, se encontra inicialmente na posição zero (0.0, 0.0) do sistema de simulação.

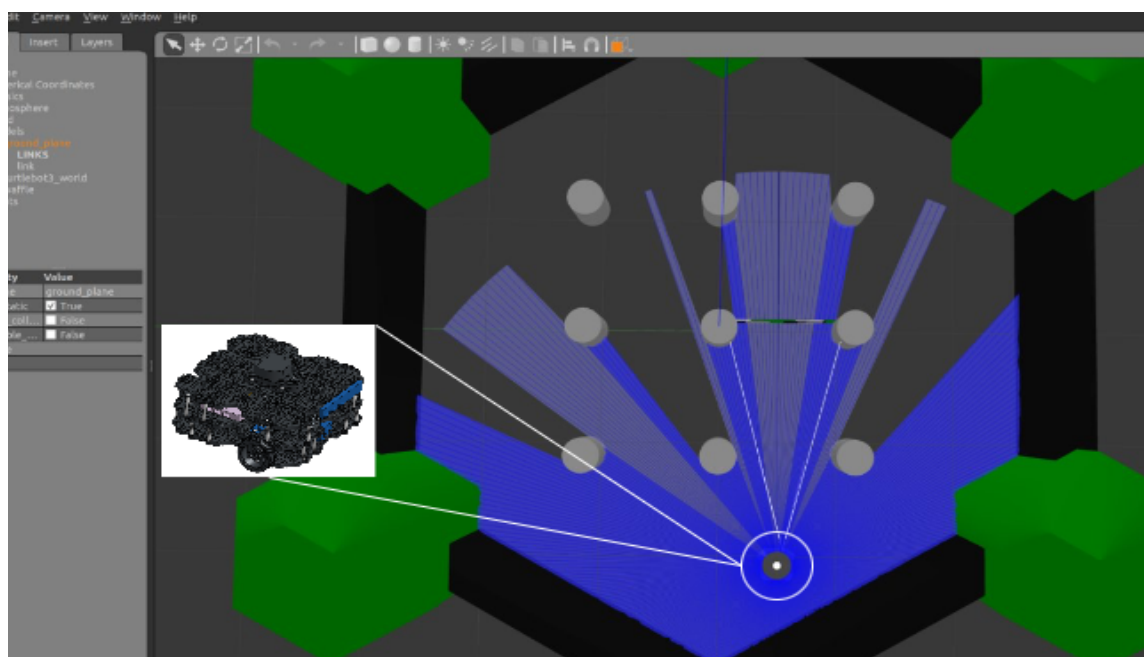


Figura 5.9 - Robô de simulação na posição (0.0, 0.0).

Avaliação do tempo e distância

Em menos de 0.5 segundo são verificados os critérios de cada um dos robôs selecionados e é feito o cálculo para verificar o tempo gasto até o robô solicitante.

Na Fig. 5.10 podemos ver as informações que são geradas pelo sistema enquanto faz a avaliação do tempo e o cálculo da distância.

Robot Log Viewer Prevision

Log Section 1

Robot6 distance to Robot2: (3.5, 0.0) is 0.5
Total distance / time needed is: 1.0
Robot5 distance to Robot2: (3.5, 0.0) is 1.5811388300841898
Total distance / time needed is: 3.1622776601683795
Robot4 distance to Robot2: (3.5, 0.0) is 2.23606797749979
Robot4 has not has not enough TIME for helping its neighbor Robot2: (3.5, 0.0)
Robot3 distance to Robot2: (3.5, 0.0) is 2.692582403567252
Robot3 has not has not enough TIME for helping its neighbor Robot2: (3.5, 0.0)

Log Section 2

The closest robot to the target Robot2 is Robot6 with a distance of 0.50

Log Section 3

Robot5 distance to Robot1: (2.5, 0.5) is 2.0615528128088303
Total distance / time needed is: 4.123105625617661
Robot4 distance to Robot1: (2.5, 0.5) is 1.118033988749895
Robot4 has not has not enough TIME for helping its neighbor Robot1: (2.5, 0.5)
Robot3 distance to Robot1: (2.5, 0.5) is 2.1213203435596424
Robot3 has not has not enough TIME for helping its neighbor Robot1: (2.5, 0.5)

Log Section 4

The selected altruistic is [Robot(name=Robot5, Pose=(3.0, -1.5), Power=23, Time=9, Task_time=0)]

Figura 5.10 - Cálculos das distâncias e verificação do tempo entre os robôs.

Na Fig. 5.10, vê-se que na primeira sessão (*Log Section 1*) calculou-se a distância de cada um dos robôs doadores até o receptor *Robot2* e na seção 3 (*Log Section 3*) calculou-se a distância dos robôs doadores *Robot5*, *Robot4* e *Robot3* até o receptor *Robot2*.

Reavaliação de robôs não selecionados

Em muitos casos, pode ocorrer a situação em que um determinado robô não possua os critérios para ajudar o receptor1, mas seus atributos podem ser suficientes para ajudar o receptor2. Neste caso, ele deve ser reavaliado de acordo com os critérios do próximo receptor. Na Fig. 5.10, é possível verificar como os robôs doadores são avaliados mais de uma vez, com base no número de robôs receptores.

Na “Log Section 1” representado na Fig. 5.10 são avaliadas as distâncias de todos os robôs doadores (6, 5, 4 e 3) até o receptor *Robot2*, em que o *Robot4* e 3 não possuem tempo o suficiente para ajudar o receptor2, no “Log Section 2” mostra-se que o mais próximo é o doador *Robot6*. Embora os *Robot4* e 3 não possuem tempo suficiente para o receptor2, na nova iteração é selecionado a posição do receptor *Robot1* como *target*, e os robôs 4 e 3 anteriormente eliminados, foram novamente verificados incluindo o *Robot5* “Log Section 3” já que o *Robot6* foi selecionado na primeira verificação.

Seleção do robô altruísta

Os critérios são avaliados e os robôs altruístas são selecionados, para nosso exemplo foram os robôs *Robot6* para ajudar o receptor *Robot2* (Log Section 2) e o *Robot5* para ajudar o *Robot1* (Log Section 4). Quanto aos receptores *Robot3* e 4, infelizmente não possuem tempo o suficiente (Log Section 4) conforme é mostrado na Fig. 5.10. Neste caso teremos apenas 2 ações altruísticas.

Uma vez selecionado os robôs altruístas, iniciará o processo de ajuda em que o robô se moverá até a posição em que se encontra o robô solicitante. De acordo com a Fig. 5.6, os robôs possuem as posições, cargas e os altruístas selecionados e apresentados na Tabela 5.7.

Tabela 5.7 - Parâmetros dos robôs doadores e receptores.

Nome do robô doador	Posição	Energia
Robot6	y=3.5, x=0.5	21%
Robot5	y=3, x=-1.5	23%
Robot4	y=1.5, x=1	20%
Robot3	y=1, x=-1	22%
Nome do robô receptor	Posição	Energia
Robot2	y=3.5, x=0	19%
Robot1	y=2.5, x=0.5	10%
Robô altruísta	Robô receptor	Distância (metros)
Robot6	Robot2	0.50 (go on and back 1)
Robot5	Robot1	2.06 (go on and back 4.12)

De acordo com a simulação, trabalhou-se apenas com um único robô que irá simular esse movimento que neste caso será:

- O robô parte da posição (0, 0) para a posição do primeiro robô altruísta (*Robot6*). O sistema de coordenadas segundo o qual o robô se movimenta, é apresentado na Fig. 5.11.

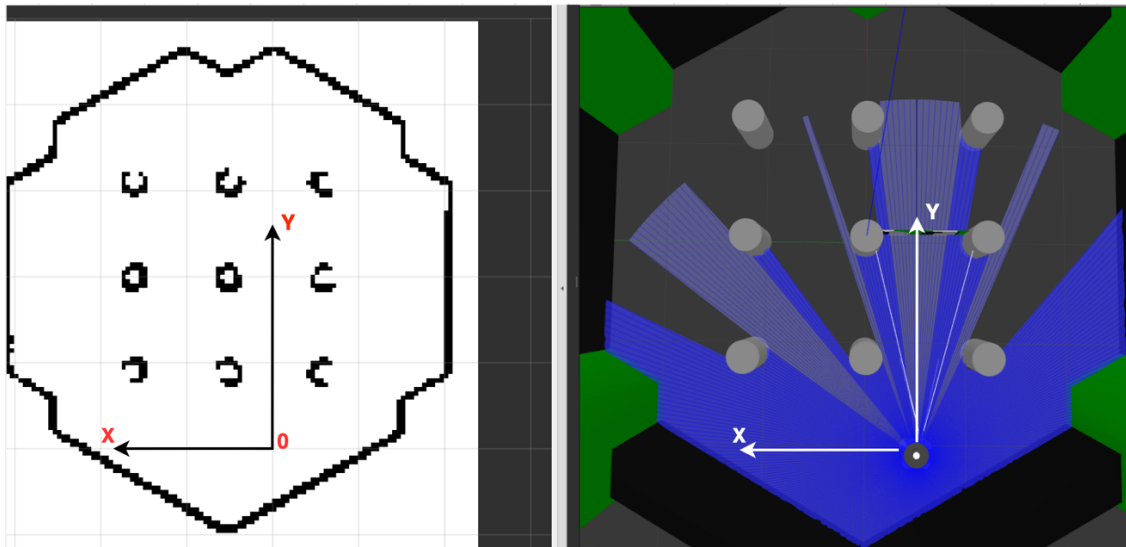


Figura 5.11 - Sistema de coordenadas do robô na posição (0, 0).

- O nosso robô de simulação começa a mover-se até a posição do altruísta (Robot6, $y=3.5$, $x=0.5$) para imitar a sua movimentação de acordo com o nosso algoritmo. A Fig. 5.12 mostra o ambiente de simulação no Gazebo (direita) e o ambiente de construção da trajetória do Navigation2 e o SLAM (esquerdo).

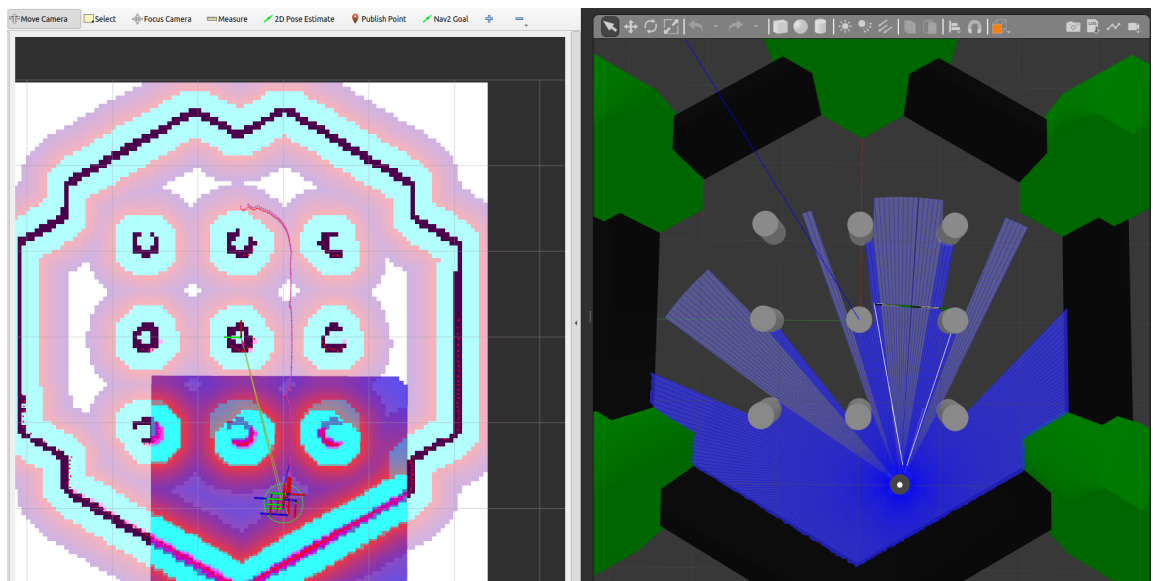


Figura 5.12 - Robô de simulação movimenta-se para posição do altruísta Robot6 (3.5, 0.5).

- Estando na posição do primeiro robô altruísta (Robot6), inicia o movimento de ajuda a caminho do solicitante (Robot2, pose $x=3.5$, $y=0$), dentro de um tempo controlado até chegar ao destino, considera-se que se faz partilha da sua carga de bateria de forma instantânea. Na Fig. 5.13, o ambiente de simulação da trajetória projeta o caminho com uma sinalização muito curta, visto que a distância entre eles é de apenas 0.5m.

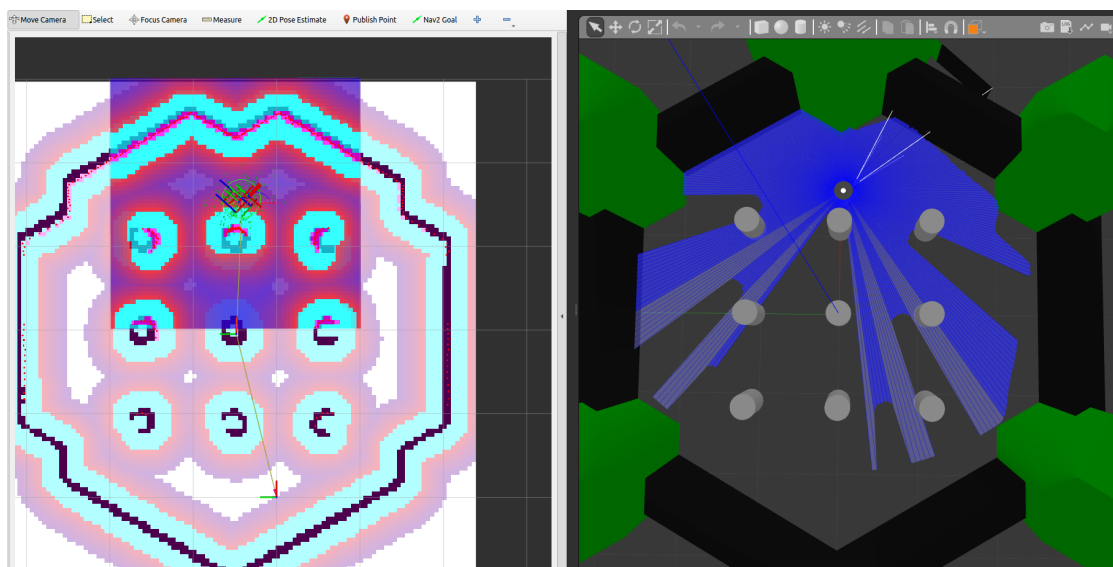


Figura 5.13 - Movimento do robô altruísta para a posição do Robot2 (0, 3.5).

- Tendo partilhado a carga, será decrementado no robô altruísta (*Robot6*) a percentagem doada.

Desta forma, o *Robot6* que possui 21%, acaba por doar 10% ao receptor2 como veremos na Fig. 5.16. Uma vez feita a ação altruística, o robô de simulação inicia a se movimentar até a posição do segundo robô altruísta (*Robot5*, $y=3$, $x=-1.5$) para executar o seu movimento de altruísta tal como mostra a Fig. 5.14.

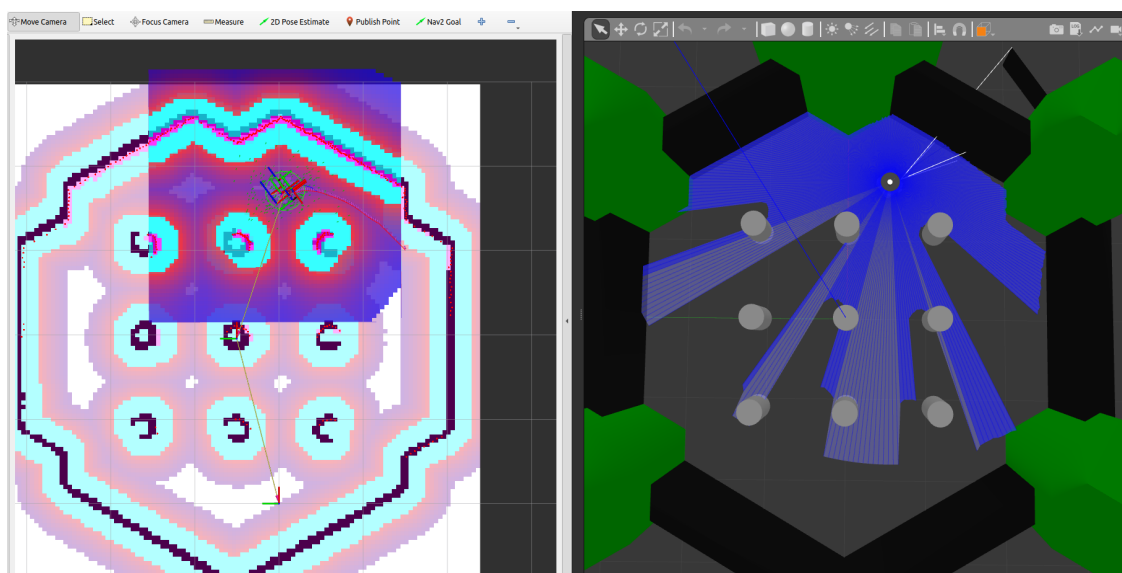


Figura 5.14 - Robô de simulação movimentando-se para posição do altruísta Robot5 (-1.5, 3).

- Posto na posição do Robot5, inicia o processo de ajuda em direção ao robô solicitante Robot1 na posição $y=2.5$, $x=0.5$ como mostra a Fig. 5.15.

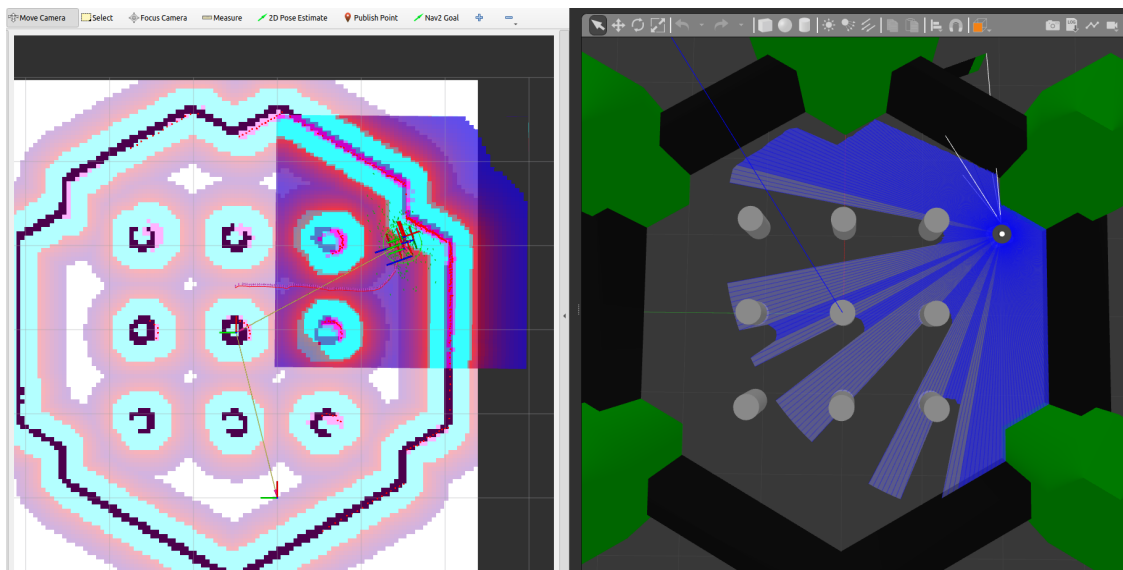


Figura 5.15 - Movimento do robô altruísta Robot5 para a posição do Robot1 (0.5, 2.5).

- Após alcançar a posição do Robot1, o Robot5 partilha a sua carga e volta para a sua posição inicial ($y=3$, $x=-1.5$) tal como é mostrado na Fig. 5.16.

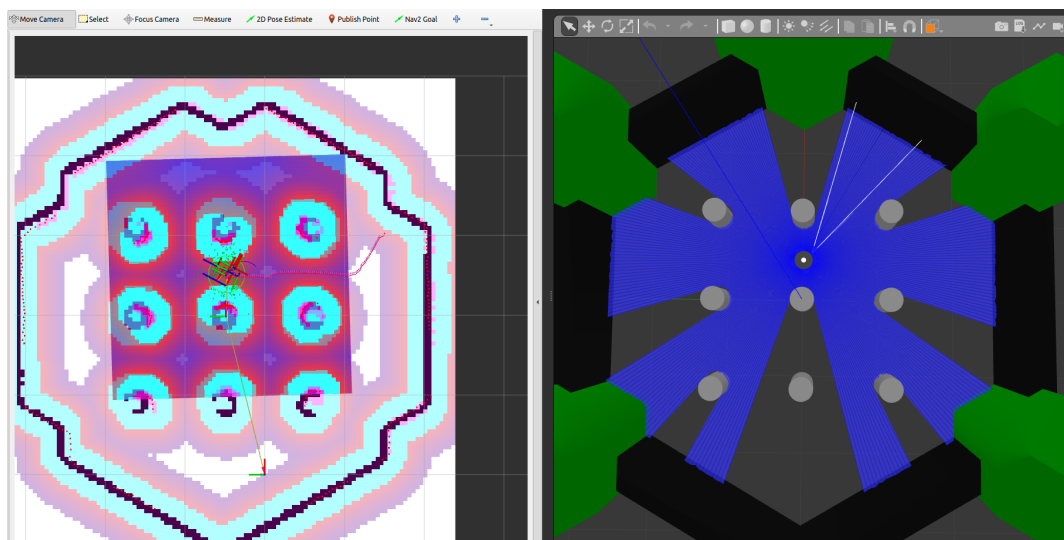


Figura 5.16 - Movimento do robô altruísta Robot5 voltando à sua posição inicial (-1.5, 3).

No caso de haver mais robôs altruístas, este processo vai se repetindo até que não haja mais ações altruístas a serem realizadas. Esta simulação de 6 robôs foi baseada na célula de produção da Senseair com apenas 4 tipos de sensores produzidos, um em cada estação de produção.

Tendo em conta que o processo altruísta foi realizado na sua totalidade, teremos o output atualizado em tempo real cada vez que cada uma das ações é concluída. Na Fig. 5.17, vemos as informações geradas pelo sistema durante o processo todo de execução.

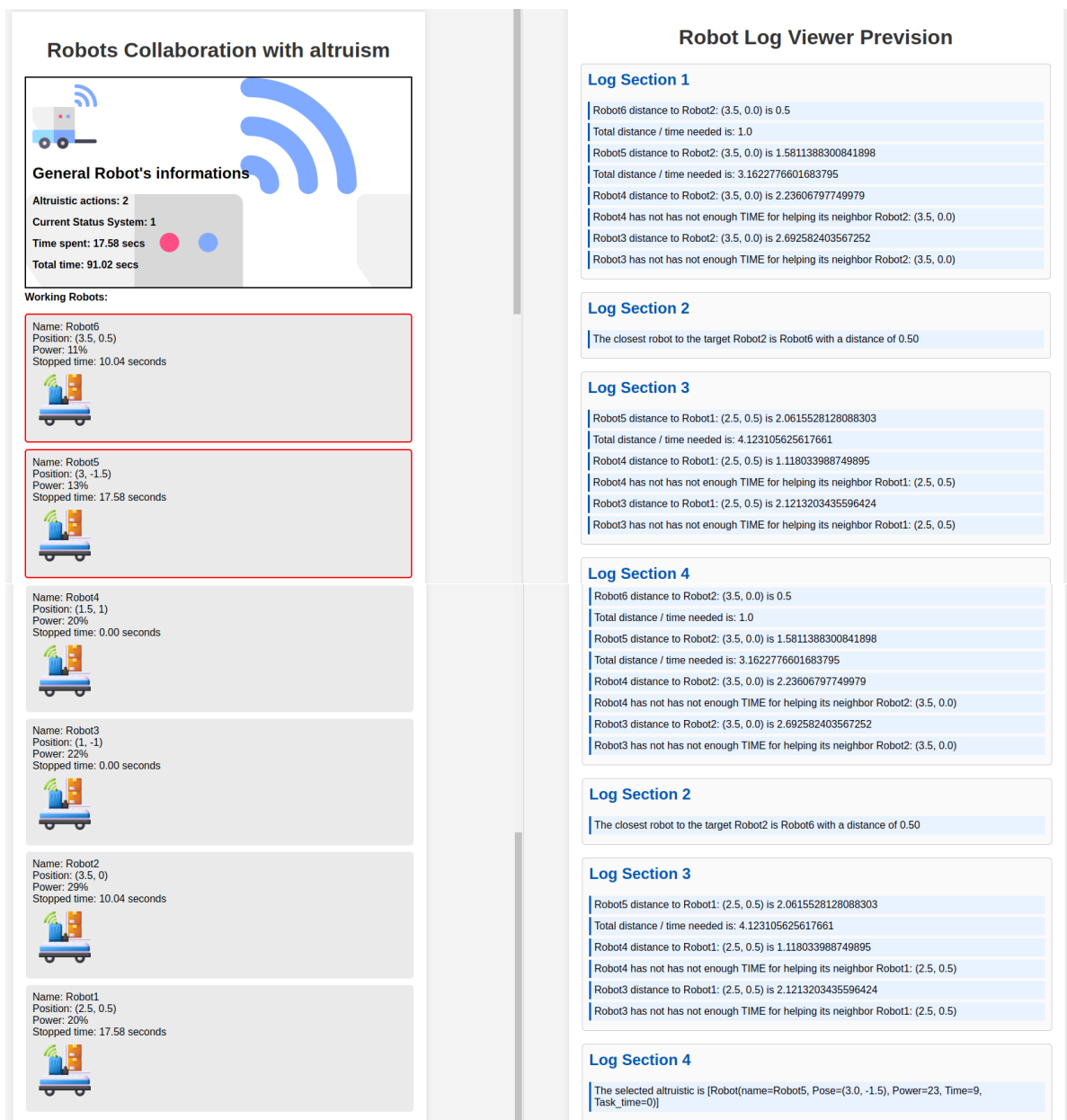


Figura 5.17 - Output do sistema após a finalização do altruísmo.

Focando-se na parte esquerda, vê-se na parte superior a quantidade de ações altruísticas realizadas (*Altruistic action*) que para o nosso caso foram duas, o campo *Current Status System* indica o estado do sistema, se for 1 indica que tudo correu bem, se for 0, indica que algum erro no processo, o *Time spend* indica o tempo gasto pelo atual robô altruísta durante a movimentação até o receptor, logo, o último altruísta acionado (*Robot5*) gastou 17.58 segundos e o *Total time* indica a soma do tempo total gasto por todos os robôs altruístas de ida e volta, o tempo que os receptores ficaram parados aguardando pela chegada do altruísta bem como o tempo gasto por causas não previstas como por exemplo sinalização de *Stop* porque outro robô passava na mesma linha de circulação.

Temos a sessão dos “Working Robots” que se refere aos robôs que fazem parte do processo de verificação consoante a solicitação de ajuda de um receptor. Nele já podemos notar que o *Robot2* e 1 que anteriormente estava com carga baixa, agora possuem carga suficiente para continuar a executar suas tarefas, ao passo que os robôs que agiram de forma altruístas (*Robot6* e 5) ficaram com a carga baixa devido a partilha. Cada um dos robôs mostra o tempo em segundos que gastou durante a execução (altruístas) ou espera (solicitantes) da ajuda. Os robôs 4 e 3 possuem seus dados inalteráveis porque não participaram do processo altruísticos, embora tivessem sido considerados doadores.

Ao lado direto, vemos as informações geradas pelo sistema mostrando dados de cálculo da distância bem como os robôs mais próximos dos solicitantes, nos permitindo acompanhar de perto o processo de escolha e seleção do altruísta no sistema.

5.3 Comparação entre o processo altruístico e o não- altruístico (manual) e métricas de avaliação

A solução de partilha de carga da bateria baseada no altruísmo entre os robôs é uma das várias abordagens para reduzir o tempo de inatividade de um robô durante o carregamento da sua bateria. As soluções mais comuns observadas são a troca manual da bateria, geralmente realizada por um técnico, e o carregamento automático através de uma estação fixa, onde os AGV's se dirigem sempre que o nível da bateria estiver abaixo do normal.

De acordo com a simulação apresentada, é importante reforçar que o tempo de transferência de energia da bateria entre um AGV e outro foi considerado instantâneo, uma vez que, até a data da realização deste projeto, as tecnologias de carregamento de bateria por proximidade entre dois dispositivos ainda estão em desenvolvimento. Isso resulta em uma escassez de informações que poderiam contribuir para um aprimoramento mais preciso deste projeto. Assim, os dados da simulação consideram apenas os resultados do processo altruísta com base no tempo de viagem que os robôs fazem de ida e volta entre si. O tempo gasto para transferir a energia da bateria de um robô para o outro é considerado zero (0).

O processo de carregamento manual, propriamente feito por um especialista humano, envolve os seguintes fatores:

- **Tempo de reação (t_r):** o tempo que leva para que o especialista se apercebe da necessidade de trocar a bateria. Geralmente a reação é rápida desde que o especialista esteja no local de monitorização, podendo variar de 3 a 7 segundos incluindo a leitura dos dados do robô como posição e o tipo de bateria.
- **Tempo de alocação (t_a):** é o tempo gasto de ida ao armazém para pegar a bateria até a chegada ao robô. Quanto mais distante o robô estiver, mais tempo vai levar. Sendo assim, vamos considerar 3 a 5 minutos.
- **Tempo de troca da bateria (t_b).** O tempo de troca da bateria pode levar até 2 minutos [55].

Ao passo que o processo por altruísmo envolve as seguintes variáveis:

- **Tempo de negociação (t_n):** é o tempo que leva desde a solicitação de ajuda até a seleção do altruísta, ela pode durar até 4 segundos.
- **Tempo de deslocação (t_d):** é o tempo que o robô altruísta gasta até chegar no seu destino. É considerado somente o tempo de ida.
- **Tempo de carregamento (t_c):** é o tempo que leva para se fazer a transferência de energia da bateria de um robô para outro.

Quer no processo altruístico quer no manual, o tempo gasto para realizar cada uma das atividades afeta diretamente na produção sendo que pode fazer com que uma célula de produção pare durante esse tempo (*downtime*).

O *downtime* (d_t) pode variar consoante o processo em causa, porém de forma geral, o *downtime* é o somatório do tempo que é gasto durante o processo quer manual quer por altruísmo.

No processo manual temos,

$$d_t = t_r + t_{al} + t_b \quad (9)$$

No processo por altruísmo temos,

$$d_t = t_n + t_d + t_c, t_c = 0, \quad d_t = t_n + t_d \quad (10)$$

Considerando que na simulação acima, os robôs *Robot1* e *Robot2* tiveram de parar suas operações e aguardar por ajuda, vejamos nas Tabelas 5.8 e 5.9 abaixo, o *downtime* (em segundos) que cada um dos processos pode causar dentro do ambiente de fabricação para cada um dos robôs.

Tabela 5.8 - Cálculo do Downtime em cada um dos processos (altruísmo e manual) para o Robot1.

ROBOT1			
Processo Manual		Processo Altruístico	
Tempo de reação	6s	Tempo de negociação	4s
Tempo de alocação	300s	Tempo de deslocação	17.58s
Tempo de substituição	40s	----	
Tempo de carregamento	0	Tempo de carregamento	0
Downtime	346s	Downtime	21.58s

Tabela 5.9 - Cálculo do Downtime em cada um dos processos (altruísmo e manual) para o Robot2.

ROBOT2			
Processo Manual		Processo Altruístico	
Tempo de reação	5s	Tempo de negociação	3s
Tempo de alocação	240s	Tempo de deslocação	10.06s
Tempo de substituição	35s	----	
Tempo de carregamento	0	Tempo de carregamento	0
Downtime	280s	Downtime	13.06s

Os *downtime's* apresentados acima foram calculados para apenas 2 robôs, porém convém lembrar que dentro de um ambiente de fabricação com vários robôs, o *downtime* pode ser ainda maior principalmente para o processo manual.

Supondo que esse processo se repete durante todos os dias que a empresa funciona, isso teria um custo muito alto para a empresa se for a considerar 1 ano de trabalho nas mesmas condições.

E quanto maior o tempo gasto neste processo, maior é a perda em questão financeira. Nesse contexto entra uma nova variável muito importante que chamamos de “Perda financeira pela paragem de uma estação (PF)”.

A perda financeira é o valor em dinheiro que a empresa perde causado pelo *downtime* considerando o tempo que leva para a produção de um determinado sensor. Segundo os dados disponíveis, a fabricação de cada um dos sensores da Senseair obedece a várias etapas tais como: fabricação de componentes, montagem, calibragem e teste de qualidade. O tempo total gasto nesse processo todo pode durar de 1h a 1h:30 minutos (excluindo a fabricação de componentes) dependendo do sensor. Na Tabela 5.10 abaixo, vemos uma separação do tempo por etapas.

Tabela 5.10 - Duração em minutos de cada etapa de produção.

ETAPA	DURAÇÃO EM MINUTOS
Montagem	15 a 20
Calibragem	10 a 25
Teste de qualidade	30 a 60

Baseado nestes dados, assume-se que o processo de altruísmo e manual afectam diretamente a etapa de montagem de cada sensor dentro da célula que pode durar em média 17.5 minutos. Outra questão importante para podermos determinar o PF é o custo de cada produto, da Senseair, na Tabela 5.11 é apresentado o valor de venda de alguns dos produtos da empresa.

Tabela 5.11 - Preço dos produtos da Senseair.

PRODUTO	PREÇO EM EUROS
Sunrise	52
Sunlight	60
S8	59
K30	100
Média	67,75

Tendo achado o tempo médio da montagem e do valor de venda de alguns dos produtos que se beneficiariam com o processo altruístico no seu sistema de fabricação, pode-se verificar claramente que,

$$PF = \frac{\text{dias_de_produção} * \text{preço_médio} * \text{downtime}}{\text{tempo_média_de_produção}} \quad (11)$$

Onde,

o downtime e o tempo_médio_de_produção devem estar na mesma unidade [minutos].

Desta forma podemos ter uma estimativa da PF no decorrer de 1 ano tal como se vê na Tabela 5.12.

Tabela 5.12 - Comparação da perda financeira entre processo manual e altruístico.

DADOS GERAIS		MANUAL		ALTRUÍSMO	
Montagem	Dias	Downtime	PF	Downtime	PF
17,50 €	1	5,8	22,45 €	0,36	1,39 €
Preço	10	58	224,54 €	3,6	13,94 €
67,75 €	20	116	449,09 €	7,2	27,87 €
	31	179,8	696,08 €	11,16	43,21 €
	60	348	1 347,26 €	21,6	83,62 €
	90	522	2 020,89 €	32,4	125,43 €
	120	696	2 694,51 €	43,2	167,25 €
	160	928	3 592,69 €	57,6	222,99 €
	200	1160	4 490,86 €	72	278,74 €
	230	1334	5 164,49 €	82,8	320,55 €
	260	1508	5 838,11 €	93,6	362,37 €
	290	1682	6 511,74 €	104,4	404,18 €
	320	1856	7 185,37 €	115,2	445,99 €
	360	2088	8 083,54 €	129,6	501,74 €

5.3.1 Testes em cenários diferentes

Considerando os dados assumidos nos pontos acima e nas Tabelas 5.10 e 5.11 relacionados com os tempos gastos quer no processo manual quer no altruísta, apresenta-se na Tabela 5.13 vários testes considerando diferentes cenários em que o *Downtime* e a *montagem* são dados em minutos.

Tabela 5.13 - Testes do processo manual e altruístico em diferentes cenários.

TESTES EM CENÁRIOS DIFERENTES							
Montagem	17,50 €	Preço	67,75 €				
CENÁRIOS	ROBÔS			MANUAL		ALTRUÍSMO	
---	Doadores	Receptores	Actos altruíst.	Down-time	PF	Down-time	PF
1	R2, R3, R4, R5	R1, R6	R3:R1, R4:R6	9	34,84 €	1,359	5,26 €
2	R1, R3, R4, R5, R6	R2	R1:R2	4	15,49 €	0,168	0,65 €
3	R2, R4	R1, R3, R5, R6	R2:R5, R4:R1	13	50,33 €	0,579	2,24 €
4	R1, R2, R4, R5	R3, R6	R4:R6, R1:R3	8	30,97 €	0,369	1,43 €
5	R2, R3, R5	R1, R4, R6	R3:R4, R2:R6	11	42,59 €	0,353	1,36 €
6	R1, R5, R6	R2, R3, R4	R1:R4, R5:R3, R6:R2	16	61,94 €	0,932	3,61 €
7	R3, R4	R1, R2, R5, R6	R3:R5	6	23,23 €	0,235	0,91 €
8	R2, R3, R6	R1, R4, R5	R2:R4	4,5	17,42 €	0,411	1,59 €
9	R1, R2, R4	R3, R5, R6	R1:R5, R4:R6, R2:R3	15	58,07 €	1,258	4,87 €
10	R1, R2	R3, R4, R5, R6	R2:R3	5,5	21,29 €	0,386	1,49 €
11	R5	R1, R2, R3, R4, R6	R5:R3	5,5	21,29 €	0,322	1,25 €
12	R1, R2, R3, R6	R4, R5	R6:R5	6	23,23 €	0,378	1,46 €
13	R1, R3, R4, R5, R6	R2	R6:R2	5	19,36 €	0,327	1,27 €
14	R2, R3, R4, R5, R6	R1	R2:R1	7	27,10 €	0,202	0,78 €
15	R2, R4, R5	R1, R3, R6	R4:R1, R2:R3	12,5	48,39 €	0,999	3,87 €
16	R1, R2, R3, R6	R4, R5	R6:R5, R3:R4	14	54,20 €	0,504	1,95 €
17	R2, R3, R4, R5	R1, R6	R2:R1, R4:R6	10,5	40,65 €	0,588	2,27 €
18	R1, R2, R3	R4, R5, R6	R2:R4, R3:R6, R1:R5	17	65,81 €	1,486	5,75 €
19	R5, R6	R1, R2, R3, R4	R5:R3	5,5	21,29 €	0,319	1,23 €
20	R2, R4, R5	R1, R3, R6	R2:R3, R5:R1	11	42,59 €	0,520	2,01 €
21	R1, R4, R6	R2, R3, R5	R1:R2, R4:R3	11,5	44,52 €	0,478	1,85 €
22	R1, R3, R5	R2, R4, R6	R1:R2, R5:R4, R3:R6	15,5	60,01 €	0,588	2,28 €
23	R2, R3, R5, R6	R1, R4	R5:R1, R6:R4	13	50,33 €	0,378	1,46 €

24	R1, R4, R5, R6	R2, R3	R4:R2, R1:R3	10	38,71 €	0,546	2,11 €
25	R2, R5	R1, R3, R4, R6	R5:R3	6	23,23 €	0,353	1,36 €
26	R1, R3, R4, R5	R2, R6	R1:R6, R3:R2	10,5	40,65 €	0,638	2,47 €
27	R2, R5	R1, R3, R4, R6	R2:R3	5,5	21,29 €	0,285	1,10 €
28	R1, R3, R5, R6	R2, R4	R3:R2, R5:R4	11	42,59 €	0,754	2,92 €
29	R1, R4, R5, R6	R2, R3	R1:R3, R4:R3	13,5	52,26 €	0,621	2,40 €
30	R1, R2, R4, R6	R3, R5	R2:R3, R4:R5	10	38,71 €	0,386	1,50 €
TOTAL	---	---	---	292,5	1 132,39 €	16,719	64,73 €

Na tabela acima testou-se o projeto em 30 cenários diferentes e como resultado podemos comparar os resultados totais de cada processo. Visivelmente o valor total de perda financeira (PF) do processo manual é 17,5 vezes maior que do altruísmo, mostrando assim o quão económico é o altruísmo. Na Tabela 5.14 é apresentado a média dos valores obtidos da tabela acima.

Baseado na Tabela 5.13 acima, apresenta-se os gráficos que indicam de forma visual a real diferença entre o Downtime do altruísmo e do manual (Fig. 5.18) bem como a Perda financeira do altruísmo face ao manual na Fig. 5.19.

Downtime

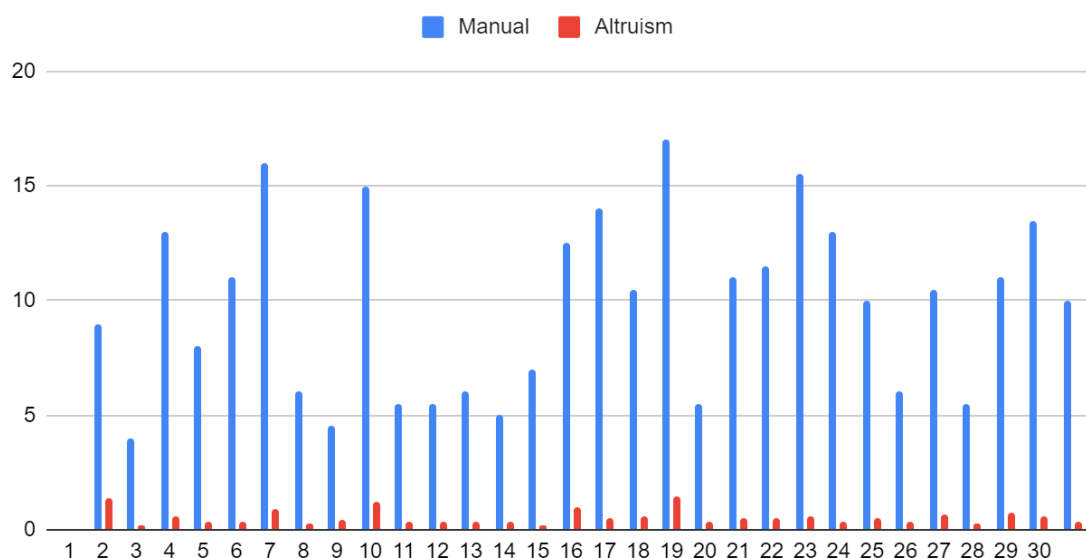


Figura 5.18 - Comparação entre o Downtime do altruísmo e o manual.

Perda Financeira (P.F)

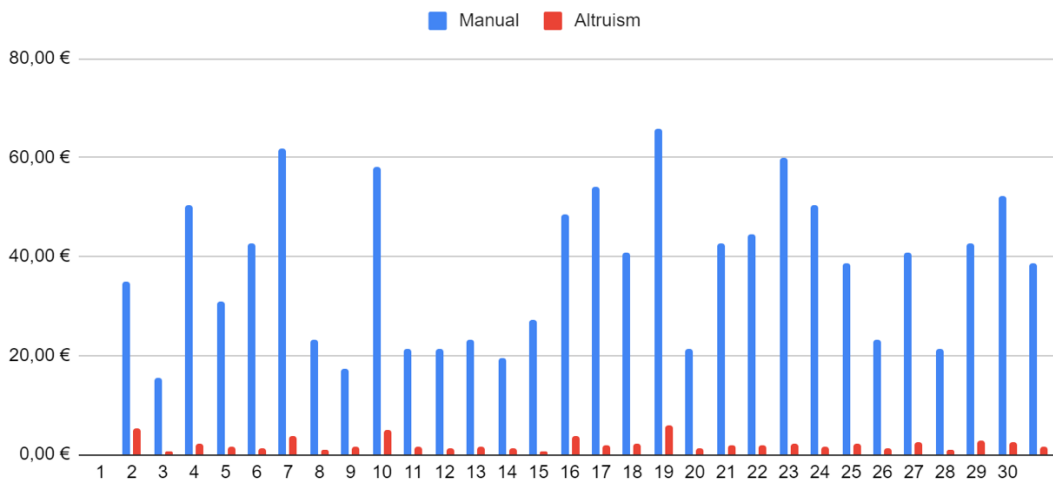


Figura 5.19 - Comparação entre a Perda financeira do altruísmo e o manual.

Tabela 5.14 - Comparação das médias do processo manual e altruístico.

MÉDIA GERAL		
PROCESSO	DOWNTIME	P.F
MANUAL	9,75	37,75 €
ALTRUÍSMO	0,56	2,16 €

Desta forma pode-se concluir que em média perde-se 9,75 minutos por cada processo realizado manualmente, sendo equivalente à €37,75 de perda financeira.

Os cálculos realizados para a obtenção dos resultados apresentados nas Fig. 5.18 e 5.19, que resultaram nos valores da Tabela 5.14, foram feitos assumindo que, durante o acto de altruísmo, a partilha de energia é instantânea, ou seja, $t_c = 0$ (t_c , tempo de partilha de energia do altruísmo). Assim, para tornar os resultados mais próximos da realidade, nas Fig. 5.20 e 5.21 são apresentados novos resultados (PF) com base em tempos de partilha de 60 segundos e 180 segundos, conforme ilustrado nas Tabelas 5.15 e 5.16.

Perda Financeira (P.F)

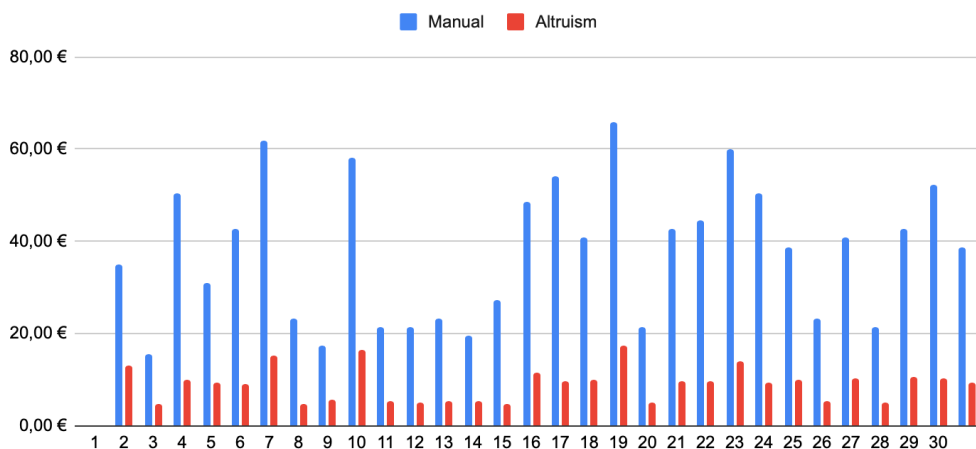


Figura 5.20 - Comparação entre a Perda financeira do altruísmo e o manual com tempo de partilha de 60s.

Tabela 5.15 - Comparação das médias do processo manual e altruístico com tempo de partilha de 60s.

MÉDIA GERAL		
PROCESSO	DOWNTIME	P.F
MANUAL	9,75	37,75 €
ALTRUÍSMO	2,32	8,99 €

Perda Financeira (P.F)

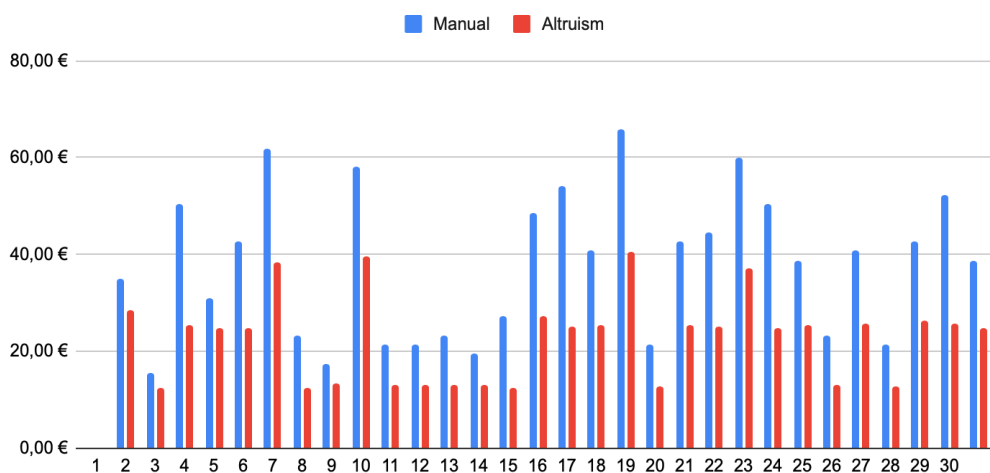


Figura 5.21 - Comparação entre a Perda financeira do altruísmo e o manual com tempo de partilha de 180s.

Tabela 5.16 - Comparação das médias do processo manual e altruístico com tempo de partilha de 180s.

MÉDIA GERAL		
PROCESSO	DOWNTIME	P.F
MANUAL	9,75	37,75 €
ALTRUÍSMO	5,86	22,67 €

As Tabelas 5.15 e 5.16 apresentam novos dados que nos permitem concluir que, ao assumir que o tempo de partilha utilizando as tecnologias emergentes de carregamento sem fio, para a transferência de uma pequena porção de energia entre dois robôs, seja de 60 segundos, verifica-se uma perda de 9,75 minutos, o que corresponde a uma perda financeira de €37,75 por cada processo realizado manualmente. Por outro lado, no processo de altruísmo, a perda é de apenas 2,32 minutos, equivalente a €8,99. Considerando um tempo de 180 segundos, a perda no processo manual permanece inalterada, enquanto no altruísmo a perda aumenta para 5,86 minutos, correspondendo a €22,67. Em ambas as situações, o processo de altruísmo apresenta uma poupança financeira significativa para as indústrias.

É importante lembrar que, nos testes realizados, foram considerados apenas dois robôs solicitantes. No entanto, em um ambiente real, podem ocorrer várias substituições de baterias ou carregamentos, o que aumentaria a necessidade dos atos de altruísmo em casos específicos. Como consequência, o downtime causado pelo processo manual seria ainda maior, uma vez que o processo de troca de baterias é realizado de forma sequencial, aumentando o tempo de espera em cada estação de produção. Por outro lado, o altruísmo pode ocorrer simultaneamente, o que resulta em uma redução do tempo de espera e, consequentemente, do tempo de inatividade (downtime) nas estações de produção.

5.4 Alternativas de carregamento dos AGV's

A automação tem-se tornado um dos temas mais relevantes nesta nova revolução industrial e nas indústrias inteligentes. Assim, muitas empresas atualmente têm implementado os robôs móveis autoguiados (AGV), que podem operar dentro do ambiente de produção sem a constante intervenção humana. Um dos temas mais questionados sobre a independência desses robôs é a forma de carregamento de suas baterias. Algumas empresas consolidadas possuem projetos para solucionar essa questão, mas todas elas diferem do projeto apresentado neste trabalho. A solução proposta neste projeto, focada na implementação do altruísmo entre os robôs, apresenta várias características positivas em relação aos projetos do mercado, mas também tem suas desvantagens. Na Tabela 5.17, é apresentada uma comparação entre os diversos projetos desta área, considerando os seguintes critérios:

- **Rapidez de carregamento:** Trata-se do tempo que pode levar até o carregamento da bateria. Para o caso do altruísmo considerado neste trabalho, a partilha de carga é

feita entre 10 e 25% de um AGV para o outro. Neste caso, dura menos tempo em relação aos sistemas que fazem o carregamento até o *full power*.

- **Tempo de espera da estação:** Refere-se ao tempo que uma estação que depende daquele robô pode ficar parada enquanto o robô é carregado.
- **Produtividade:** É a eficiência do sistema de produção em geral segundo os benefícios que o projeto oferece.
- **Carregamento automático:** Refere-se à forma em que as baterias podem ser carregadas. Automática significa que o robô não precisaria do auxílio de um especialista. Geralmente a forma automática produz mais vantagens em relação à manual.
- **Oportunidade de carregamento:** Refere-se a chance que um robô tem de ser atendido ou carregado quando a sua bateria chega a um nível considerado crítico. Em termos de ter uma estação fixa de carregamento, pode haver filas de AGV's aguardando para obter carga, neste caso seria preciso um número muito grande de estações para que todos os robôs sejam atendidos sem muita espera.
- **Duração de trabalho:** Refere-se ao tempo de trabalho que o AGV pode operar com a carga recebida.

Diante destes critérios, foram comparadas as seguintes soluções existentes no mercado em comparação ao altruísmo:

1. **Daihem D-Broad:** Oferece uma solução baseada em uma estação de carregamento fixo em que os robôs se dirigem até à estação para se autocarregar. Essa tecnologia permite uma forma de carregamento sem fio por ressonância magnética.
2. **Wiferion:** É uma tecnologia de carregamento sem fio focada em AGV's. Diferente da D-Broad, ela oferece uma solução de carregamento baseada em um robô móvel na forma de uma estação de carregamento móvel dentro do ambiente colaborativo.
3. **Wewo techmotion:** É uma solução que oferece a opção de troca de bateria, carregamento manual e oportunidade de carregamento automático baseada em uma estação de carregamento fixo.
4. **Método manual:** Refere-se à forma de carregamento manual em que quando o AGV atinge o limite da carga, espera por um especialista humano para lhe dar suporte quer levando-o para a estação de carregamento ou fazendo a troca da bateria no local.

Tabela 5.17 - Comparação das formas de carregamentos dos AGV's.

Tecnologias / Critérios	Rapidez de carregamento	Tempo de espera da estação	Produtividade	Carregamento automático	Oportunidade de carregamento	Duração de trabalho
Altruísmo	✓	✓	✓	✓	✓	✗
Daihem D-Broad	✗	✗	✓	✓	---	✓
Wife-rion	✓	✗	✓	✓	---	✓
Wewo techmotion for AGV	✗	✗	✓	✓	---	✓
Método manual	✗	✗	✓	✗	---	✓
✓ : Positivo						
✗ : Negativo						
--- : Relativo						

De reforçar que a opção “Relativo” na coluna “Oportunidade de Carregamento” significa que pode ser uma boa opção caso haja várias estações de carregamento na área, mas pode ser desfavorável caso contrário.

Com o objetivo de dinamizar o processo de industrialização e reduzir a necessidade constante de um especialista, o altruísmo entre os robôs dentro do ambiente colaborativo oferece diversas vantagens, não apenas pela diminuição da supervisão, mas também em questões económicas, como foi apresentado anteriormente. Embora seja uma excelente solução, também possui algumas desvantagens, em particular o tempo de duração da bateria do robô após receber ou partilhar a sua carga, conforme mostrado na Tabela 5.13 acima. Esta é uma limitação que pode ser resolvida de várias maneiras, incluindo a integração de soluções oferecidas pelas empresas que atuam neste mercado, como, por exemplo, estações de carregamento sem fio, entre outras opções.

CONCLUSÃO

Em suma, este trabalho apresentou uma solução viável para beneficiar o ambiente colaborativo no sistema de fabricação, envolvendo robôs capazes de agir de forma altruísta uns com os outros. Com os avanços da Indústria 4.0 e o crescimento da autonomia dos robôs, o altruísmo entre os robôs pode se tornar uma necessidade, visto que a tendência é que eles se adaptem e se auxiliem sempre que possível, em diversos ambientes. Desta forma, com base nos capítulos desenvolvidos durante a elaboração do projeto, o trabalho respondeu às questões e à proposta apresentadas sobre o altruísmo dentro do ambiente colaborativo. Foi capaz de demonstrar e simular como os robôs podem partilhar informações, como localização e solicitações de ajuda, para se apoiarem mutuamente e superar falhas (como falta de energia na bateria ou a quebra de uma ferramenta), permitindo que a produção continue ao mesmo ritmo. O caso de estudo apresentado mostrou que o altruísmo pode ser utilizado para criar um ambiente colaborativo no sistema de fabricação de sensores da Senseair.

Este projeto contribuiu na base do *“altruísmo entre os robôs dentro do ambiente colaborativo”* tornando esse conceito mais firme e como pode beneficiar a produção da empresa. Essa contribuição inclui:

- Implementação do comportamento altruístico entre os robôs por meio do ROS2.
- Algoritmos em forma de biblioteca para a aplicação do altruísmo.
- Simulação do robô no contexto de altruísmo.
- Apresentação de um caso de estudo.

O teste e validação dos resultados do altruísmo mostraram que este processo possui maior eficiência em relação ao processo manual utilizado em muitas empresas. Além de permitir que se economiza uma grande quantia financeira anualmente, visto que estas atividades de troca de bateria podem acontecer com mais frequência dependendo da quantidade de robôs, o altruísmo também faz um bom proveito dos pequenos instantes que os robôs ficam aguardando por uma nova tarefa.

Essas vantagens que o altruísmo entre os robôs dentro do ambiente colaborativo apresenta, tem como principal foco diminuir a interrupção ou paragem no processo de produção

causadas pelos robôs, contribuindo para a ascensão do novo formato da industrialização que chamamos de *Smart manufacturing* ou indústrias inteligentes.

6.1 Futuro trabalho

Para uma futura continuação do trabalho, deve-se focar na extensão da arquitectura do altruísmo e nas limitações que o trabalho apresenta até o atual momento:

- Inexistência de uma alternativa para o carregamento da bateria dos robôs cujas solicitações não são respondidas por falta de candidatos.
Sugestão: a implementação de uma tecnologia semelhante à Wiferion para dar suporte a todos robôs não atendidos pelo altruísmo seria uma ótima solução fazendo com que uma estação móvel vá até o ponto em que o robô se encontra para poder atribuir-lhe energia da bateria de longa duração. O que será necessário que ele tenha acesso aos dados trocados no processo altruístico e assim poder localizar aqueles que não
- Forma de carregamento que os robôs usariam para poder trocar energia da bateria entre si com maior eficiência.
- **Sugestão:** a princípio uma tecnologia de carregamento sem fio talvez seria uma ótima solução, visto que se pretende não envolver um especialista humano na operação.
- Implementação e simulação do altruísmo baseado na troca de ferramentas, desde os tipos de ferramentas até o processo detalhado de como seria realizada essa troca sem a intervenção humana.

Além disso, a arquitectura proposta e sua implementação poderia ser testada dentro de um ambiente industrial diferente da abordagem apresentada, envolvendo robôs reais se possível.

BIBLIOGRAFIA

- [1] M. O. Cardoso. (2014). Indústria 4.0: a quarta revolução industrial [Online]. Available: https://riut.utfpr.edu.br/jspui/bitstream/1/17086/1/CT_CEAUT_2015_08.pdf.
- [2] M. Xu, J. M. David and S. H. Kim. (2018, March). "The Fourth Industrial Revolution: Opportunities and Challenges." *International Journal of Financial and Research* [Online]. vol. 9, no. 2. Available: https://www.researchgate.net/publication/323638914_The_Fourth_Industrial_Revolution_Opportunities_and_Challenges.
- [3] N. Davis, "What is the fourth industrial Revolution", *Journal of International Affairs Editorial Board*, vol. 19, pp. 17-22, January 2016.
- [4] E. Ricardo. (2016). Lós 9 pilares de la Indústria 4.0 [Online]. Available: <http://www.ecointeligencia.com/2016/06/9-pilares-industria-40-1/>.
- [5] G. Nuno, "Cyber segurança na Indústria Nacional. Robótica", *Porto: CIE*, n. 114, p.56-62, March 2019.
- [6] Fiebe. (2020). Realidade Aumentada [Online]. Available: http://portais.fieb.org.br/portal_ead/inovacao-tecnologica/realidade-aumentada.html.
- [7] G. Shao, S. J. Shin, and S. Jain. Data analytics using simulation for smart manufacturing. *Proceedings of the Winter Simulation Conference, IEEE*. pp. 2192-2203. 2014, December.
- [8] G. Pedro. (2017). Robôs Autônomos [Online]. Available: <https://www.pedrogarcia12av1.wordpress.com/about/robos-autonomos/>.
- [9] T. D. Ngo, A. Kashani, G. Imbalzano, K. T. Q. Nguyen and D. Hui, "Additive manufacturing (3D printing): A review of materials, methods, applications, and challenges", *Composites Part B: Engineering*, no. 143, 172–196. 2018.
- [10] Google Cloud. (2018). What is cloud computing [Online]. <https://cloud.google.com/learn/what-is-cloud-computing?hl=pt-br>.
- [11] Haug, K. Candel, T. Kretschmer and S. Thomas. (2019) Cloud Adaptiveness Within Industry Sectors—Measurement and Observations [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0308596115001238>.
- [12] A. Silva, C.E.S. et al. (2007). Contribuição da análise do valor da simulação da manufatura [Online]. Available: https://www.researchgate.net/publication/309035132_CONTRIBUICAO_DA_ANALISE_DO_VALOR_NA_SIMULACAO_DA_MANUFATURA.

- [13] Arktis. (2015). Indústria 4.0, a Quarta Revolução Industrial [Online]. Available: <http://arktis.com.br/a-quarta-revolucao-da-industria>.
- [14] B. P. Santos, L. A. Silva, C. S. Celes, J. B. Borges, B. S. Peres, M. M. Vieira, A. A. Loureiro, Internet das Coisas: da Teoria á Prática, *Minicursos*, vol.1, pp. 15. Maio, 2016.
- [15] Ros2 Documentation. (2021). Understand ROS2 Nodes [Online]. Available: <https://docs.ros.org/en/iron/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html>.
- [16] S. Cousins, "Exponential growth of ROS Topics", *IEEE Robotics & Automation Magazine*, vol.1, pp. (18):19–20, 2011.
- [17] K. Wyobek, E. Berger, H. V. der Loos, and K. Salisbury, "Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot", *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2008.
- [18] M. Quigley, E. Berger and A.Y. Ng, "STAIR: Hardware and Software Architecture," *AAAI 2007 Robotics Workshop, Vancouver, B.C*, August, 2007.
- [19] G. Byrne, D. Dimitrov, L. Monostori, R. Teti, F. Houten and Rafi Wertheim. (2018, May) "Biologicalisation: Biological transformation in manufacturing". *Science Direct* [Online]. vol. 21, pp. 2-4. Available: <https://www.sciencedirect.com/science/article/abs/pii/S1755581718300129>.
- [20] P. Miller. (2007, July). "The Genius of Swarms". *National Geographic* [Online]. Available: <https://nphsworld16huggins.wordpress.com/wp-content/uploads/2015/11/nphsemergencysubplans-1116.pdf>.
- [21] N. Kubota, T. Fukuda, F. Arai and K. Shimojima, "Genetic algorithm with age structure and its application to self-organizing manufacturing system", *Emerging Technologies and Factory Automation*, pp. 472-477, 1994.
- [22] K. Rajeshwar, P. J. Hesketh, "Biomimetic or Bioinspired?", *The Electrochemical Society*, vol. 1, pp. 1, 2014.
- [23] PasserioDireto. (2020, August). Desenvolvimento industrial e os impactos causados ao meio ambiente [Online]. Available: <https://www.passeidireto.com/arquivo/127744615/geografia>.
- [24] R. Pfeifer, M. Lungarella and F. Iida, "Self-organization embodiment and biologically inspired robotics", *Science*, vol. 318, no. 5853, pp. 1088-1093, 2007.
- [25] G. M. Serugendo, M. P. Gleizes and A. Karageorgos, "Self-Organising Software: From Natural to Artificial Adaptation", *Berlin, Germany: Springer*, vol. 2, pp. 7-32, May 2011.
- [26] Articalist on render. (2016, January). Nine pillars of industry 4.0 [Online]. Available: <https://articalist.onrender.com/nine-pillars-of-industry-4-0.html>.
- [27] U.G.K. Wegst, H. Bai, E. Saiz, A.P. Tomsia and R.O. Ritchie, "Bioinspired structural materials", *Nat Mater*, vol. 14, pp. 23 - 36, October 2015.
- [28] S. V. D. Leeuw, "Undoing Unintended Consequences for the Earth's Survival", *Social Sustainability Past and Future*, vol.1, pp. 1, May 2020.

- [29] B. W. Jeon, S. C. Yoon and S. Suk-Hwan. (2016). An architecture design for smart manufacturing execution system. *Computer-Aided Design and Applications* [Online]. Available: <https://doi.org/10.1080/16864360.2016.1257189>.
- [30] L. A. Estrada-Jimenez, S. Kalateh, S. Nikghadam-Hojjati, J. Barata, "An Altruistic-Based Framework to Support Collaborative Healing of Manufacturing Resources in a Self-Organized Shop-Floor", *UNINOVA*, vol.12, pp. 42882 - 4298 January 2024.
- [31] R. Cioffi, M. Travaglioni, G. Piscitelli, A. Petrillo and F. De Felice. (2020, January, 08). Artificial intelligence and machine learning applications in smart production: progress, trends, and directions. *Sustainability* [Online]. Available: [doi.org. https://doi.org/10.3390/su12020492](https://doi.org/10.3390/su12020492).
- [32] KUKA. (2020). Matrix Production: An Example for Industrie 4.0 [Online]. Available: <https://www.kuka.com/en-de/industries/solutions-database/2016/10/matrix-production>.
- [33] L. A. Estrada-Jimenez, T. Pulikottil, S. Nikghadam-Hojjati, and J. Barata, "Self-organization in smart manufacturing - Background, systematic review, challenges and outlook," *IEEE Access*, vol. 11, pp. 10107–10136, 2023.
- [34] Psaiar, H., & Dustdar, S. (2011). "A survey on self-healing systems: Approaches and systems. *Computing*, 91(1), 43–73". doi.org. <https://doi.org/10.1007/s00607-010-0107-y>Return to ref 2011 in article.
- [35] R.L. Trivers, "The evolution of reciprocal altruism", *Q. Rev. Biol.*, vol. 46(1), pp. 35–57. January 1971.
- [36] J.J.M. Massen, M. Lambert, M. Schiestl and T. Bugnyar. (2015). Subadult ravens generally don't transfer valuable tokens to conspecifics when there is nothing to gain for themselves [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4484978/>.
- [37] M. Dorigo et al., "Ant colony optimization theory: A survey", *Theoret Comput Sci.*, Vol. 344, pp. 243-278, November 2005.
- [38] K. Altisen, S. Devismes, A. Durand, C. Johnen and F. Petit, "Self-stabilizing systems in spite of high dynamics", *Proceedings of the 22nd International Conference on distributed computing and networking*, vol.1, pp. 156–165, February 2021.
- [39] J. C. Chen, T. Chen, B. R. Pratama and Q. Tu, "Capacity planning with ant colony optimization for tft-lcd array manufacturing" *Journal of Intelligent Manufacturing*, vol. 1, pp. 1–19, May 2016.
- [40] Y. Song, Y. Guan, "Design and Modeling of a Novel Biomimetic Variable Stiffness Actuator Inspired by Skeletal Muscle", *IEEE International Conference on Robotics and Biomimetics*, vol. 4, pp. 6 – 20, December 2021.
- [41] R. A. Brooks, "The artificial life route to artificial intelligence: Building embodied, situated agents", *Intelligence without reason. In L. Steels & R. Brooks (Eds.)*, vol. 3, pp. 25–81, February 2001.
- [42] X. Yang and S. Deb, "Engineering optimisation by cuckoo Search", *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, pp. 330–343, April 2010.

- [43] M. K. Sayadi, R. Ramezani, and N. Ghaffari-Nasab, "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems", *International Journal of Industrial Engineering Computations*, vol.1, pp. 1–10, January 2010.
- [44] J. D. Ferreira, L. Ribeiro, M. Onori, J. Barata, "Challenges and Properties for bio-inspiration in Manufacturing", *Doctoral Conference on Computing, Electrical and Industrial Systems*, vol. 1, pp.139-148, 2014.
- [45] L. Ribeiro and J. Barata, "Self-organizing multiagent mechatronic systems in perspective", *11th IEEE International Conference on Industrial Informatics*, vol. 1, pp. 392-397, July, 2013.
- [46] Cloud Google. (2020, January). Big Data [Online]. Available: <https://cloud.google.com/learn/what-is-big-data?hl=pt-br>.
- [47] Infopédia Dicionários Porto. (2024). Altruísmo [Online]. Available: [infopedia.pt. https://www.infopedia.pt/dicionarios/lingua-portuguesa/Altruismo](https://www.infopedia.pt/dicionarios/lingua-portuguesa/Altruismo).
- [48] J. Dias-Ferreira, L. Ribeiro, H. Akillioglu, P. Neves and M. Onori, "Bioinspired self-organising architecture for manufacturing cyber-physical shopfloors", *Journal of Intelligent Manufacturing*, vol. 29, pp. 1659-1682, March, 2018.
- [49] R. Pannequin and A. Thomas, "Another interpretation of stigmergy for product-driven systems architecture", *Journal of Intelligent Manufacturing*, vol. 23, pp. 2587-2599, June, 2012.
- [50] A. D. Rocha, P. Lima-Monteiro, M. Parreira-Rocha and J. Barata, "Artificial immune systems based multi-agent architecture to perform distributed diagnosis", *Journal of Intelligent Manufacturing*, vol. 30, pp. 2025-2037. February, 2019.
- [51] Daihen. (2023). Wireless charge system [Online]. Available: <https://www.daihen.co.jp/en/d-broad/>.
- [52] Cad onshape. (2024). Robotis turtlebot3 Waffle Pi [Online]. Available: <https://cad.onshape.com/documents/179d4ba2a1f9231f93cae6e2/w/71e9066aee42d5039f880e2f/e/3cb7d487c93655e7067a3d5f>.
- [53] Open Navegation. (2024). Nav2 documentation [Online]. Available: <https://docs.nav2.org>.
- [54] AGV Network. (2022). Maximum AGV and AMR speed [Online]. Available: <https://www.agvnetwork.com/agv-technology/agv-speed>.
- [55] FlexQube. How to swap an AGV Battery [Online]. Available: <https://www.youtube.com/watch?app=de-sktop&v=QPbyW0VK9mY>.
- [56] Senseair. (2024). Senseair Products [Online]. Available: <https://senseair.com/products/>.

