



Estratégias de Controlo para Sistemas Robóticos Cooperativos

Sumário pormenorizado do Seminário apresentado no âmbito de provas de
agregação

Nuno Miguel Fonseca Ferreira

Vila Real, março de 2019

Sumário pormenorizado do Seminário apresentado no âmbito de Provas de Agregação, de harmonia com o disposto na alínea c), do número 2, do artigo 8º do Decreto-Lei nº 239/2007, de 19 de junho de 2007.

Índice

Lista de Acrónimos.....	iii
Contexto.....	1
1. Breve Introdução aos Sistemas Robóticos Autónomos e Cooperativos	3
2. A Utilização de Múltiplos Robôs em Aplicações de Cooperação e Coordenação... ..	7
3. Métodos de Localização de Sistemas Robóticos.....	13
4. Estratégias de Controlo e Coordenação.....	19
5. Caso de Estudo	23
5.1 Inteligência Coletiva do Tipo Enxame	23
6. Trabalhos futuros	35
Anexos	35
Anexo 1 – Filtro de <i>Kalman</i>	39
Anexo 2 – Algoritmo <i>FastSLAM</i> (Filtro de partículas Rao-Blackwellized).....	43
Anexo 3 – Modelo do enxame de robôs	51
Anexo 4 – Cinemática do robô diferencial.....	53
Referências Bibliográficas.....	55

Lista de Acrónimos

DPSO – Darwinian Particle Swarm Optimization

EKF – Extended Kalman Filter

MANET – Mobile Ad-hoc NETWORKs

MAS – MultiAgent Systems

MMW – Millimeter-Wave

MR-MT – MultiRobot MultiTask

MRS – MultiRobots Systems

PSO – Particle Swarm Optimization

RDPSO – Robot Darwinian Particle Swarm Optimization

ROS – Robot Operating System

SaR – Search and Rescue

SLAM – Simultaneous Localization And Mapping

SRC – Sistemas Robóticos Cooperativos

Contexto

Este seminário tem como objetivo apresentar estratégias de controlo para a robótica autónoma cooperativa e colaborativa. Parte dos conteúdos abordados no seminário foram adaptados de publicações escritas pelo autor e por outros investigadores. O material apresentado neste documento pode ser adaptado de forma a enquadrá-lo numa unidade curricular na área da Robótica, destinada a alunos de Mestrado e/ou de Doutoramento. Exigem-se conceitos básicos adquiridos nas unidades curriculares dos estudos de Licenciatura em Engenharia Eletrotécnica e de Computadores ou áreas afins.

O assunto deste seminário não é tratado de forma completa em nenhum manual ou livro. Ao longo do documento indicam-se várias publicações que abordam os diferentes assuntos procurando realçar os avanços na investigação desta última década, incluindo a experiência adquirida pelo autor em trabalhos de investigação realizados, nomeadamente artigos científicos, projetos científicos e tecnológicos, na orientação de alunos e ainda com empresas. O autor pretende mostrar a atualidade do tema e motivar os alunos à iniciação da atividade de investigação científica, em aplicações da robótica autónoma cooperativa e colaborativa. Para além dos conceitos gerais expostos neste documento, algumas das técnicas e métodos foram desenvolvidos ou adaptados com a finalidade de serem utilizadas em trabalhos de investigação sob a orientação do autor.

Uma pergunta que pode ser feita aos alunos sobre a robótica cooperativa e colaborativa, “*À primeira vista ter múltiplos robôs a executarem uma tarefa ou um conjunto de tarefas em comum pode não parecer propriamente útil. Porque não utilizar um único robô mais complexo capaz de realizar todas essas tarefas?*”

Alguns dos desenvolvimentos na área dos Sistemas Robóticos Cooperativos (SRC) citam os sistemas biológicos como fonte de inspiração, onde podemos obter resposta a esta pergunta. O comportamento coletivo de formigas, abelhas, pássaros e outras sociedades similares fornecem fortes indícios de que os sistemas compostos por agentes simples conseguem desempenhar tarefas complexas no mundo real.

A robustez e a capacidade de adaptação dos sistemas biológicos representam uma motivação para esses mecanismos com as mesmas características. Estas e muitas outras razões serão abordadas neste seminário evidenciando as vantagens dos sistemas robóticos cooperativos face a um único robô.

Os SRC descrevem a situação em que um determinado grupo de robôs obtém um benefício global. Uma primeira questão fundamental na cooperação consiste em saber se os robôs devem ser idênticos (grupos homogêneos) ou diferentes (grupos heterogêneos) e uma outra questão, se a eficiência deve entrar em consideração com o desempenho de todo o grupo ou apenas de cada robô individualmente.

Desta forma, a secção 1 começa com uma breve introdução dos Sistemas Robóticos Cooperativos. Na secção 2 apresentam-se algumas aplicações de sistemas robóticos autónomos. De seguida, na secção 3, introduzem-se os modelos de localização dos sistemas robóticos. Na secção 4, apresentam-se as estratégias de controlo e coordenação, e por último, é apresentado um caso de estudo onde se exemplifica a aplicabilidade das estratégias e algoritmos de controlo. O caso de estudo descrito tem como finalidade ilustrar como podem ser usados os métodos apresentados na resolução de problemas de engenharia.

As metodologias esboçadas neste documento apontam para um avanço expressivo no estado da arte no domínio da robótica cooperativa e colaborativa, destacando-se o carácter fortemente inovador dos sistemas estudados.

1. Breve Introdução aos Sistemas Robóticos Autónomos e Cooperativos

Os sistemas robóticos podem ser classificados segundo o meio em que operam (subaquáticos, aquáticos de superfície, terrestres e aéreos), a sua forma de locomoção (rodas, pernas, hélices, etc.) e o grau de atuação. No entanto em todos eles, procura-se agregar capacidades sensoriais de perceção, tanto do sistema robótico em si quanto do meio onde ele evolui, visando o estabelecimento de estratégias de operação autónomas.

Alguns exemplos de aplicações contam-se entre: a exploração de ambientes inacessíveis; a inspeção em diferentes contextos; aplicações em agricultura; o transporte e manuseio de cargas; o apoio em casos de sinistros naturais, ou os sistemas de auxílio à condução, entre outros.

A utilização de sistemas robóticos autónomos e cooperativos, à semelhança do que ocorre com o ser humano, permite efetuar tarefas que dificilmente seriam executadas por apenas um indivíduo. A exigência do controlo de posição e orientação é crucial, assim como um bom sistema de comunicação entre os robôs, para a execução da missão [Marayong, 2007].

Um sistema robótico cooperativo, requer para o seu desenvolvimento e operação, uma infraestrutura compreendendo basicamente, uma estação de controlo no sistema robótico, uma estação de operação e um sistema de comunicação entre ambas. Isso implica em termos de *hardware* e *software* que devam satisfazer diferentes requisitos de integração, tais como o processamento, a operação em tempo real ou a robustez, quer seja na partilha de soluções, na troca de informações ou ainda na extensão das metodologias para a coordenação de sistemas robóticos. [Chaimowicz *et al.*, 2004, Hyondong *et al.*, 2017].

Esse tipo de cooperação pode variar desde existirem apenas dois robôs a executarem uma simples tarefa em conjunto (e.g., dois braços industriais a manipularem um objeto de grandes dimensões) [Fonseca Ferreira, 2006] até um grupo de agentes robóticos heterogêneos capazes de se juntarem e formarem uma estrutura mais complexa [Fukuda, 1989]. A integração de técnicas de navegação e localização tem vindo a focar o interesse nos sistemas robóticos cooperativos, usufruindo dessa forma de todas as vantagens inerentes à cooperação entre robôs.

Na perspectiva de sistemas robóticos autónomos, é certamente necessário um esforço multidisciplinar e uma fundamentação para os dotar com mais autonomia. Por exemplo, o desenvolvimento de métodos de navegação e de controlo, bem como o de técnicas de coordenação que permitem aumentar a autonomia dos sistemas robóticos cooperativos [Oh *et al.*, 2017].

Em muitas das aplicações de vigilância, em particular em interiores de edifícios, o interesse está centrado na atividade humana através da interpretação automática dessas ações sendo, muitas destas aplicações realizadas em ambientes bem estruturados, com condições e restrições específicas e bem definidas. Associado ao problema da análise de atividade existe o reconhecimento de ações, que é sobretudo, um problema de classificação, ao qual diversos métodos são propostos, alguns requerendo, ou não, um processo prévio de aprendizagem. Nas operações de mapeamento muitas das soluções são dirigidas ao problema com o intuito de estimar um mapa global através de um conjunto de imagens observadas. Estas técnicas de mapeamento normalmente não utilizam informação fornecida por sensores complementares [Kelley *et al.*, 2008].

A estimativa destes mapas é baseada em técnicas de Bayes, que são utilizadas em diversas aplicações em que múltiplos robôs podem operar de maneira coordenada e de forma cooperativa [Thrun *et al.*, 2005]. Essa possibilidade torna-se viável somente se os diferentes aspetos referentes ao controlo nos diversos níveis tenham sido corretamente abordados e estejam integrados de forma robusta.

Na coordenação de múltiplos robôs autónomos em contexto de condução autónoma, o desenvolvimento de cenários pilotos do tipo *duckietown* [Paull *et al.*, 2017], permite desenvolver capacidades na modelação de sistemas robóticos autónomos (modelos cinemático e dinâmico) e na integração de sensores possibilitando para além disso, estabelecer erros de medida, bem como a construção do mapa e planeamento das trajetórias dos múltiplos robôs [Rosman *et al.*, 2017].

Os desafios científicos e as possibilidades da criação de novas estratégias e algoritmos quer para a deteção quer para a representação do ambiente, são muitas sendo mesmo possível considerar todas as limitações impostas tais como, o campo de visão, a comunicação ou a iluminação entre outras [Liu *et al.*, 2017]. Além destes existe também a possibilidade da

navegação numa pequena rede rodoviária com a integração sensorial para a deteção de objetos (carros, sinais de trânsito, semáforos), de efetuar a localização global, bem como estudar as reações aos obstáculos, desta forma, é possível fazer o planeamento de modo a executar uma determinada missão [Page *et al.*, 2017].

A atividade educativa e de investigação em Portugal já é considerável na área dos sistemas robóticos, existindo já algumas iniciativas nacionais competitivas, como é o caso na UTAD do *Micromouse Portugues Contest* [<https://www.micromouse.utad.pt/>] que procuram fomentar a investigação e desenvolvimento neste campo. São de referir as competições existentes no Festival Nacional de Robótica, que estimulam os mais jovens a iniciarem as suas atividades nesta área de conhecimento e na robótica colaborativa onde se pode destacar o campeonato do Futebol Robótico.

2. A Utilização de Múltiplos Robôs em Aplicações de Cooperação e Coordenação

Algumas tarefas robóticas, especialmente aquelas que são intrinsecamente distribuídas e complexas necessitam, para viabilizar a realização da missão, com um desempenho melhor, mais do que um robô. Quando comparado com soluções baseadas em um único robô, os sistemas robóticos cooperativos têm importantes vantagens relacionadas com a distribuição do tempo e do espaço, a decomposição de problemas complexos em problemas menores, o grau de confiança, a robustez e o custo. A distribuição espacial é uma importante vantagem da utilização de múltiplos robôs, até porque podem ocupar diversos lugares ao mesmo tempo. Desde que, cada robô apresente um conjunto de recursos tais como sensores, capacidade de percepção, poder computacional, comunicação, locomoção e serem suficientemente autônomos para realizar parte das tarefas, um Sistema de Múltiplos Robôs (MRS) apresenta diversas vantagens.

Os sistemas de cooperação de múltiplos robôs têm recebido grande atenção pelos investigadores da comunidade robótica na última década, e sua utilização tem inquestionável relevância social e económica em diversos domínios. A figura 1 apresenta um exemplo de utilização de um sistema robótico numa missão de transporte.



Figura 1 – Robô a ser utilizado no transporte de objetos.

Os desenvolvimentos tecnológicos e a facilidade na construção de robôs tornaram possíveis a substituição de pessoas por robôs em ambientes de risco, que compreendiam tarefas

perigosas para os seres humanos, por exemplo. Num cenário de risco é tolerável perder um ou mais robôs mas que em nenhuma situação venha a ocorrer com pessoas.

Além disso, em cenários menos perigosos, este tipo de sistemas robóticos cooperativos podem ser utilizados em tarefas monótonas e repetitivas (vigilância, patrulhamento, entre outras), permitindo assim que as pessoas façam tarefas mais nobres (coordenar os grupos de robôs, desenvolver ou criar novas tarefas, por exemplo) [David Portugal *et al.*, 2013]. Na figura 2 a) pode observar-se robôs a montar uma cadeira.

Nos MRS, a operação simultânea de robôs permite a realização de diversas tarefas, análogas ou complementares e diferentes. Levando em consideração que pode haver distribuição do poder computacional pelos múltiplos robôs, certos problemas complexos, podem ser decompostos em problemas mais simples, que podem ser alocados a cada robô individualmente. Consequentemente, uma missão complexa pode ser designada a uma equipa de robôs inteligentes, que decompõem a missão em tarefas simples e cooperam para alcançar o objetivo final. Um exemplo disso é o futebol robótico, como se ilustra na figura 2 b).



Figura 2 – Cooperação e coordenação entre robôs [MIT, 2015]: a) Montagem de uma cadeira e b) Futebol Robótico [Rule Book, 2018].

Em missões complexas, é necessário uma quantidade considerável de recursos distintos (diferentes capacidades de sensores, mobilidade na terra ou no ar, entre outros), os sistemas robóticos cooperativos permitem uma maior flexibilidade através da distribuição das tarefas e diminuindo assim o risco da tarefa não ser realizada. Se os recursos necessários para executar uma missão complexa são distribuídos em diferentes robôs com capacidades heterogêneas, logo apresentam alguma redundância, a equipa de robôs acaba por ser mais robusta e confiável, porque mesmo que aconteça alguma falha ou algum robô sofra alguma avaria, a missão ainda pode ser completada pelos restantes, embora com degradação do desempenho da equipa. Considerando que cada robô da equipa possa ser mais simples, e

consequentemente, mais confiável do que um único robô com todas as capacidades necessárias para a missão, um sistema com múltiplos robôs pode até apresentar custos mais reduzidos.

A aceitação de qualquer sistema robótico é altamente dependente da sua confiabilidade, especialmente em aplicações críticas como segurança, missões militares, salvamento, exploração espacial, entre outras. Além disso, existem certos tipos de tarefas que requerem a utilização de múltiplos robôs, em geral essas tarefas não podem ser executadas por um único robô.

Pode-se afirmar que existem tarefas fortemente acopladas, que requerem que os robôs tenham informações sobre o estado dos outros, além disso é necessário sincronizar as ações dos robôs. Neste tipo de cooperação, cada membro da equipa é crítico na execução da tarefa e é fundamental que exista comunicação entre os robôs. Por outro lado, existem tarefas fracamente acopladas que podem ser executadas por um único robô. Em termos de desempenho e tolerância a falhas, quando comparados com múltiplos robôs apresentam piores desempenhos. Em geral, tarefas fracamente acopladas não exigem um alto nível de coordenação entre os membros da equipa. Cada robô pode agir de forma mais independente e uma falha de um dos membros não implica uma falha na missão, mas para que seja possível é necessário o desenvolvimento de mecanismos de coordenação eficientes que permitam a sincronização e colaboração entre os diversos membros da equipa.

Em sistemas de múltiplos robôs, existe uma arquitetura distribuída para mapeamento, onde os robôs móveis cooperam por meio de uma eficiente partilha de informação. Ou seja, cada robô constrói o seu próprio mapa, que é enviado aos outros robôs, partilhando as informações.

A técnica de recuperação topológica pode ser usada como método de exploração para superar mínimos locais. Com esta técnica, além de construir o mapa, o robô constrói simultaneamente um grafo topológico do ambiente que é utilizado para recuperar a missão de exploração em locais previamente visitados, em caso de um mínimo local ocorrer, consequentemente permite a exploração completa do ambiente.

Outra direção de investigação é o desenvolvimento de cooperação de sistemas autónomos cooperantes na ausência de qualquer infraestrutura de comunicação. [Couceiro *et al.*, 2011a; Gonçalo *et al.*, 2014]. A utilização de grupos de robôs a cooperarem na execução de missões

complexas, tais como o mapeamento, a exploração, e o patrulhamento, por exemplo, requer a utilização de comunicação entre robôs por meio de uma rede de comunicação, como por exemplo a comunicação Mobile Ad-hoc NETworks (MANET) que é criada dinamicamente pelas equipas de robôs, em que cada robô atua como sendo um nó na rede. Assim, além da missão, estes robôs devem conhecer a posição relativa de cada membro da equipa de forma a manter a ligação da rede.

O desenvolvimento de equipas robóticas é de extrema importância no domínio de aplicação em que grupos de robôs não dependem das infraestruturas de comunicação preexistentes até porque podem estar danificadas, não estarem acessíveis ou não existirem. Como por exemplo, em missões de procura e salvamento (SaR), em grandes áreas onde ocorreram catástrofes, entre outras situações. A eventual complementaridade de robôs aumenta a capacidade de resolver uma série de problemas. Isso é ainda mais significativo quando os robôs agem em diferentes meios, como por exemplo, a cooperação entre robôs aéreos e terrestres (figura 3).

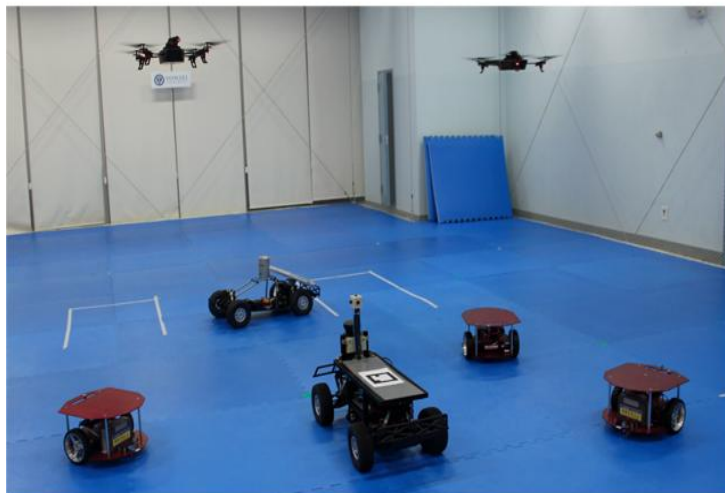


Figura 3 – Cooperação entre robôs heterogêneos que agem em diferentes meios.

Ao nível da tarefa também pode haver heterogeneidade. Ou seja, os robôs podem ser semelhantes, mas as tarefas alocadas a cada um deles ser diferente. Nesse caso, mecanismos de planeamento e coordenação eficientes têm um papel importante, principalmente quando existe um grande número de robôs. Outro problema complexo e aliciante, é ter múltiplos robôs a realizarem múltiplas tarefas em simultâneo - *Multirobot Multitask Problems* (MR-MT) [Gerkey *et al.*, 2004].

Os MRS têm sido objeto de investigação, principalmente no que se refere ao aumento de robustez e eficiência nas tarefas. No entanto, alguns problemas fundamentais precisam de ser abordados, tais como a distribuição de tarefas e a coordenação das ações da missão. Em Sistemas MultiAgentes (MAS) a organização temporária de agentes na realização de uma tarefa específica, não pode ser diretamente transposta para cenários de múltiplos robôs, por não considerarem as restrições do mundo real, como por exemplo, a falha de comunicação.

Aos grupos compostos por um grande número de robôs mais simples, que individualmente não possuem muita capacidade, mas em conjunto podem realizar diversos tipos de tarefas, são genericamente chamados de “enxames” (*swarms*). A utilização de enxames de robôs tem uma forte inspiração biológica uma vez que diversas sociedades de animais requerem, de uma forma ou de outra, a coordenação de múltiplos agentes. Quando se trabalha com enxames, normalmente a heterogeneidade se dá ao nível da tarefa, ou seja, robôs iguais são alocados a diferentes tarefas para a resolução de diferentes problemas. Esse tipo de alocação foi utilizado com sucesso num problema de navegação de enxames [Rocha *et al.*, 2005], onde robôs que completavam a missão de navegação recebiam uma nova tarefa: ajudar outros robôs que ficaram presos em mínimos locais.

Quanto maior for o conhecimento do robô sobre o ambiente, maior será a probabilidade de que a tarefa seja realizada. O mesmo pode ser dito sobre a rede de sensores. À medida que a rede de sensores obtém dados do ambiente, consegue criar um repositório de informação, por meio do qual, pode fornecer informações valiosas para os robôs.

O robô pode escolher a trajetória mais curta para diminuir os gastos de energia ou por outro lado, até poderiam ser escolhidas trajetórias mais longas por conterem regiões ainda não exploradas, que precisariam de ser mapeadas.

3. Métodos de Localização de Sistemas Robóticos

A cinemática desempenha um papel fundamental para definir a posição, a orientação, a velocidade e a aceleração dos sistemas robóticos. A equação da cinemática direta permite-nos prever o movimento do robô sabendo a velocidade angular de cada roda. Por outro lado, a cinemática inversa possibilita o cálculo da velocidade de cada roda necessária para produzir a velocidade e a rotação desejada para o movimento do robô.

Os robôs móveis possuem características de movimentação que dependem dos seus dispositivos de locomoção, que podem restringir os seus movimentos ou a realização de trajetórias mais complexas. Em particular, alguns robôs podem apresentar restrições de movimentos, denominadas restrições não-holonómicas. No caso de robôs com acionamento diferencial, a restrição é imposta pela impossibilidade do robô se movimentar em todas as direções, devido ao sistema não possuir atuadores que permitam tais movimentos.

Por exemplo, se o robô apresenta uma configuração do tipo automóvel, é um exemplo de um robô não holonómico, em que numa tarefa simples como a do estacionamento, a trajetória mais intuitiva sem levar em consideração o sistema de locomoção, seria uma linha reta movendo o robô lateralmente para a posição final desejada. No entanto, esta trajetória não é possível de ser realizada devido à presença de restrições não-holonómicas.

Os métodos de navegação podem ser classificados em três tipos: métodos locais, globais e híbridos. Os métodos locais permitem ao robô a construção de um mapa local a partir das informações sensoriais. Por outro lado, os métodos globais são baseados no conhecimento quer do ambiente de atuação quer da localização do robô. E por último, os métodos híbridos combinam a aplicabilidade dos métodos locais e globais, criando um modelo mais completo, permitindo uma navegação mais robusta e precisa. Este método é aconselhável em ambientes desconhecidos dada a sua flexibilidade, precisão e robustez.

A localização consiste em estimar a posição de um robô no ambiente de trabalho, e determinar a própria posição e orientação, é uma capacidade básica para que qualquer tarefa de navegação seja executada. É fundamental que um robô consiga determinar exatamente a posição em que se encontra, dentro de um cenário, para que o mesmo possa planear uma trajetória até o seu destino e cumprir as tarefas que lhe foram alocadas [Thrun *et al.*, 2002].

A grande maioria dos robôs possui um sistema de odometria que permite estimar o deslocamento e, conseqüentemente, calcular a posição do robô no ambiente. Normalmente, as informações odométricas são obtidas através de sensores acoplados às rodas dos robôs, de forma a calcular o deslocamento a partir do movimento das rodas. Na prática, existem diversos fatores que causam erros nesse tipo de dispositivo como a imprecisão mecânica do sistema e irregularidades do terreno. Outro fator que dificulta e pode até inviabilizar a localização do robô a partir da odometria é o fato de que os pequenos erros causados durante o cálculo da posição do robô vão se acumulando ao longo do tempo, aumentando consideravelmente o erro na localização do robô.

O problema da localização vem sendo amplamente estudado pelos investigadores e, existem algoritmos bastante eficientes na solução do problema. Normalmente, é fornecido um mapa do ambiente e a localização é estimada com base nas informações fornecidas pelos sensores. O grande desafio na solução do problema de localização está no fato de que tanto as informações sobre o ambiente como os dados fornecidos pelos sensores são normalmente limitados e imprecisos. Nesse contexto, técnicas probabilísticas têm sido amplamente utilizadas na solução desse problema. [Elfes *et al.*, 1989; Mataric *et al.*, 1990; Smith *et al.*, 1990].

No caso da localização global, o robô não tem informações prévias da sua posição no mapa, o que torna o processo de localização consideravelmente mais complexo. Nos algoritmos de localização destaca-se o método de *Markov*, tratando-se de uma formulação geral para a solução do problema, existem diversas formas de implementar o algoritmo e de representar os dados do cenário e da posição dos robôs. Onde se destacam o filtro de *Kalman*, o método de *grid* e o método de *Monte Carlo*, apresentados com detalhe nos anexos 1 e 2.

No método de localização baseado no filtro de *Kalman*, aplica-se uma técnica de filtragem. Uma vez conhecida a localização do robô, é possível estimar a sua próxima localização, a partir do seu deslocamento, mesmo considerando os erros associados às informações obtidas pelos sensores. A formulação tradicional do filtro de *Kalman* utiliza determinados pontos no espaço, que servem como referência, para que a posição do robô seja estimada [Leonard *et al.*, 1991], utilizando funções Gaussianas para representar cada uma das coordenadas da posição do robô no cenário. Este algoritmo é amplamente utilizado em sistemas nos quais as informações são imprecisas e nem sempre estão disponíveis. Este método prova a sua convergência quando determinados requisitos estão presentes.

O algoritmo de localização baseado no método de *grid*, consiste em dividir o ambiente em pequenas unidades de espaço chamadas células, fazendo uma grelha [Elfes *et al.*, 1989; Fox *et al.*, 1999a]. Para cada uma dessas células é calculada a probabilidade de o robô estar na célula. Este método permite que múltiplas hipóteses sobre a posição do robô sejam mantidas. Este método é adequado para a solução dos problemas de localização local e global. O ambiente também é representado através de células, no qual cada uma delas é marcada como ocupada ou livre. As células livres representam espaços onde não há presença de obstáculos.

O algoritmo de localização baseado em técnicas de Monte Carlo [Fox *et al.*, 1999b] consiste em manter várias hipóteses de localização no ambiente, mas cada hipótese é representada por uma unidade chamada partícula. Ao ser iniciado, o algoritmo distribui partículas por todo o mapa do ambiente. Comparando os dados obtidos pelos sensores com as informações disponíveis no mapa, associa um peso para cada partícula. Este peso representa a possibilidade dessa partícula representar a real localização do robô. Durante a execução do algoritmo, o peso de cada partícula é calculado. Partículas com peso mais baixo são eliminadas e as partículas com peso mais elevado são replicadas. Com o tempo todas as partículas tendem a se concentrar no local do mapa que representa a localização correta do robô. Este método também utiliza grelhas de ocupação para representar o cenário.

Através da partilha de informação das leituras dos sensores de vários robôs é possível acelerar o processo de exploração do ambiente, assim os robôs, em equipa, podem criar um mapa comum de forma mais rápida. O mapa global pode ser construído de forma centralizada ou cada robô pode construir seu próprio mapa com as medidas de todos os outros robôs de maneira distribuída e mais robusta.

Existem várias técnicas de múltiplos robôs com Localização e Mapeamento Simultâneos (SLAM) para executar esta tarefa. No entanto, outra opção é construir mapas locais, apenas com as informações de cada robô individual, juntando a informação proveniente dos outros robôs. Em qualquer caso, a utilização de múltiplos robôs a cooperar entre si, permite a criação de um mapa comum do ambiente, beneficiando das observações de todos os robôs.

Algumas das técnicas já existentes para um único robô foram estendidas para múltiplos robôs. Por exemplo, Yamauchi [Yamauchi *et al.* 1998] ampliou a exploração do ambiente, baseada em fronteiras para múltiplos robôs. Nesta abordagem, cada robô transmite o seu mapa de grelha local sempre que atinge uma fronteira. Consequentemente, cada robô integra no seu próprio mapa global, os dados recolhidos por todos os membros da equipa, usando a localização odométrica. No entanto, apesar dos robôs cooperarem na construção do mapa, eles não se

movem de forma coordenada e consequentemente os membros da equipa não estão otimizados.

Nos trabalhos realizados por Schultz *et al.* [Schultz *et al.* 1999] melhoraram a abordagem anterior por meio da utilização de um SLAM completo, em vez da localização odométrica. Um dos problemas destas técnicas é o seu carácter centralizado, nesse sentido, uma abordagem mais distribuída é desejável para se obter uma melhor robustez na aplicação da técnica.

Outros autores propuseram usar um modelo de mercado para atribuir um destino aos robôs. Nesta abordagem, cada robô tem uma fila de destinos atribuídos [Zlot *et al.*, 2002] ou os robôs negociam esses destinos com os restantes robôs [Sheng *et al.*, 2006].

Os robôs também podem ser coordenados por meio da divisão do ambiente em tantas regiões quanto os robôs existentes [Solanas e Garcia, 2004], ou através de algoritmos de agrupamento *Kmeans* [Hartigan e Wong, 1979] ou segmentando o ambiente usando gráficos de Voronoi tal que a distância até o obstáculo mais próximo no mapa é considerado um mínimo local [Wurm *et al.*, 2008; Choset e Burdick, 2000].

Um modelo reativo pode ser implementado com três comportamentos básicos: evitar os obstáculos, deslocar-se para a fronteira e evitar outros robôs [Lau, 2003]. Este tipo de modelo foi melhorado adicionando dois novos comportamentos: melhorar pontos de referência imprecisos e ir para zonas inexploradas, introduzindo o conceito de utilidade ou ganho de informação [Juliá *et al.*, 2008]. Os modelos reativos funcionam bem em ambientes simples, desde que não existam mínimos locais. Um dos requisitos para a exploração de um ambiente através da utilização de múltiplos robôs é a limitação da distância que um robô pode distanciar-se dos outros mantendo um canal de comunicação ativo entre eles [Arkin e Diaz, 2002].

Com a aplicação do SLAM, onde cada robô constrói o seu próprio mapa usando um filtro de Monte Carlo, é possível os robôs partilharem os seus mapas locais e continuarem a explorar o ambiente [Fox *et al.*, 2006]. A posição e a orientação dos robôs e os mapas criados são considerados pelo algoritmo de navegação como se fossem conhecidos sem incertezas associadas. E uma vez que, a existência de incertezas aumenta à medida que os robôs exploram o ambiente, o algoritmo SLAM tem maior probabilidade de falhar e os mapas criados pelos robôs podem tornar-se inúteis para a navegação. Para um planeamento integrado sem SLAM, o robô identifica possíveis alvos como fronteiras ou regiões mal mapeadas, faz uma rota de viagem através desses alvos, executa essa rota e, retorna à posição e orientação inicial [Sim e Little, 2006].

Alguns autores integraram o algoritmo SLAM de forma direta usando um SLAM topológico. As informações topológicas são guardadas durante a exploração em que o conjunto de nós é representado por lugares distintos unidos por arestas através de um conjunto de estratégias de controle local que permite mover o robô de um local para outro.

O algoritmo *Next Best View* [Grabowski *et al.*, 2003] foi projetado para a utilização de sensores sonar, considerando que a incerteza nas leituras do sonar é bastante elevada, torna-se importante obter leituras suficientes de um obstáculo através de diferentes ângulos.

Existem modelos preditivos de movimento e observação a partir de um filtro de Kalman estendido (EKF) baseado em características, para prever a incerteza das diferentes ações de controle do robô [Feder *et al.*, 1999]. A matriz de covariância prevista para cada ação, permite que o robô possa escolher a ação que minimiza todas as elipses de erro. No entanto, esta abordagem não pode ser considerada em ambientes sem pontos de referência.

Outros autores aplicam uma curva paramétrica para controlar os movimentos dos robôs [Sim *et al.*, 2004]. Esses parâmetros da curva são selecionados de forma a maximizar a função objetivo, considerando o espaço não explorado bem como a incerteza medida. Nesse caso, a estratégia de retornar a zonas previamente exploradas, como a origem, é usada para reduzir a incerteza.

Da mesma forma, pode ser calculada a incerteza dos robôs e pontos de referência usando uma medida como a técnica apresentada por Feder *et al.* [Feder *et al.*, 1999; Bourgoult *et al.*, 2002; Makarenko *et al.*, 2002].

Um filtro de partículas *Rao-Blackwellized* baseado em grelha para SLAM pode ser implementado e permite obter um ganho de informação das áreas inexploradas, bem como informações para reduzir a incerteza do mapa e da posição e orientação dos robôs [Stachniss *et al.*, 2005].

Um outro exemplo para melhorar a localização e mapeamento na utilização de dois robôs, em que um deles é estático e serve de referência para a localização do robô em movimento. Desta forma, a localização e o mapeamento do sistema geral é melhorado significativamente [Rekleitis *et al.*, 2001]. Outra abordagem considera apenas pontos aleatórios próximos das fronteiras e a ordem de navegação entre essas é avaliada através de uma árvore de decisão [Tovar *et al.*, 2006].

4. Estratégias de Controlo e Coordenação

O controlo de um sistema robótico, juntamente com a coordenação, são os pontos principais da robótica cooperativa. Embora haja muitos resultados científicos e tecnológicos já desenvolvidos para sistemas robóticos *indoor*, grandes desafios ainda existem, quando se trata na utilização *outdoor* (i.e. sujeitos a inclinações, efeitos dinâmicos da interação veículo-terreno, etc.) grandes desafios ainda existem a serem ultrapassados.

Com intuito de apresentar diferentes estratégias de controlo é necessário fazer uma breve introdução aos problemas típicos de controlo de robôs móveis, estes problemas podem ser divididos em:

- Problema da estabilização, que consiste em atingir uma configuração final desejada para o robô, que é definida por uma localização no espaço cartesiano e uma determinada orientação, a partir de uma localização e orientação da configuração inicial do robô.
- Problema de seguimento de trajectória, que consiste em fazer com que o robô atinja e siga uma determinada trajetória no espaço cartesiano, iniciando numa determinada configuração inicial, dentro ou fora da trajetória.
- Problema do seguimento do caminho, que pode ser considerado como um problema de seguimento de trajetória, em que o robô deve apenas seguir o caminho geométrico mas sem restrições temporais.

Na tarefa do seguimento da trajetória, o robô deve seguir um determinado caminho cartesiano com uma lei temporal específica associada, desta forma, o problema torna-se mais simples, porque passa a ser um problema de estabilização do erro entre a configuração cartesiana do robô e da referência.

Na execução da navegação do robô devem existir mecanismos para evitar colisões, assim como para o tratamento de situações imprevistas, envolvendo uma utilização de uma estratégia de controlo do tipo reativo, deliberativo ou hierárquico/híbrido. Inicialmente o

sistema de controlo deve estabelecer uma rota, ou seja, deve planear uma trajetória, baseando-se no mapa do cenário. Os métodos mais utilizados para a determinação do melhor caminho são, por exemplo, o algoritmo A* (AStar) ou também denominado algoritmo Dijkstra [Latombe *et al.*, 1991] e o algoritmo de campos potenciais [Dudek *et al.*, 2000], entre outros.

A fusão sensorial e a comunicação entre os diversos sistemas robóticos bem como a utilização do mapa global partilhado por todos os sistemas robóticos permite uma melhor estimativa quer da posição do sistema robótico quer da partilha dos recursos existentes.

A arquitetura de controlo de um sistema robótico cooperativo deve apresentar diferentes níveis de controlo, conforme se apresenta na figura 4. O controlo de baixo nível é dedicado aos atuadores do robô e o nível mais elevado de controlo é destinado à tomada de decisão, sendo que, neste nível de decisão podem existir várias camadas.

A estrutura de *software* adotada para implementar um sistema de controlo de múltiplos robôs deve ter também como objetivo, reduzir ao mínimo, o tempo de desenvolvimento da aplicação, por essa razão a adoção do sistema operativo ROS, *Robot Operating System*, além de ser uma *framework open-source*, facilita o desenvolvimento de software na robótica.

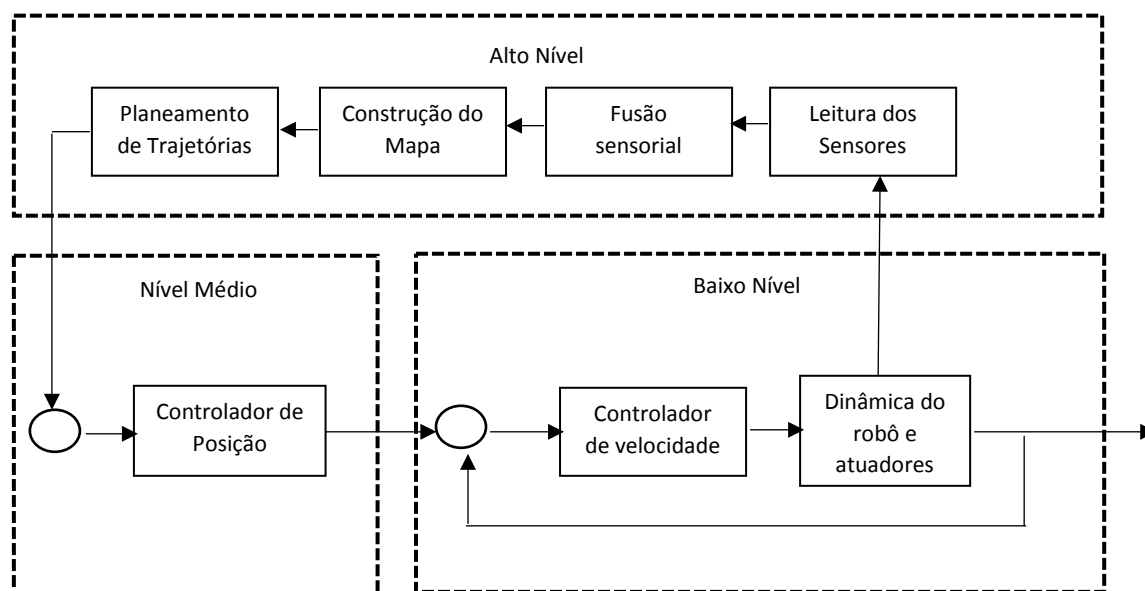


Figura 4 – Arquitetura de controlo de um sistema robótico.

Qualquer sistema robótico comporta um conjunto de módulos, que poderá ser complexo de gerir, quando se trabalha com grupos de robôs. O sistema ROS oferece um conjunto de ferramentas de forma a simplificar a construção de um robô complexo e robusto permitindo subdividir tarefas pelos elementos da equipa, e no fim, fazer a integração, de forma simples,

numa arquitetura de desenvolvimento cooperativo que permite a comunicação entre todos os robôs.

O ROS estabelece uma camada de comunicação em execução no topo de um sistema operacional comum, que permite a troca de mensagens entre vários sistemas operacionais ROS de cada sistema robótico. Os nós ROS são programas que utilizam os recursos fornecidos do sistema para comunicar, executar tarefas e operam de forma independente e concorrente, e nem precisam estar em execução no mesmo computador, além disso, é possível usar diretamente o código desenvolvido de outros robôs para, por exemplo, implementar, testar e comparar um determinado algoritmo.

O sistema ROS permite abstrair das complexidades de *hardware* e promove em simultâneo a reutilização de código que praticamente se tornou um padrão na robótica. Por exemplo, os pacotes de SLAM no ROS geralmente lidam com sensores do robô e o nó SLAM que cria o mapa, geralmente em forma de grelha de ocupação, e o publica num tópico dedicado, transmitindo-o assim a qualquer robô. Além desta função, o nó SLAM também é responsável por determinar a posição do robô no mapa criado, constituindo assim uma solução para o problema completo do SLAM [Frese *et al.*, 2006].

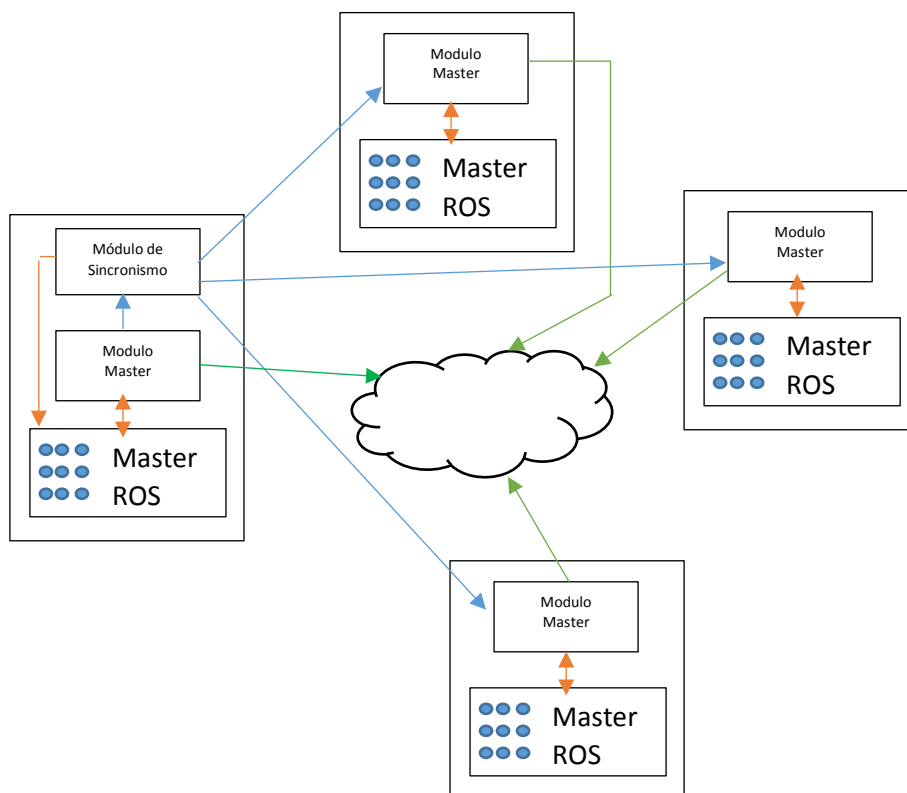


Figura 5 – Arquitetura de partilha de informação entre robôs.

Na cooperação entre sistemas robóticos é necessário a utilização de uma rede sem fios sendo portanto necessário a minimização da quantidade de dados a passar pela rede, estes dados são essenciais para auxiliar a navegação do robô no ambiente, figura 5. No entanto, em cenários mais complexos a ligação pode tornar-se num problema, e em aplicações do mundo real, o esforço de navegação pode ser apenas uma pequena parte das tarefas, portanto, um modelo eficiente de comunicação também é um elemento chave para uma implementação escalável, pois à medida que o número de sistemas robóticos aumenta, a quantidade de dados também aumenta.

Existem diversos métodos de compressão quer na transmissão quer no armazenamento de dados [Ziv and Lempel, 1977]. Os métodos de compressão são divididos em dois grupos principais: os métodos sem perdas, que permitem reconstruir os dados originais sem erros; e os métodos com perdas, que fazem uso do modo como o ser humano percebem sinais para descartar dados irrelevantes [Salomon, 2007].

5. Caso de Estudo

A utilização de técnicas de cooperação de múltiplos robôs, como o controlo descentralizado, baseado em redes móveis *ad-hoc*, e da fusão sensorial, permite que os sistemas robóticos consigam desenvolver aplicações em diferentes domínios, tais como vigilância, inspeção de infraestruturas ou ações de cooperação entre múltiplos robôs eventualmente em cenários de catástrofe entre outros [Lobo *et al.*, 2005].

5.1 Inteligência Coletiva do Tipo Enxame

Sistemas robóticos cooperativos que incorporem mecanismos de inteligência coletiva do tipo enxames [Marcolino *et al.*, 2008; Couceiro *et al.*, 2011b; Bakhshipoura *et al.*, 2017], normalmente, ocorrem em cenários dinâmicos de elevadas dimensões e com condições adversas, que colocam em causa a aplicabilidade desses mesmos sistemas. As contribuições desenvolvidas sustentam-se em torno da extensão do método *Particle Swarm Optimization* (PSO) aplicado a sistemas robóticos cooperativos, denominado de *Robotic Darwinian Particle Swarm Optimization* (RDPSO).

No PSO a população funciona de forma coordenada, em que cada partícula (isto é, cada robô) tende a deslocar-se no decorrer do desenvolvimento da tarefa. Aqui, nenhuma partícula assume o controlo, o mérito de cada partícula é medido em cada iteração de acordo com a sua posição atual. Cada partícula memoriza a “melhor posição” onde já esteve. E a melhor posição encontrada é conhecida por todo o enxame.

No DPSO o algoritmo é uma extensão do PSO onde se usa a seleção natural, ou a sobrevivência do mais apto, com a vantagem de aumentar a capacidade de evitar soluções locais [Couceiro *et al.*, 2012a; Couceiro *et al.*, 2012b]. O DPSO apresenta um mecanismo de “punir-recompensar” baseado em comportamentos biológicos e sociológicos que aumentam a capacidade de evitar mínimos locais. O DPSO utiliza grupos de agentes virtuais interativos para alcançar a otimização do método. Ao contrário dos agentes virtuais, os robôs operam no mundo real, onde os obstáculos precisam ser levados em consideração.

A extensão da otimização do DPSO, para o domínio dos MRS é designada por RDPSO onde entram em consideração as restrições do mundo real, tal como a prevenção de obstáculos e as redes de comunicação. O RDPSO consiste numa arquitetura robótica de enxame distribuída que beneficia do particionamento dinâmico da população de robôs utilizando mecanismos

Apesar de várias abordagens possíveis para implementar um sistema de controlo, a lógica Fuzzy parece ser uma das mais adequadas como uma ferramenta de análise de múltiplos critérios, L.A. Zadeh [Zadeh, 1965] apresentou a lógica Fuzzy para lidar e representar incertezas. A indefinição e a imprecisão associadas a dados qualitativos podem ser representadas de maneira lógica usando variáveis linguísticas e funções de associação sobrepostas no intervalo incerto. Tradicionalmente, um sistema Fuzzy é um mapeamento não linear estático entre as suas entradas e saídas, ou seja, não é um sistema dinâmico.

O termo Fuzzy refere-se à capacidade do método de lidar com informações imprecisas ou vagas. Embora as informações de entrada para o sistema possam ser imprecisas, os resultados da análise difusa não o são. A lógica difusa permite descrever com precisão um sistema de controlo em termos linguísticos para definir a relação entre a informação de entrada e a ação de saída. A lógica Fuzzy permite quantificar o significado de variáveis, valores e regras linguísticas que são especificadas de acordo com a aplicação. Conjuntos difusos necessitam de funções de associação, funções lineares, por exemplo triangulares ou trapezoidais, devido à sua simplicidade e eficiência se considerarmos os problemas computacionais para processar a entrada para obter o raciocínio da saída. As etapas envolvidas de um sistema Fuzzy baseado em regras são:

1. Identificar as entradas e a sua gama de variação;
2. Identificar as saídas e a sua gama de variação;
3. Criar o grau de função de associação Fuzzy para cada entrada e saída;
4. Construir a base de regras na qual o sistema vai operar;
5. Decidir como a ação será executada atribuindo pontos fortes às regras;
6. Combinar as regras e *defuzzificar* a saída.

Os conjuntos difusos quantificam as informações com base nas regras, e o mecanismo de inferência vai operar os conjuntos difusos. Deve-se especificar como o sistema converte as entradas numéricas em conjuntos Fuzzy, para que estes possam ser utilizados nas saídas, este processo é chamado de *Fuzzificação*. O processo envolve a utilização de regras bem definidas e para especificar as regras, é necessário efetuar uma descrição linguística. As expressões linguísticas são necessárias para obter as características das entradas e saídas. Como exemplo, as temperaturas nem sempre são dadas em °C ou °F, mas em termos linguísticos como frio ou quente. Como os valores linguísticos não são representações precisas, as regras linguísticas

também podem não ser precisas. São simplesmente ideias abstratas sobre como conseguir um bom controlo que pode significar coisas diferentes para pessoas diferentes. No entanto, estão num nível de abstração com o qual é possível especificar para controlar um determinado processo. As regras IF-THEN-ELSE geralmente são expressas no formulário: IF variável IS propriedade THEN action_1 ELSE action_2.

A lógica da decisão determina como as operações são executadas e determinam as saídas de cada uma das regras IF-THEN-ELSE. Muitos métodos de implicação Fuzzy podem ser escolhidos, por exemplo, o método de inferência mínima de *Mamdani* é usado para representar a ligação AND. Este método permite realizar a *fuzzificação*, onde o valor mínimo é selecionado entre as variáveis disponíveis. As variáveis de saída linguística necessitam de serem traduzidas num valor, a este processo designa-se por *defuzzificação*, existem também diversos métodos de *defuzzificação* que tendem a considerar todos os elementos do conjunto difuso resultante com os pesos correspondentes. Outros métodos levam em conta apenas os elementos correspondentes aos pontos máximos das funções de associação resultantes.

```

IF [t] IS Active THEN  $\alpha$  IS Small

ELSE  $\alpha$  IS Large

IF [t] IS Close OR [t] IS Far OR [t] IS NOT Connected THEN  $\alpha$  IS Nominal

IF [t] IS Social OR [t] IS Close OR [t] IS Far OR [t] IS NOT Connected THEN  $\rho_2$  IS Small

ELSE  $\rho_2$  IS Large

IF [t] IS Close THEN  $\rho_3$  IS Large

ELSE-IF [t] IS Far OR [t] IS NOT Connected THEN  $\rho_3$  IS Small

IF [t] IS Far OR [t] IS NOT Connected THEN  $\rho_4$  IS Large

ELSE-IF [t] IS Close THEN  $\rho_4$  IS Small

```

Figura 7 – Exemplo de um conjunto de regras para controlar o comportamento dos robôs com base em informações contextuais.

As regras apresentadas na figura 7 mostram que se atribuem diferentes prioridades aos parâmetros do RDPSO, por exemplo no ρ_3 (isto é, evitar obstáculos) e no ρ_4 (isto é, manter a ligação MANET). Embora colisões menores entre robôs sejam aceitáveis, a perda de vários robôs pode comprometer o objetivo da missão. Por outro lado, o parâmetro de evitar os obstáculos ρ_3 afeta apenas o comportamento de um robô específico quando deteta um

obstáculo. O sistema Fuzzy aqui proposto ajusta sistematicamente o comportamento do RDPSO de tal forma que se pode facilmente compreender as informações contextuais relativas ao robô e ao subgrupo simplesmente observando a evolução dos parâmetros. Assim, o uso do conhecimento contextual melhora a percepção dos robôs, permitindo a detecção rápida de mudanças ambientais ou de missão, explorando as informações sobre a dinâmica dos recursos do mundo real.

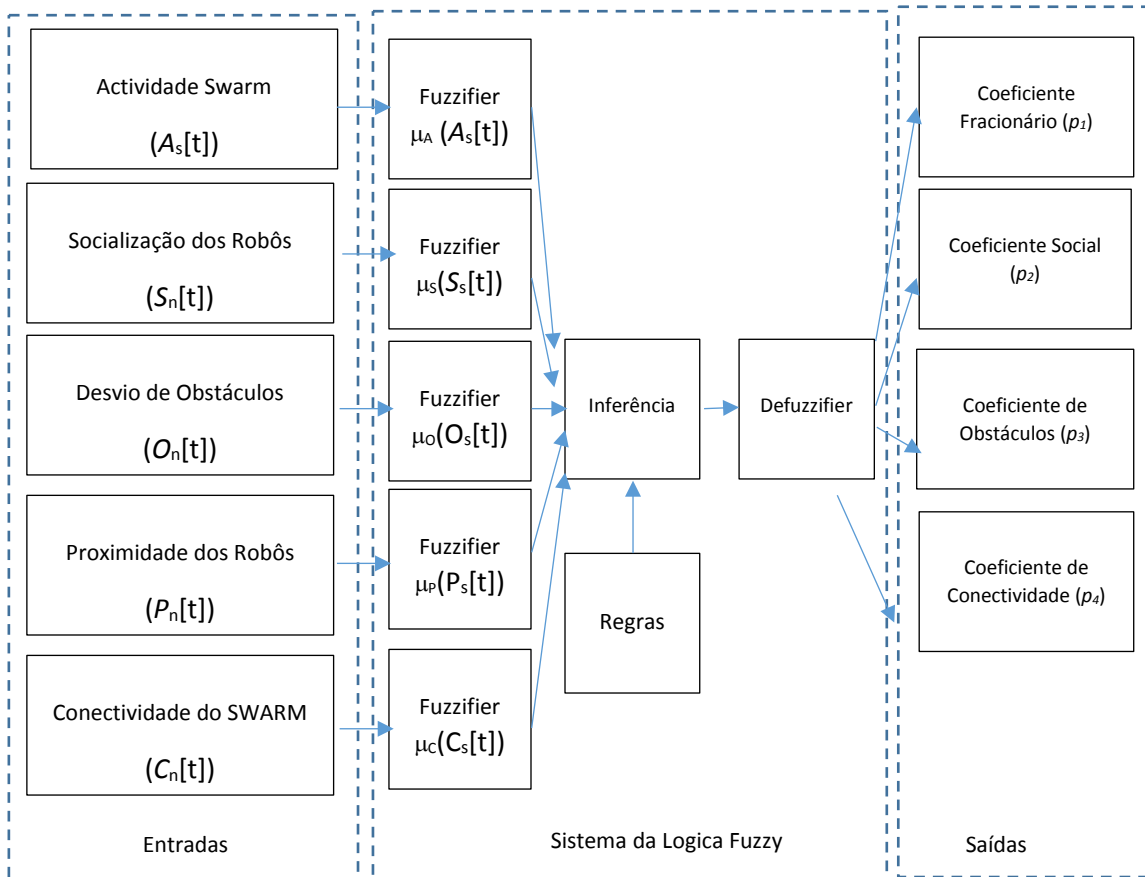


Figura 8 – Arquitetura de controle para adaptação dinâmica dos parâmetros do algoritmo RDPSO.

O RDPSO foi projetado para diferentes tipos de sistemas robóticos através da utilização de duas realimentações distintas, não considerando assim, as características cinemáticas e dinâmicas do robô. Estes aspetos são considerados pelo controlador de baixo nível que é projetado para um sistema robótico específico (figura 9). Em outras palavras, se o *hardware* for alterado, apenas o bloco de baixo nível precisará de ser substituído. O diagrama de estados representado na figura 10 apresenta o Modelo de Estados Finitos do algoritmo RDPSO evolucionário bem como as probabilidades de transição associados aos subgrupos socialmente excluídos [Couceiro *et al.*, 2012d].

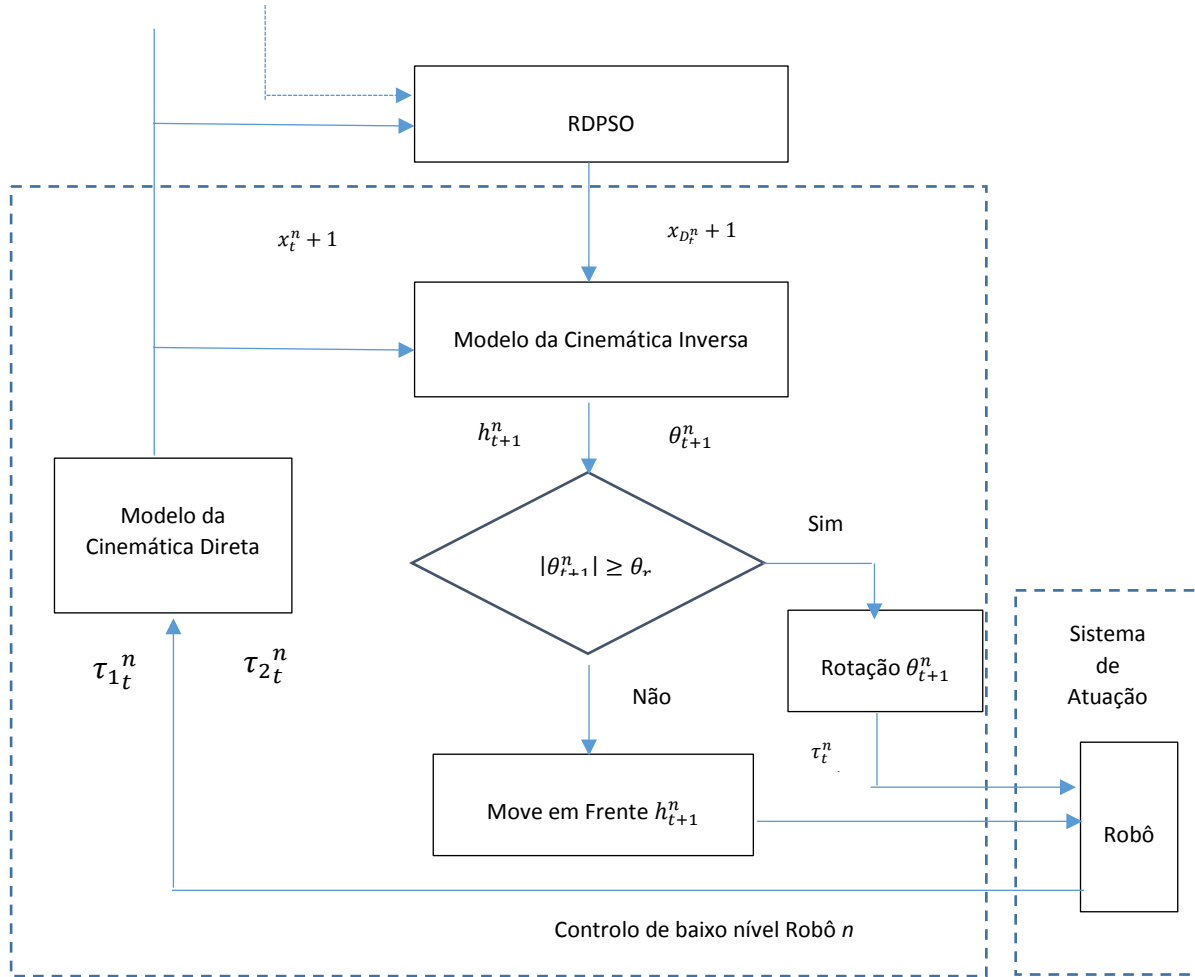


Figura 9 – Arquitetura do diagrama de controle de baixo nível.

A experiência com um enxame de 12 robôs diferenciais [Couceiro *et al.*, 2012e], figura 11, foi realizada em um cenário de 2,55 m $\times$ 2,45 m ($A \approx 6,25$ m²). O ambiente experimental (figura 12) era constituído por um cenário com dois locais distintos, representados por pontos iluminados com diferentes intensidades de luz. Apesar de ser um cenário sem obstáculos, os próprios robôs atuam como obstáculos dinâmicos. Foi considerado um número máximo de 12 robôs, o que corresponde a uma densidade populacional de aproximadamente 2 robôs por m². Cada robô possui sensores de luz (LDR) que permitem encontrar locais candidatos e medir a sua intensidade de luz. O local com maior intensidade luminosa é a solução ótima. O objetivo dos robôs do enxame é escolher coletivamente o local com maior intensidade luminosa. Os valores de intensidade foram obtidos varrendo todo o cenário com um único robô no qual o sensor de luz utilizado se encontra no centro de cada robô.

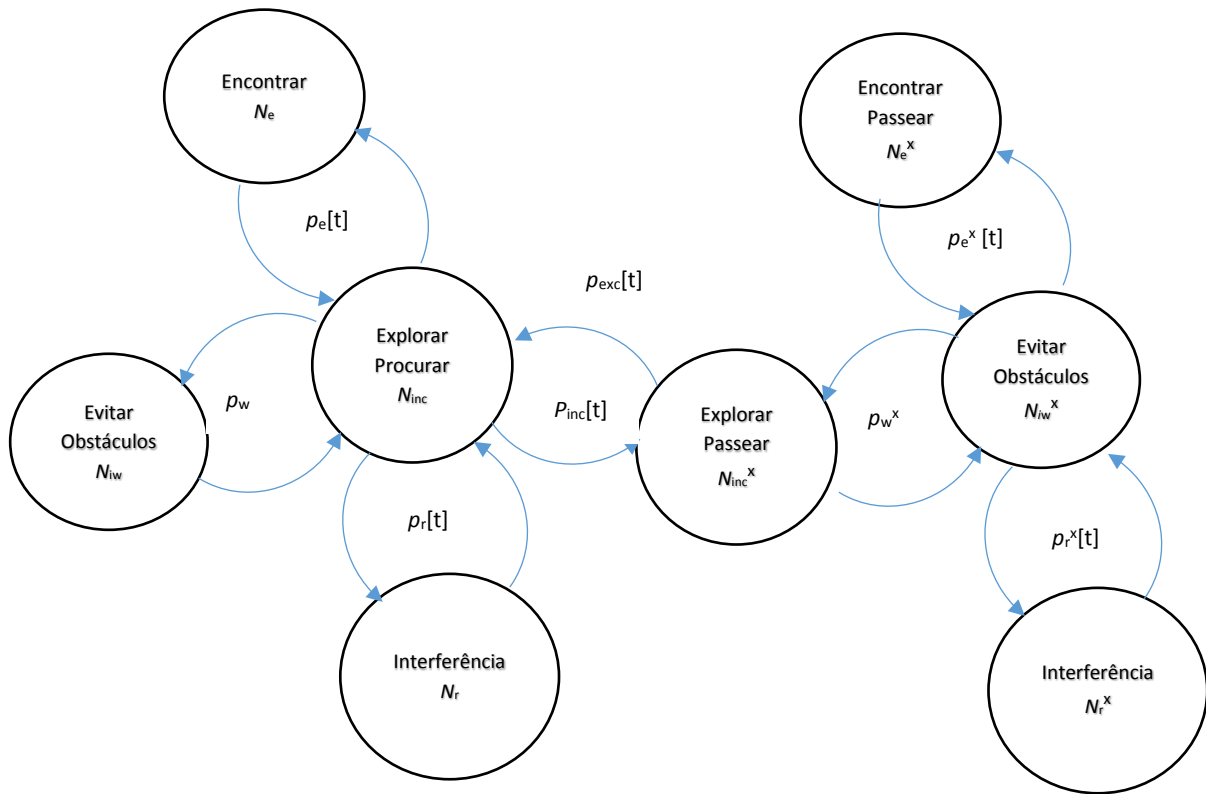


Figura 10 – Modelo de Estados Finitos Semi-Markov do algoritmo RDPSO evolucionário e probabilidades de transição associados aos subgrupos socialmente excluídos.



Figura 11 – Exemplar de um robô diferencial usado no enxame de 12 robôs.



Figura 12 – Cenário de testes experimentais utilizado. Dimensões: 2,55 m x 2,45m, com dois locais distintos com luminosidades diferentes.

A comunicação entre robôs para partilhar posições e soluções locais foi realizada usando o protocolo sem fio *ZigBee* 802.15.4. permitindo um alcance máximo de comunicação maior do

que o cenário utilizado. Os robôs receberam uma lista de endereços dos seus colegas de equipa para simular a comunicação MANET com alcance limitado.

A sequência das figuras mostra o desempenho do RDPSO com uma população de 12 robôs. Na figura 13 a) a população é inicialmente dividida em dois enxames distintos - verde e vermelho - desdobrados em espiral; na 13 b) os enxames procuram de forma independente o local mais luminoso, tomando em consideração uma distância máxima de comunicação de 1,5 metros entre robôs do mesmo enxame; na 13 c) um robô do enxame vermelho e outro do verde encontram a solução sub-ótima e ótima, respetivamente; na 13 d) como o enxame vermelho não melhora o seu desempenho global, alguns robôs são excluídos, sendo adicionados ao subgrupo socialmente excluído (enxame branco); na 13 e) desde que o enxame verde melhorou o seu desempenho global, pode chamar novos membros do subgrupo socialmente excluído; na 13 f) o enxame verde aumenta, em número, com os robôs excluídos anteriormente que não conseguiram melhorar sua solução. De notar que, nem todos os robôs convergem para a solução ideal, uma vez que têm que manter uma distância máxima entre eles de 1,5 metros.

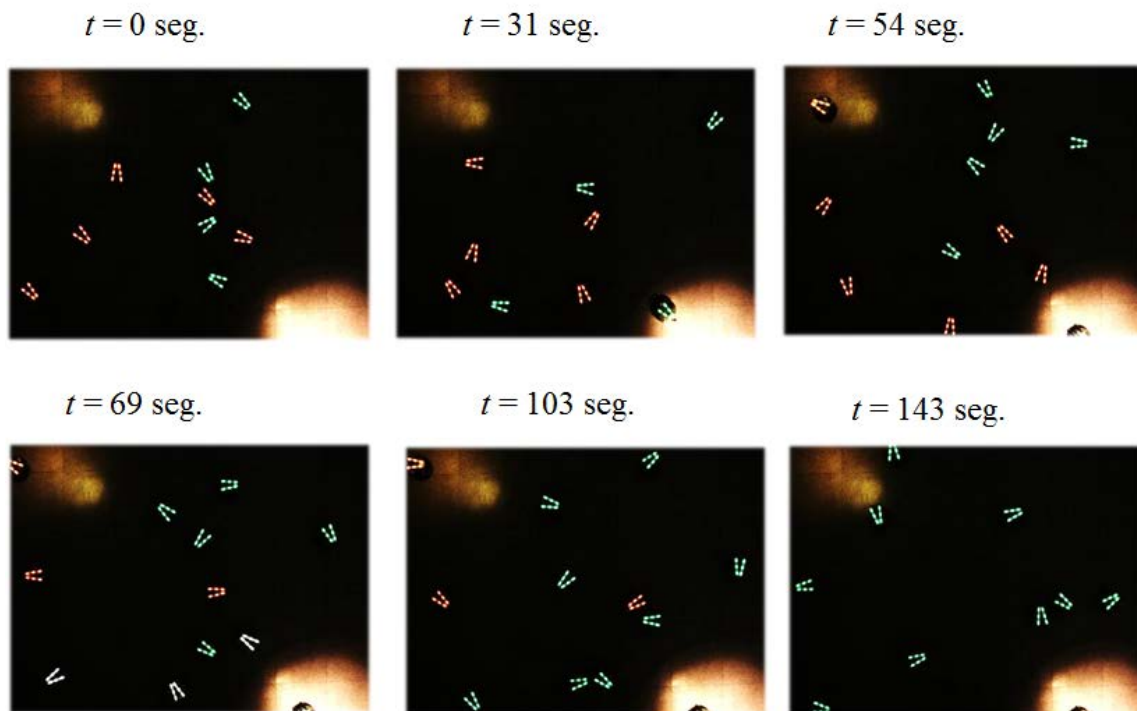


Figura 13 – Evolução temporal de uma experiência SaR:
a) t=0 seg; b) t=31 seg; c) t=54 seg; d) t=69 seg; e) t=103 seg e f) t=143 seg.

Os fundamentos em torno do *RDPSO* são introduzidos, focando-se na dinâmica dos robôs, nos constrangimentos introduzidos pelos obstáculos e pela comunicação, e nas suas propriedades evolucionárias. Considerando a colocação inicial dos robôs no ambiente como algo

fundamental para aplicar sistemas de enxames em aplicações reais, foi introduzida uma estratégia de colocação de robôs mais realista com obstáculos fixos, figura 14, [Couceiro *et al.*, 2014b].

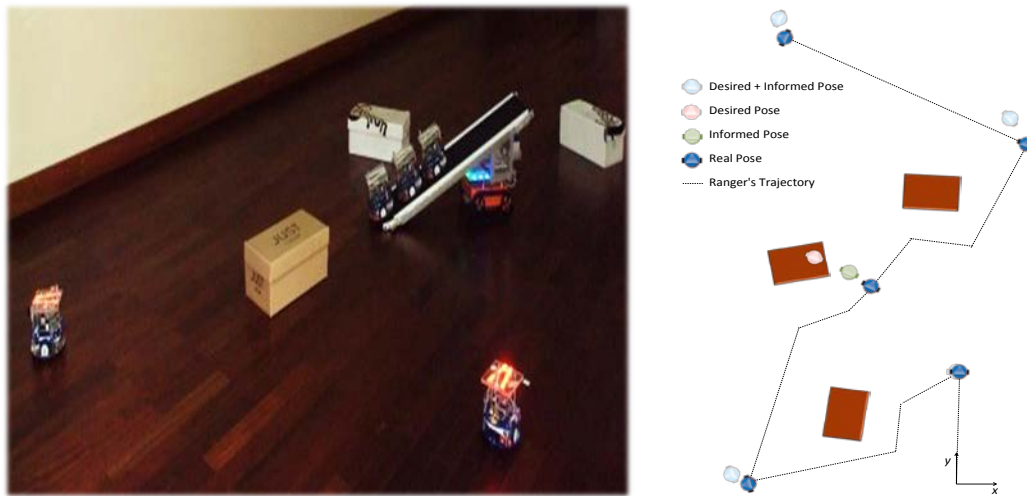


Figura 14 – Exemplo de um sistema real com obstáculos fixos.

Para tal, a população de robôs é dividida de forma hierárquica, em que são utilizadas plataformas mais robustas para colocar as plataformas de enxame no cenário de forma autónoma, conforme se pode observar na figura 15 e com mais detalhe no anexo 3 e 4. Após a colocação dos robôs no cenário, é apresentada uma estratégia para permitir a criação e manutenção de uma rede de comunicação móvel *ad-hoc* com tolerância a falhas. Esta estratégia não considera somente a distância entre robôs, mas também a qualidade do nível de sinal radiofrequência, redefinindo assim a sua aplicabilidade em cenários reais [Couceiro *et al.*, 2014a, Couceiro *et al.*, 2014b, Couceiro *et al.*, 2014c, Couceiro *et al.*, 2014d].



Figura 15 – Duas equipas de robôs inseridos num cenário com obstáculos fixos.

É feita uma análise detalhada do sistema de comunicação inerente ao algoritmo, para atingir uma implementação mais escalável do *RDPSO* a cenários de elevada complexidade. Esta

elevada complexidade inerente à dinâmica dos cenários, motivaram a ultimar o desenvolvimento do *RDPSO*, integrando para o efeito um mecanismo adaptativo baseado em informação contextual (por exemplo, o nível de atividade do grupo). A figura 16 ilustra um ambiente de simulação que após ser testado, pode servir para simular situações com outras dimensões.

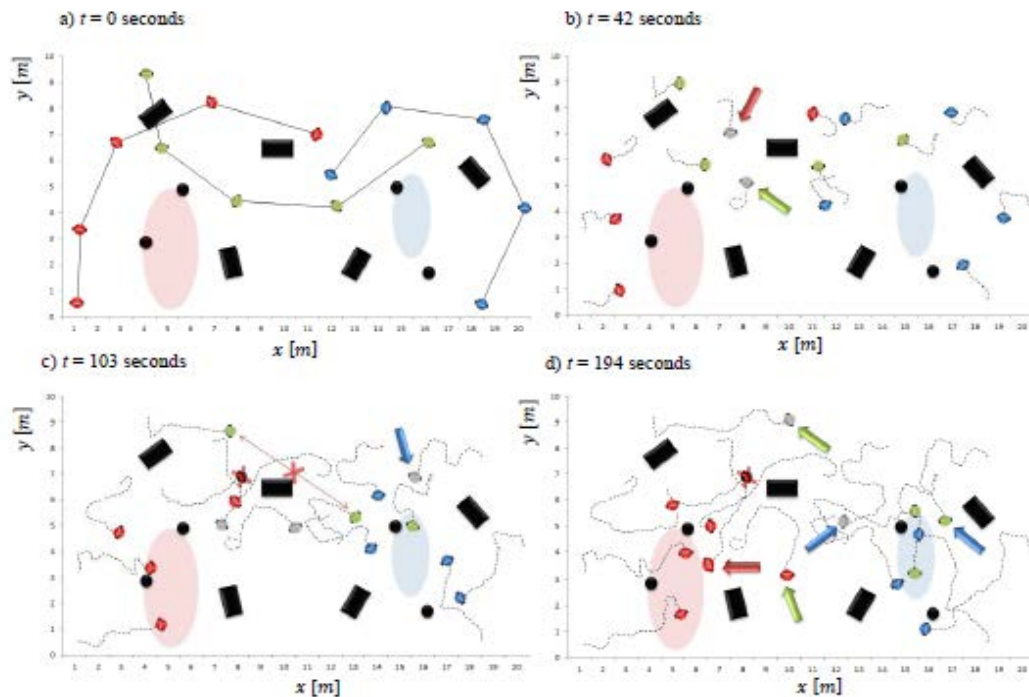


Figura 16 – Simulação de equipas de robôs em cenários com obstáculos.

Face a estas considerações, foi possível contribuir para expandir o estado-da-arte em robótica de enxame com algoritmos inovadores aplicados em contexto real. Neste sentido, todos os métodos propostos foram extensivamente validados e comparados com alternativas, tanto num ambiente de simulação como com robôs reais. Para além disso, e dadas as limitações do número de robôs, das dimensões dos cenários, e nos constrangimentos reais foi possível simular situações reais. Em suma, este tipo de investigação teve e tem aplicabilidade prática ao colmatar a lacuna que se faz sentir no âmbito das estratégias de enxames de robôs em contexto real e, em particular, em cenários de SaR.

As plataformas e ferramentas desenvolvidas e apresentadas neste caso de estudo deram origem a diversas contribuições científicas encontradas nas referências bibliográficas [Couceiro *et al.*, 2011a – 2011b; Ghamisi *et al.*, 2012; Couceiro *et al.*, 2012a - 2012e; Couceiro *et al.*, 2013a – 2013c; Couceiro *et al.*, 2014a – 2014d]. Nomeadamente:

- Desenvolveram-se plataformas robóticas ideais, a que designámos *eSwarBot* (figura 11), para serem usadas em aplicações de enxames (figura 12 e 15). Estas plataformas foram desenvolvidas tomando em consideração a sua dimensão e baixo custo, as limitações de odometria, as capacidades de detecção, as limitações da bateria, bem como a comunicação sem fios.
- Em termos de ferramentas de simulação foi desenvolvido um simulador de múltiplos robôs para o MatLab (figura 16), e uma extensão da compatibilidade do simulador Webots aplicado às plataformas robóticas e-pucks.
- A arquitetura robótica RDPSO desenvolvida, que beneficia do particionamento dinâmico de todo o enxame de robôs por meio de um mecanismo evolucionário de exclusão social baseado na sobrevivência do mais apto de Darwin. Na análise formal do RDPSO, e para entender melhor a relação entre os parâmetros do algoritmo e a sua convergência, é apresentada um conjunto de métricas de avaliação baseadas em contexto. Em ambos os níveis microscópicos e macroscópicos, são propostas métricas para avaliar o comportamento RDPSO, estudando os vários conceitos inerentes às técnicas de enxame com uma análise de espaço de fase do movimento do robô.
- A utilização da lógica difusa como um mecanismo de tomada de decisão para adaptação sistemática dos parâmetros RDPSO e para múltiplos robôs. Devido à baixa complexidade computacional das estratégias de lógica difusa e aos resultados positivos obtidos, a mesma arquitetura pode ser reproduzida para se adequar a qualquer outro método, em robótica ou não, que exija um mecanismo adaptativo sistemático.
- Outras contribuições científicas consistiram em delinear um conjunto de regras difusas nas quais as saídas correspondentes aos parâmetros RDPSO se adaptariam sistematicamente, com base nas entradas definidas por várias métricas de avaliação baseadas num determinado contexto. Por exemplo, quando um determinado robô não deteta qualquer obstáculo dentro do seu raio de ação, então o sistema Fuzzy reduzirá o peso de suscetibilidade do obstáculo ρ_3 para zero, aumentando assim o peso de outros parâmetros.

6. Trabalhos futuros

Em 2020, as multinacionais, General Motors, Volkswagen, Toyota e Google esperam vender veículos autónomos. Em 2035, prevê-se que 25% dos veículos que circularão nas estradas sejam autónomos, beneficiando da cooperação de veículo-para-veículo. Sabendo desta realidade, torna-se premente a investigação na área da condução autónoma. Tendo em conta esta linha de investigação, existem desafios que precisarão de ser ultrapassados:

- Melhorar a Gestão da Segurança Rodoviária;
- Fazer com que os utilizadores se sintam mais seguros;
- Tornar mais seguras as infraestruturas;
- Veículos mais seguros;
- Melhorar a assistência e o apoio a vítimas.

Portugal tem uma já longa tradição de ser representado em competições internacionais de robôs móveis desde 1995, através de equipas de estudantes e professores do Instituto Superior Técnico (IST), Universidade de Aveiro (UA), Universidade do Minho (UM), Faculdade de Engenharia da Universidade do Porto (FEUP), Faculdade de Engenharia da Universidade de Trás dos Montes e Alto Douro (UTAD).

A investigação com o projeto *CyberMove* no desenvolvimento do Veículo *Move*, pela empresa portuguesa *Move mile* que teve início em 2009 em Coimbra, com o Projeto STOP – Seguranças robóticas coOPERativos, que teve início em 2017 e que irá terminar em 2019, bem como o Projeto SeMFire -Segurança, Exploração e Manutenção de Florestas com Integração de Robótica Ecológica, que teve início no final de 2018 e vai ter a duração de três anos onde se pretende desenvolver um sistema robótico para reduzir o material combustível nas florestas, são exemplos de linhas de investigação relevantes nesta área.

Outro projeto aliciante, também já iniciado, é o de um sistema robótico para a condução autónoma, desenvolvendo simuladores (figura 16). Por exemplo, um simulador de condução autónoma, desenvolvido em Gazebo, com funcionalidades idênticas ao sistema real, com interligação ao Matlab® e ROS de modo a que possam ser testados novos algoritmos de

controlo e comparar desempenhos, torna-se uma aposta interessante e de relevância para esta área. Assim como a construção futura de um cenário de teste para implementar um sistema idêntico ao *Dukietown* e iniciar atividades na área da robótica colaborativa e robótica autónoma *outdoor* (figura 17).

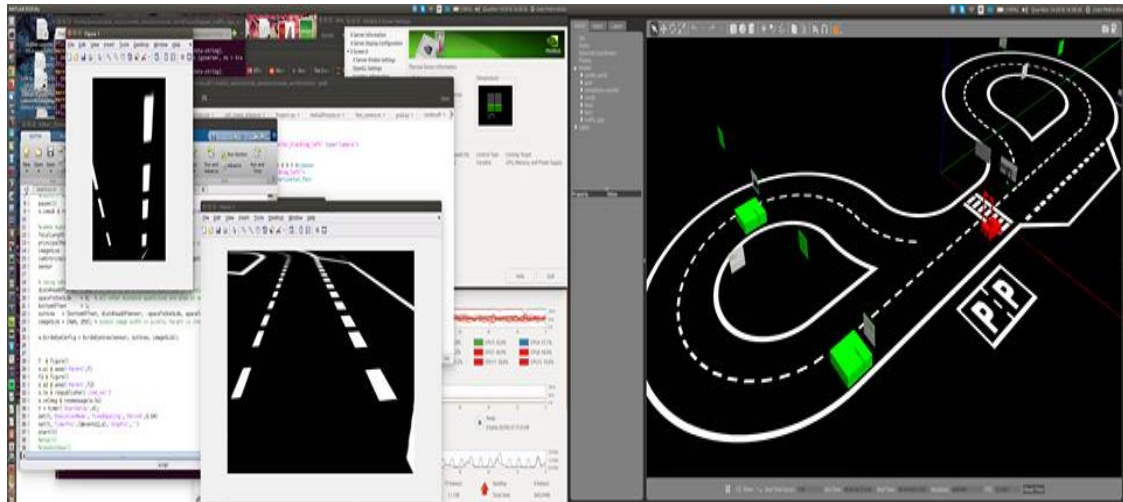


Figura 16 – Simulador da Condução autónoma.



Figura 17 – Condução autónoma *outdoor*.

Anexos

Anexo 1 – Filtro de *Kalman*

O filtro de *Kalman* é um estimador linear que calcula uma estimativa da variância mínima para um estado que evolui ao longo do tempo a partir de observações lineares relacionadas com este estado, assumindo determinados pressupostos lineares relativamente ao ruído do processo e das observações. O filtro de *Kalman* é adequado para aplicações em problemas de navegação, controlo e robótica em geral, sendo reconhecido como um dos melhores estimadores para sistemas lineares com ruído Gaussiano [Lages, 2008].

No entanto o filtro de *Kalman* não é aconselhável para sistemas não lineares. O filtro de *Kalman* estendido (*EKF*) procura resolver esta dificuldade utilizando uma linearização em torno da estimativa do estado. Assim sendo o *EKF* pode ser utilizado para estimar a posição do robô através de dados provenientes da odometria e da observação de referências espaciais.

Na inexistência de mapa o robô apresenta um problema de *SLAM*, as matrizes do *EKF* são alteradas e surge uma vertente, *EKF-SLAM*, descrita de acordo com as seguintes variáveis [Durrant-Whyte and Bailey, 2006]:

- x_k - Vetor de estado que contém a localização e orientação do robô, isto é, a pose do robô.
- u_k - Vetor que contém os valores de saída, resultantes do comando aplicado no instante $k-1$ para mover o robô para a posição x_k no instante k .
- m_k – Vetor das referências que contém a posição estimada das N referências no instante k .
- z_k – Vetor de observações que contém a observação relativa de $n \leq N$ referências no instante k .

O método *EKF-SLAM* descreve a movimentação do veículo de acordo com:

$$P(x_k|x_{k-1}, u_k) \leftrightarrow x_k = f(x_{k-1}, u_k) + w_k \quad (\text{A.1.1})$$

Na equação (A.1.1), f é a função que modela a cinemática do veículo e w_k corresponde às perturbações adicionais ao movimento, assumidas Gaussianas, não correlacionadas, de média nula e com covariância Q_k . O modelo de observação é descrito na forma:

$$P(z_k|x_k, m_k) \leftrightarrow z_k = h(x_k, m_k) + o_k, \quad (\text{A.1.2})$$

onde h é a função que modela a geometria da observação e o_k corresponde aos erros de observação adicionais, considerados Gaussianos, não correlacionados, de média nula e com covariância R_k que afeta as observações.

Com estas definições o método *EKF* pode ser aplicado para calcular a média e a covariância da distribuição posterior $P(x_k, m_k|Z_{0:k}, U_{0:k}, x_0)$ através de uma previsão iterativa e de um algoritmo de correção:

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} = E \begin{bmatrix} x_k \\ m_k \end{bmatrix} | Z_{0:k} \quad (\text{A.1.3})$$

$$P_{k|k} = E \left[\begin{pmatrix} x_k - \hat{x}_k \\ m_k - \hat{m}_k \end{pmatrix} \begin{pmatrix} x_k - \hat{x}_k \\ m_k - \hat{m}_k \end{pmatrix}^T | Z_{0:k} \right] \quad (\text{A.1.4})$$

A etapa de atualização do tempo (previsão) para a média e covariância é dada por:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \quad (\text{A.1.5})$$

$$P_{k|k-1} = \nabla f P_{k-1|k-1} \nabla f^T + Q_k \quad (\text{A.1.6})$$

Em que as estimativas são baseadas na atualização da previsão no instante i e da atualização da observação no instante j . Para tal, $\hat{x}_{k|k}$ representa a estimativa no instante k após a atualização da previsão da observação, e $\hat{x}_{k|k-1}$ representa a estimativa no instante k depois da atualização da previsão, mas antes da atualização da observação.

Na equação (A.1.6), ∇f é o Jacobiano de f avaliado para a estimativa $\hat{x}_{k-1|k-1}$. Uma matriz Jacobiana contém todas as derivadas parciais de primeira ordem de uma função e se a função for diferenciável representa a melhor aproximação linear à função num ponto específico.

Assim a covariância é atualizada utilizando uma aproximação linear do modelo não linear em torno do ponto de operação estimado, que é a diferença fundamental entre o filtro de *Kalman* e o filtro de *Kalman* estendido. Seguindo este passo de previsão, é realizada uma atualização à observação:

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} & \hat{m}_{k-1} \end{bmatrix} + W_k [Z_k - h(\hat{x}_{k|k-1}, \hat{m}_{k-1})] \quad (\text{A1.7})$$

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^T \quad (\text{A1.8})$$

Tal que:

$$S_k = \nabla h P_{k|k-1} \nabla h^T + R_k \quad (\text{A1.9})$$

$$W_k = P_{k|k-1} \nabla h^T S_k^{-1} \quad (\text{A1.10})$$

E onde ∇h representa o jacobiano de h avaliado para $\hat{x}_{k|k-1}$ e para $\hat{m}_{k-1}$.

A solução do *EKF-SLAM* traz muitos benefícios para o problema da navegação, mas também tem limitações, como problemas de seguimento, complexidade computacional, associação de dados e não linearidades. Sob condições ideais, no *EKF-SLAM*, a covariância da estimativa da localização do robô e as posições individuais das referências espaciais convergiriam para zero. Contudo, a complexidade computacional da etapa de correção cresce quadraticamente com o número de referências espaciais, que é usualmente um problema em aplicações práticas. Ainda assim, a solução de *EKF-SLAM* é extremamente sensível à incorreta associação das referências com as observações.

Para além disso, o problema da associação de dados é muito difícil para o *SLAM* quando as referências são re-observadas por pontos de observação muito distintos. Por outro lado, a linearização do modelo não linear do movimento e ou a observação dos modelos pode potenciar resultados com soluções inconsistentes [Thrun *et al.*, 2005].

Em [Smith *et al.*, 1990] apresenta-se pela primeira vez o *EKF* como solução do problema de *SLAM*, tornando-se ao longo dos anos uma abordagem clássica na solução deste problema. Muitos trabalhos foram desenvolvidos com base nesta abordagem, entre os quais podem ser mencionados por exemplo [Guivant *et al.*, 2001; Dissanayake *et al.*, 2002].

Em [Smith *et al.*, 1990] é apresentado um algoritmo otimizado que considera a forma especial das matrizes e um novo filtro comprimido que reduz significativamente os requisitos a nível computacional quando este trabalha em áreas locais ou com sensores externos de alta frequência.

O algoritmo apresentado foi testado com êxito em ambientes reais. Em [Dissanayake *et al.*, 2002] é apresentado um algoritmo de *SLAM* em agentes móveis operando em ambientes externos utilizando radares *MMW (Millimeter-Wave)* para fornecer observações relativas do mapa. Este algoritmo serve para demonstrar que assuntos como a gestão do mapa e associação de dados podem ser tratados em ambientes reais.

Anexo 2 – Algoritmo *FastSLAM* (Filtro de partículas Rao-Blackwellized)

O algoritmo *FastSLAM* surge como alternativa ao EKF, devido a limitações quanto à sua complexidade quadrática e à sensibilidade às falhas na associação de dados.

Este algoritmo foi apresentado em [Montemerlo *et al.*, 2002; Montemerlo *et al.*, 2003], sendo o primeiro a considerar modelos não lineares e distribuições multimodais na solução do problema de SLAM, com a vantagem de ter maior robustez na associação de dados.

O algoritmo *FastSLAM* é baseado numa característica muito importante do problema SLAM, que é a independência condicional que existe entre dois conjuntos diferentes de pontos de referência espaciais no mapa, dada a pose do robô. Por outras palavras, se a trajetória real do robô for conhecida, a estimativa da posição de todas as pontos de referência espaciais do mapa poderia ser feita de modo independente entre cada uma. Isto permite que seja possível a implementação de uma versão do filtro de partículas no problema de SLAM, o filtro de partículas Rao-Blackwellized [Carlone *et al.*, 2010].

A técnica *FastSLAM* utiliza o filtro de partículas para estimar a trajetória do robô, onde cada partícula $x_k^{[i]}$ contém a estimativa da pose do robô, assim como a estimativa de cada característica presente no mapa, que são modeladas por distribuições Gaussianas de média $\mu_{j,k}^{[i]}$ e covariância $\Sigma_{j,k}^{[i]}$, onde k representa uma partícula de um total de M .

Deste modo, o *FastSLAM* pode ser visto como uma combinação entre o filtro de partículas (usado na etapa de previsão) e o EKF, onde cada medida é atualizada mediante o cálculo dos seus parâmetros μ e Σ .

A distribuição posterior que define o problema de SLAM pode ser dividida em duas componentes, uma relacionada com o robô e outra com o mapa do ambiente:

$$P(X_{0:k}|Z_{0:k}, U_{0:k}, x_0) = P(X_{0:k}|Z_{0:k}, U_{0:k}, x_0) \prod_{m=1}^M P(m_m|X_{0:k}, Z_{0:k}) \quad (\text{A.2.1})$$

No caso do *FastSLAM* o cálculo é sobre a trajetória completa. Por isso, as posições dos objetos convertem-se independentemente e podem ser representadas como funções Gaussianas individuais. A figura A.2.1 representa o processo de *FastSLAM*.

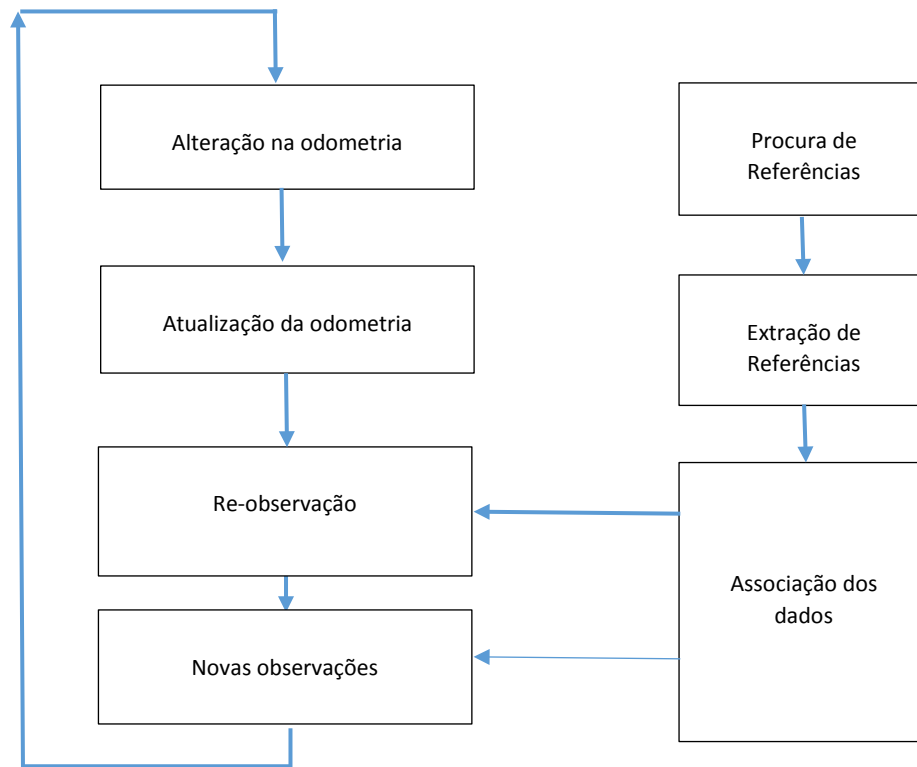


Figura A.2.1. – Representação do processo SLAM [Smith *et. al.*, 1986].

O algoritmo original efetua a amostragem sobre o estado do robô e a associação de dados para cada ponto de referência espacial. O *FastSLAM 2.0* é uma solução mais eficiente, pois melhora a amostragem tendo em conta as medições enquanto as efetua para um novo estado [Montemerlo, *et al.*, 2003].

Através da utilização de estruturas de dados eficientes o *FastSLAM* necessita de um tempo de atualização do mapa de $O(M \log N)$, onde M é o número de partículas e N é o número de pontos de referência espaciais no mapa. Há trabalhos desenvolvidos que usam esta abordagem e existe também versões melhoradas de *FastSLAM* que podem ser encontrados em [Zhang *et al.* 2009; Kim *et. al.*, 2008].

Uma aproximação para resolver o problema de SLAM no caso de cooperação entre robôs, é necessário que estes sejam capazes de comunicar entre si bem como comunicar com um

sistema central, e que os robôs sejam capazes de observar pontos de referência existentes no ambiente, essas marcas definem as posições globais.

Os robôs fazem medições relativas aos pontos de referência no ambiente quando se movem em diferentes trajetórias, a fim de construir em conjunto um mapa comum, por exemplo uma das estratégias é a um filtro de partículas *Rao-Blackwellized* para minimizar as incertezas.

O processo de observação de um ponto de referência é feito da seguinte forma: primeiro, a medição relativa obtida pela câmara instalada no robô, segundo a descrição visual do ponto de referência considerando que o ponto de referência pode ser visto sob diferentes escalas e pontos de vista.

Apesar da informação obtida pelas câmaras ser menos precisa do que as de um sensor laser, este tipo de sistema apresenta um custo bastante menor, adequando-se para uma situação de múltiplos robôs. Além disso, o sistema de visão permite fornecer uma grande quantidade de informações que permitem integrar no robô outras técnicas, como o reconhecimento de rosto ou objeto, entre outras.

Uma observação z_t , no instante t por cada robô, é constituído por $z_t = (v_t, d_t)$, em que $v_t = (X_c, Y_c, Z_c)$ é um vetor 3D relativo a referência da câmara esquerda e d_t é o descritor visual associado ao ponto de referência espacial.

O mapa L é representado por uma coleção de N pontos de referência $L = \{l_1, l_2, \dots, l_N\}$. Cada ponto de referência é descrito como: $l_k = \{\mu_k, \Sigma_k, d_k\}$ em que $\mu_k = \{X_k, Y_k, Z_k\}$ é um vetor que descreve a posição do ponto de referência observado a um referencial global, com uma matriz Σ_k de covariância associada. Além disso, para cada referência espacial l_k é associado um descritor d_k que o diferencia parcialmente dos restantes. Esta representação do mapa é compacta e é usada para localizar um robô em ambientes estruturados.

Seguindo a nomenclatura em *Rao-Blackwellized SLAM*, a posição e orientação do robô como x_t e o movimento do robô como u_t no tempo t ; o caminho do robô é referido como $x^t = \{x_1, x_2, \dots, x_t\}$, o conjunto de observações feitas pelo robô seja $z_t = \{z_1, z_2, \dots, z_t\}$ e o conjunto de ações $u_t = \{u_1, u_2, \dots, u_t\}$. Para determinar a localização de todos os pontos de referência no mapa L , o robô posicionado em x_t a partir de um conjunto de medidas z_t e ações de robô u_t . Assim, pode-se afirmar como a estimativa do posterior:

$$p(x^t, L | z^t, u^t, c^t) \quad (\text{A.2.2})$$

Em que c^t representa o conjunto de associações de dados realizadas, $c_t = \{c_1, c_2, \dots, c_t\}$. Ao explorar um ambiente, o robô tem que determinar se uma observação $z_t = (v_t, d_t)$ corresponde a um ponto de referência previamente mapeado ou se é um novo ponto de referência. Dado que, o mapa é formado por N pontos de referência espaciais, esta correspondência é representada por c_t , onde $c_t \in [1, \dots, N]$, significa que a observação z_t corresponde ao ponto de referência c_t no mapa. Quando não existe correspondência, $c_t = N + 1$, indicando que existe uma nova referência.

O problema do SLAM pode ser separado em duas partes:

- (i) A estimativa da trajetória do robô.
- (ii) A estimativa do mapa por meio de uma série de medições.

No entanto, as duas partes podem ser separadas e, se de alguma forma for possível conhecer a trajetória do robô no ambiente, então a estimativa do mapa seria trivial.

O problema *SLAM* implica:

$$p(x^t L | z^t, u^t, c^t) = p(x^t | z^t, u^t, c^t) \prod_{k=1}^N p(l_k | x^t, z^t, u^t, c^t) \quad (\text{A.2.3})$$

A equação A.2.3 mostra que o *SLAM* é decomposto na estimativa sobre trajetórias dos robôs e N estimadores independentes sobre posições de referência, cada um condicionado à estimativa de trajetória. A utilização de um conjunto de partículas M , em que cada partícula tem N estimadores de referência independentes (implementados com o algoritmo *EKF*), um para cada ponto de referência no mapa, onde se obtêm uma aproximação $p(x^t L | z^t, u^t, c^t)$ em que cada partícula é definida como:

$$S_t^{[m]} \left\{ x^{t,[m]}, \mu_{t,1}^{[m]}, \sum_{t,1} [m], d_1^{[m]}, \dots, \mu_{t,N}^{[m]}, d_N^{[m]} \right\} \quad (\text{A.2.4})$$

onde $\mu_{t,N}^{[m]}$ é a melhor estimativa no tempo t para a posição de referência l_k com base no caminho da partícula m e $\sum_{t,k} [m]$ é a matriz de covariância associada.

O descritor visual associado ao ponto de referência j é representado por $d_j^{[m]}$.

O conjunto de partículas $S_t = \{S_t^{[1]}, S_t^{[2]}, \dots, S_t^{[M]}\}$ é calculado incrementalmente a partir do conjunto S_{t-1} no tempo $t - 1$ e o controlo do robô u_t . Assim, cada partícula é amostrada a partir de uma distribuição proposta $x_t^{[m]} \sim p(x_t | x_{t-1}, u_t)$ que modela o ruído na odometria dos robôs e cada partícula é então atribuída um peso de acordo com:

$$\omega_t^{[m]} = \frac{1}{\sqrt{|2\pi Z_{c_t}|}} e^{\left\{-\frac{1}{2}(v_t - \hat{v}_{t,c_t})^T [Z_{c_t}]^{-1} (v_t - \hat{v}_{t,c_t})\right\}} \quad (\text{A.2.5})$$

onde v_t é a medida real e $\hat{v}_{t,c_t}$ é a medida estimada para o ponto de referência c_t com base na posição $x_t^{[m]}$. A matriz Z_{c_t} é a matriz de covariância associada $(v_t - \hat{v}_{t,c_t})$. Cada medição v_t foi associada ao ponto de referência c_t do mapa. Este problema é, em geral, difícil de resolver, uma vez que podem existir pontos de referência semelhantes.

No caso em que B observações de diferentes pontos de referência existem no tempo, isto é $z_t = \{z_{t,1}, z_{t,2}, \dots, z_{t,B}\}$, sendo necessário cálculo do peso total. Atribuído à partícula como $\omega_t^{[m]} = \prod_{i=1}^B \omega_{t,i}^{[m]}$, onde $\omega_{t,i}^{[m]}$ é o peso associado à observação $z_{t,i}$, calculado usando a Eq. (A.2.5). A seguir a equipa de K robôs explora o ambiente, o objetivo é estimar conjuntamente os caminhos seguidos pelos robôs no mapa.

Numa etapa de tempo, o robô $\langle i \rangle$ está na posição $x_{\langle i \rangle}^t$ e realiza uma única observação $z_{t,\langle i \rangle} = \{v_{t,\langle i \rangle}, d_{t,\langle i \rangle}\}$ do caminho do robô $\langle i \rangle$ até o tempo em que $x_{\langle i \rangle}^t = \{x_{1,\langle i \rangle}, x_{2,\langle i \rangle}, \dots, x_{t,\langle i \rangle}\}$. Para simplificar, $x_{\langle i:K \rangle}^t = \{x_{\langle i \rangle}^t, x_{\langle 2 \rangle}^t, \dots, x_{\langle K \rangle}^t\}$ o conjunto de caminhos da equipa de robôs no tempo t . Analogamente, $u_{\langle i:K \rangle}^t = \{u_{\langle i \rangle}^t, u_{\langle 2 \rangle}^t, \dots, u_{\langle K \rangle}^t\}$ o conjunto de ações realizadas pelos robôs e $z_{\langle i:K \rangle}^t = \{z_{\langle i \rangle}^t, z_{\langle 2 \rangle}^t, \dots, z_{\langle K \rangle}^t\}$ referem-se ao conjunto de observações realizadas pela equipa.

No caso de um único robô, a variável de associação será denotada como c_t , o que indica que a observação $z_{t,\langle i \rangle}$ está associada ao ponto de referência c_t no mapa, a observação está associada apenas ao ponto de referencia no mapa independentemente do robô que o observou. Os dados associação são referidos como $c^t = \{c_1, c_2, \dots, c_t\}$.

No problema *SLAM* para um sistema de múltiplos robôs pode-se estimar de acordo com a seguinte função de probabilidade:

$$\begin{aligned}
& p(x_{<1:K>}^t | L | z_{<1:K>}^t, u_{<1:K>}^t, c^t) \\
& = p(x_{<1:K>}^t | z_{<1:K>}^t, u_{<1:K>}^t, c^t) \prod_{k=1}^N p(l_k | x_{<1:K>}^t, z_{<1:K>}^t, u_{<1:K>}^t, c^t)
\end{aligned} \tag{A.2.6}$$

Esta equação propõe uma forma de estimar um grupo de K caminhos $x_{<1:K>}^t$ no mapa L para os movimentos $u_{<1:K>}^t$ numa série de observações $z_{<1:K>}^t$ associadas aos pontos de referência no mapa c_t .

Analogamente às equações (A.2.3) e (A.2.6) pode-se separar a estimativa do mapa e a estimativa de K caminhos diferentes em duas partes:

- A função $(x_{<1:K>}^t | z_{<1:K>}^t, u_{<1:K>}^t, c^t)$ é estimada usando um filtro de partículas, enquanto o mapa é estimado usando N estimativas independentes aos caminhos $x_{<1:K>}^t$.
- O problema de localização de K robôs no ambiente é uma série de estimativas de pontos individuais dos caminhos dos robôs $x_{<1:K>}^t$. Assim sendo, cada uma das partículas M no filtro é acompanhada por N estimadores independentes para cada um dos pontos de referência espaciais.

Para os filtros de *Kalman* serão condicionados os caminhos K dos robôs, em que cada partícula é representada como:

$$S_t^{[m]} \left\{ x_{<1:K>}^{t,[m]}, \mu_{t,1}^{[m]}, \sum_{1,t}^{[m]} d_1^{[m]}, \dots, \mu_{N,t}^{[m]}, \sum_{N,t}^{[m]} d_N^{[m]} \right\} \tag{A.2.7}$$

A diferença em relação à partícula é definida em (A.2.7) e o estado que se pretende estimar é composto pela posição e orientação (x, y, θ) dos K robôs, assim $x_{t<1:K>} = \{x_{t<1>}, x_{t<2>}, \dots, x_{t<K>}\}$ o resultado, apresenta-se a estimativa conjunta sobre um estado de caminho da dimensão $3K$.

Segundo [Thrun *et al.*, 2005], o número de partículas necessárias para se obter uma boa estimativa aumenta exponencialmente com a dimensão do estado. No entanto, a abordagem funciona perfeitamente para equipas de 2 a 3 robôs com um número razoável de partículas.

O mapa é compartilhado por todos os robôs, o que significa que uma observação realizada por um robô, em particular, afeta o mapa de toda a equipa. No entanto, um membro da equipa pode observar um ponto de referência previamente mapeado por outro robô e atualizar a sua

estimativa. Além disso, significa que um robô não necessita de retornar à origem para reduzir a incerteza na sua posição e orientação.

Um robô pode reduzir a incerteza da sua pose quando observa pontos de referência mapeados por outros robôs. O método baseado num filtro de partículas *Rao-Blackwellized* pode ser decomposto em 4 etapas básicas:

- Criar um novo conjunto de partículas com base no conjunto anterior.
- Atualizar a estimativa de cada ponto de referência com base nas observações.
- Calcular o peso para cada partícula.
- Realizar uma reamostragem com base no peso de cada partícula.

Anexo 3 – Modelo do enxame de robôs

No modelo do enxame, considera-se que cada partícula n move-se em um espaço multidimensional de acordo com os valores de posição (x_t^n) e velocidade (v_t^n) que são altamente dependentes das melhores informações locais ($\check{x}_t^n$), Vizinhança melhor ($\check{n}_t^n$) e melhor ($\check{g}_t^n$) global:

$$v_{t+1}^n = wv_t^n + \rho_1 r_1 (\check{g}_t^n + x_t^n) + \rho_2 r_2 (\check{x}_t^n + x_t^n) + \rho_3 r_3 (\check{n}_t^n + x_t^n) \quad \text{A.3.1}$$

$$x_{t+1}^n = x_t^n + v_{t+1}^n \quad \text{A.3.2}$$

Os coeficientes w , ρ_1 , ρ_2 e ρ_3 atribuem pesos à influência inercial, ao melhor global, ao melhor local e à vizinhança, quando determinam a nova velocidade.

Normalmente, a influência inicial é definida como um valor ligeiramente menor que 1, ρ_1 , ρ_2 e ρ_3 são valores inteiros constantes, que representam componentes cognitivos e “sociais” dos robôs.

$$SC_s = SC_s \left[1 - \frac{1}{N_s^{kill} + 1} \right] \quad \text{A.3.3}$$

com o número de partículas excluídas N_s^{kill} do subgrupo s durante um período em que não houve melhoria. Um subgrupo para criar um novo subgrupo, não deve ter nenhuma partícula excluída e o número máximo de subgrupos não deve ser excedido. Ainda assim, o novo subgrupo é criado apenas com uma pequena probabilidade de:

$$\rho_{sp} = \frac{r_{sp}}{NS} \quad \text{A.3.4}$$

em que r_{sp} é um número aleatório entre 0 e 1 e NS o número de subgrupos. Esse fator evita a criação de subgrupos mais novos quando há um grande número de subgrupos existentes. Todas essas regras pertencem ao mecanismo de “punição-recompensa” do DPSO. Portanto,

criar o modelo um enxame robótico, cada robô move-se num espaço multidimensional o RDPSO apresenta o sistema de equação discreta (DE):

$$v_n[t+1] = w_n[t]v_n[t] + \sum_{i=1}^4 \rho_i r_i (x_i[t] - x_n[t]) \quad \text{A.3.5}$$

$$x_n[t+1] = x_n[t] + v_n[t+1] \quad \text{A.3.6}$$

em que $w_n[t]$ e ρ_i , $i = 1, 2, 3, 4$, atribuem pesos à influência inercial, ao melhor local (componente cognitivo), ao melhor global (componente social), à componente de evitação de obstáculos e ao componente de comunicação vigente ao determinar a nova velocidade, com $\rho_i > 0$. Como antes, r_i são vetores aleatórios onde em cada componente é geralmente um número aleatório uniforme entre 0 e 1.

E $v_n[t]$ e $x_n[t]$ representam a velocidade e o vetor de posição do robô n , respetivamente. O robô recebe a posição desejada, que corresponde a uma posição cartesiana, e calcula o modelo cinemático inverso baseado nas seguintes equações:

$$h_n[t+1] = \sqrt{((v_n^1[t+1])^2 + (v_n^2[t+1])^2)} \quad \text{A.3.7}$$

$$\theta_n^d[t+1] = \text{atan2}(v_n^2[t+1], v_n^1[t+1]) \quad \text{A.3.8}$$

$$\theta_n[t+1] = \theta_n^d[t+1] - \theta_n^d[t] \quad \text{A.3.9}$$

em que $v_n^1[t+1]$ e $v_n^2[t+1]$ são os elementos do vetor $v_n[t+1]$ da equação (A.3.7). A função atan2 em (A.3.8) é uma variante da função arco tangente trigonométrica, mas é responsável pelo quadrante em que $\theta_n^d[t+1]$ se encontra.

$$\tau_n^{d1}[t+1] = \tau_{rev} \frac{\theta_n[t+1]}{2\pi} \frac{R_{robot}}{R_{wheel}} \quad \text{A.3.10}$$

$$\tau_n^{d2}[t+1] = \tau_{rev} \frac{h_n[t+1]}{2\pi} \frac{1}{R_{wheel}} \quad \text{A.3.11}$$

Em que τ_{rev} é o número total de pulsos por revolução, o raio do robô e das rodas são definidos por R_{robot} e R_{wheel} , respetivamente, e a fim de melhorar a resposta temporal a suavidade do movimento do robô, criou-se um limiar rotacional de θ_T .

Anexo 4 – Cinemática do robô diferencial

Considerando que a odometria é uma das mais importantes formas de localização na robótica móvel e tem como base a integração da informação incremental do movimento rotacional das rodas. Apesar de não possuir grande precisão a longo prazo devido à acumulação de erros, tanto de natureza aleatória como sistemática, esta permite a curto prazo obter uma boa estimativa do deslocamento do robô. Contudo, como é um método simples de implementar e considerando uma elevada frequência de amostragem, e com auxílio de outros métodos de posicionamento absoluto contribui para uma estimativa de posição.

No caso do robô diferencial, figura A.4.1, os principais parâmetros de odometria são o raio das rodas R_{wheel} e o raio do robô (metade da distância roda-roda) R_{robot} figura A.4.2.



Figura A.4.1 – Robô diferencial

A rotação ϑ é então calculada como a diferença entre a variação da função de arco tangente atan2 e a orientação real do robô. Para minimizar a energia consumida, o robô executa sempre o ângulo mínimo necessário para se orientar para a posição alvo, a distância euclidiana, $dist$, é então calculada entre a posição atual e a posição alvo (isto é, a distância entre dois pontos). Tanto o ângulo quanto a distância que o robô precisa alcançar são as entradas da rotação.

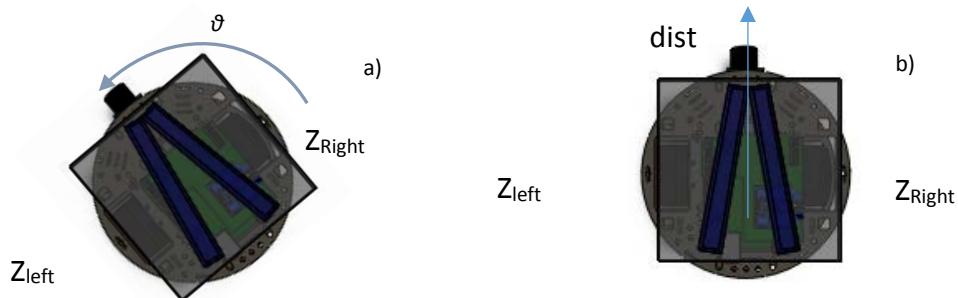


Figura A.4.2 – Robô diferencial.

O ângulo desejado ϑ de radianos para pulsos $z_{Right} = -z_{Left}$, de acordo com a seguinte equação:

$$z_{Right} = -z_{Left} = N_{pulses/revolution} \times \frac{\theta}{2\pi} \times \frac{R_{robot}}{R_{wheel}} \quad (A.4.1)$$

Considerando que $R_{wheel} = 21.09$ mm, $R_{robot} = 44.50$ mm e a combinação entre o encoder e a roda fornece $N_{pulses/revolution} = 48$ pulsos/revolução, a resolução é de 3,6 graus para um único pulso.

A distância euclidiana, $dist$, de milímetros para pulsos $z_{Right} = z_{Left}$ usando a seguinte equação:

$$z_{Right} = z_{Left} = N_{pulses/revolution} \times \frac{dist}{2\pi \times R_{wheel}} \quad (A.4.2)$$

Com os parâmetros mencionados anteriormente, uma resolução de 2,76 mm é então obtida para um único pulso. A velocidade desejada de cada roda é controlada independentemente usando um sinal de Modulação de Largura de Pulso que é geralmente proporcional ao ciclo de trabalho. No entanto, esta proporcionalidade depende muito das características físicas dos motores.

Como os motores podem não ser idênticos, é impossível assegurar que a aplicação da mesma tensão nos dois motores resulte na mesma velocidade. Por outro lado, se os motores não tiverem a mesma velocidade enquanto se deslocam para frente, o robô irá girar. Para resolver este tipo de problemas, implementa-se um controlador proporcional para controlar a velocidade, de acordo com a seguinte equação:

$$\begin{cases} v_{Right} = v_{base} - k \cdot (z_{Right}^r - z_{Left}^r) \\ v_{Left} = v_{base} + k \cdot (z_{Right}^r - z_{Left}^r) \end{cases} \quad (A.4.3)$$

onde z_{Right}^r e z_{Left}^r são o número real de pulsos de cada roda em cada iteração. Um ganho proporcional de $k = 3$ foi considerado como fator de correção adequado.

Referências Bibliográficas

Arkin, R. and Diaz, J. (2002). "Line-of-sight constrained exploration for reactive multiagent robotic teams". *Proceedings of the Int. Workshop on Advanced Motion Control (AMC'02)*, Maribor, Slovenia.

Bakhshipoura, M.; Ghadib M. J. and Namdaria F. (2017). "Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach". *Journal Applied Soft Computing*, 57, 708–726.

Bourgault, F.; Makarenko, A.; Williams, S.; Grocholsky, B. and Durrant-Whyte, F. (2002). "Information based adaptive robotic exploration". *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland.

Carlone, L.; Kaouk, M.; Du, J.; Bona, B. and Indri, M. (2010). "Rao-Blackwellized particle filters multi robot slam with unknown initial correspondences and limited communication". *IEEE International Conference on Robotics and Automation*, 243-249.

Chaimowicz, L.; Kumar V. and Campos, M. F. M. (2004). "A Paradigm for Dynamic Coordination of Multiple Robots", *Autonomous Robots* 17(1), 7-21.

Choset, H. and Burdick, J. (2000). "Sensor-based exploration: The hierarchical generalized voronoi graph". *Int. Journal of Robotics Research*, 19(2), 96-125.

Couceiro M.S.; Rocha R.P.; Fonseca Ferreira N. M. (2011a), "Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Particle Swarms", *Proceedings of the 2011 IEEE International Symposium on Safety, Security and Rescue Robotics*, 1-5 Nov, Kyoto, Japan, 284-289.

Couceiro, M. S.; Rocha, R.P. and Fonseca Ferreira, N. M. (2011b). "A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms". *Proceedings of the 2011 IEEE International Symposium on Safety, Security and Rescue Robotics*, 1-5 Nov, Kyoto, Japan, 327-332.

Couceiro M.S.; Rocha, R.P.; Figueiredo, C. M.; Luz J.M.A. and Fonseca Ferreira N. M. (2012a). "Multi-Robot Foraging based on Darwin's Survival of the Fittest", *Proceedings of IROS'12, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7-12 Oct, Vilamoura, Portugal, 801-806.

Couceiro, M.S.; Figueiredo C.M.; Portugal D.; Rocha R.P. and Fonseca Ferreira N. M. (2012b). "Initial Deployment of a Robotic Team - A Hierarchical Approach Under Communication Constraints Verified on Low-Cost Platforms". *Proceedings of IROS'12, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7-12 Oct, Vilamoura, Portugal, 4614-4619.

Couceiro, M.S.; Tenreiro Machado, J.A.; Rocha R.P. and Fonseca Ferreira, N. M. (2012c). "A fuzzified systematic adjustment of the robotic Darwinian PSO". *Robotics and Autonomous Systems*, 60 (12), 1625-1639.

Couceiro M.S.; Martins, F. M. L.; Rocha R.P. and Fonseca Ferreira, N. M. (2012d). "Analysis and Parameter Adjustment of the RDP SO - Towards an Understanding of Robotic Network Dynamic Partitioning based on Darwin's Theory", *International Mathematical Forum, Hikari, Ltd.*, 7(32), 1587-1601.

- Couceiro, M.S.; Figueiredo, C.M.; Luz J. M. A.; Fonseca Ferreira, N. M. and Rocha, R.P. (2012e). "A Low-Cost Educational Platform for Swarm Robotics", *International Journal of Robots, Education and Art, IJREA*, 2(1), 1-15.
- Couceiro M.S.; Rocha, R.P.; Fonseca Ferreira, N. M. and Vargas P.A. (2013a). "Darwinian Robotic Swarms for Exploration with Minimal Communication". *Proceedings of CEC 2013, 2013 IEEE Congress on Evolutionary Computation, Special Session on Evolutionary Robotics*, 20-23 June, Cancun, Mexico, 127-134.
- Couceiro M.S.; Rocha, R.P. and Fonseca Ferreira, N. M. (2013b). "A PSO Multi-Robot Exploration Approach Over Unreliable MANETs". *Advanced Robotics*, 27 (16), 1221 – 1234.
- Couceiro, M.S.; Portugal, D.; Rocha, R.P. and Fonseca Ferreira, N. M. (2014a). "Marsupial Teams of Robots: Deployment of Miniature Robots for Swarm Exploration under Communication Constraints". *Robotica, Cambridge University Press*, 32, issue 7, 1017-1038.
- Couceiro M. S.; Vargas, P.A.; Rocha, R.P. and Fonseca Ferreira, N. M. (2014b). "Benchmark of Swarm Robotics Distributed Techniques in a Search Task". *Robotics and Autonomous Systems*, 62, issue 2, 200-213.
- Couceiro, M.S.; Martins, F. M. L.; Rocha, R.P. and Fonseca Ferreira, N. M. (2014c). "Mechanism and Convergence Analysis of a Multi-Robot Swarm Approach Based on Natural Selection". *Journal of Intelligent & Robotic Systems*, 76(2), 353-381.
- Couceiro, M.S.; Figueiredo, C. M.; Rocha, R.P. and Fonseca Ferreira, N. M. (2014d). "Darwinian swarm exploration under communication constraints: initial deployment and fault-tolerance assessment". *Robotics and Autonomous Systems*, 62(4), 528-544.
- Couceiro, M.S.; Araújo, A.G.; Tatarian, K. and Fonseca Ferreira, N.M. (2018). "RobotCraft: The first international collective internship for advanced robotics training". *Proceedings of RiE2018, 9th International Conference on Robotics in Education*, Malta, 18-20 April.
- Dissanayake, M.W.M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H. F. and Csorba, M. (2002). "A solution to the simultaneous localization and map building (SLAM) problem". *IEEE Transactions on Robotics and Automation*, 17(3), 229–241.
- Dudek, G. and Jenkin, M. (2000). "Computational Principles of Mobile Robotics". *Cambridge, London, UK: The MIT Press*, 280 p.
- Durrant-Whyte, H. and Bailey, T. (2006). "Simultaneous localisation and mapping (slam): Part i the essential algorithms". *IEEE Robotics and Automation Magazine*, 2.
- Elfes, A. (1989). "Using occupancy grids for mobile robot perception and navigation". *Computer*, vol. 22(6), 46-57.
- Everett, H. R. and Gage, D. W. (1999). "From Laboratory to Warehouse: Security Robots Meet the Real World," *Int. Journal of Robotics Research*, 18(7), Jul., 760-768.
- Farinelli, A.; Iocchi, L. and Nardi, D. (2004). "Multirobot systems: a classification focused on coordination". *IEEE Trans. Syst. Man Cybernetics B*, 34 (5), 2015-2028.
- Feder, H.; Leonard, J. and Smith, C. (1999). "Adaptive mobile robot navigation and mapping". *Int. Journal of Robotics Research*, 18(7), 650-668.

Fonseca Ferreira N. M., (2006), "Sistemas Dinâmicos e Controlo de Robôs Cooperantes", *Phd Thesis*, University of Trás-os-Montes and Alto Douro.

Fox, D.; Burgard, W. and Thrun, S. (1999a). "Markov localization for reliable robot navigation and people detection". *Modeling and Planning for Sensor-Based Intelligent Robot Systems*. Springer Verlag, Berlin.

Fox, D.; Burgard, W.; Dellaert, F. and Thrun, S. (1999b). "Monte Carlo Localization: Efficient position estimation for mobile robots". *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Orlando, FL. AAAI.

Fox, D.; Ko, J.; Limketkai, B.; Schulz, D. and Stewart, B. (2006). "Distributed multi-robot exploration and mapping". *Proceedings of the IEEE*, 94(7), 1325-1339.

Frese U. (2006). "A discussion of simultaneous localization and mapping". *Autonomous Robots*, 20 (1), 25-42.

Fukuda T. and Nakagawa S. (1987). "A dynamically reconfigurable robotic system -concept of a system and optimal configurations". *International Conference on Industrial Electronics, Control, and Instrumentation*, 588-595.

Gerkey, B. and M. Mataric (2004). "A formal analysis and taxonomy of task allocation in multi-robot systems". *International Journal of Robotics Research*, 23(9), 939-954.

Ghamisi, P.; Couceiro, M.S.; Fonseca Ferreira, N. M. and Kumar, L. (2012). "Use of Darwinian Particle Swarm Optimization Technique for the Segmentation of Remote Sensing Images". *Proceedings of IGARSS'2012, IEEE International Geoscience and Remote Sensing Symposium, Remote Sensing for a Dynamic Earth*, 22-27 July, Munich, Germany, 4295-4298.

Grabowski, R.; Khosla, P. and Choset, H. (2003). "Autonomous exploration via regions of interest". *Proceedings of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03)*, Las Vegas, Nevada, US.

Guivant, J. and Nebot, E. (2001). "Optimization of the simultaneous localization and map building algorithm for real time implementation". *IEEE Transactions on Robotics and Automation*, 17(3), 242-257.

Hartigan, J. and Wong, M. (1979). "A k-means clustering algorithm". *Journal of the Royal Statistical Society, Series C, Applied statistics*, 28, 100-108.

Juliá, M.; Gil, A.; Payá, L. and Reinoso, O. (2008). "Emerging Technologies, Robotics and Control Systems", Second edition, *International Society for Advanced Research*, chapter "Local Minima Detection in Potential Field Based Cooperative Multi-robot Exploration", 17-22.

Junior, F.C.F.M.; Sousa, J. V.M. and Fonseca Ferreira, N.M. (2018). "Computer Vision System with Deep Learning for Robotic Arm Control". *IEEE LARS 2018 – 15th Latin American Robotics Symposium, SBR 2018 – 6th Brazilian Robotics Symposium*.

Kelley, R.; Tavakkoli, A.; King, C.; Nicolescu, M.; Nicolescu, M. and Bebis, G. (2008). "Understanding human intentions via hidden markov models in autonomous mobile robots". *Proceedings of the 3rd International Conference on Human Robot Interaction - HRI '08*.

Kim, C.; Sakhivel, R and Chung, W. K. (2008). "Unscented fastslam: A robust and efficient solution to the slam problem". *IEEE Transactions on Robotics*, 24(4), 808-820.

Lages, W. F., (2008). “Filtro de Kalman”. *Universidade Federal do Rio Grande do Sul Escola de Engenharia, Departamento de Engenharia Elétrica*.

Latombe, J. (1991). *Robot Motion Planning*. Kluwer Academic Publisger, Boston, MA.

Lau, H. (2003). “Behavioural approach for multi-robot exploration”. *Proceedings Of the Australasian Conf. on Robotics and Automation (ACRA'03)*, Brisbane, Australia.

Leonard, J. J. and Durrant-White, H. F. (1991). “Mobile Robot Localization by tracking Geometric Beacons”. *IEEE Transactions on robotics and Automation*, vol.7, 376-382.

Liu M.; Sivakumar, K.; Omidshafiei S.; AmatoC. and How J. P. (2017). “Learning for Multi-robot Cooperation in Partially Observable Stochastic Environments with Macro-actions”, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017)*.

Lobo, J. and Dias, J. (2007). “Relative pose calibration between visual and inertial sensors”. *International Journal of Robotics Research*, 26(6), 561–575.

Makarenko, A.; Williams, S.; Bourgoult, F. and Durrant-Whyte, F. (2002). “An experiment in integrated exploration”. *Proceedings of the IEEE-RSJ Int. Conf. On Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland.

Marayong, P. (2007). “Motion Control Methods for Human-Machine Cooperative Systems”. *PhD Thesis*, John Hopkins University, USA.

Marcolino, L. S. and Chaimowicz, L. (2008). “No robot left behind: Coordination to overcome local minima in swarm navigation”. *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, 1904–1909.

Martins G. S.; Portugal D. and Rocha R.P. (2014). “A Comparison of General-Purpose FOSS Compression Techniques for Efficient Communication in Cooperative Multi-Robot Tasks”. *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO*, 136-147.

Massachusetts Institute of Technology, (2015)” New algorithm lets autonomous robots divvy up assembly tasks on the fly”, ScienceDaily. ScienceDaily, 27 May 2015. <www.sciencedaily.com/releases/2015/05/150527142100.htm>.

Mataric, M. J. (1990). “A distributed model for mobile robot environment-learning and navigation”. *Master’s thesis, MIT*, Cambridge, MA.

Montemerlo, M.; Thrun, S.; Koller, D. and Wegbreit, B. (2002). “FastSLAM: a factored solution to the simultaneous localization and mapping problem”. *AAAI National Conference on Artificial Intelligence*.

Montemerlo, M. and Thrun, S. (2003). “Simultaneous localization and mapping with unknown data association using FastSLAM”. *Proceedings of the IEEE Int. Conf. On Robotics and Automation (ICRA'03)*, Taipei, Taiwan.

Montemerlo, M.; Thrun, S.; Koller, D. and Wegbreit, B. (2003). “Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges”. *International Joint Conferences on Artificial Intelligence*.

Oh, Hyondong; Ramezan Shirazi, A.; Sun, C. and Jin, Y. (2017). “Bio-inspired self-organising multi-robot pattern formation: A review.” *Robotics and Autonomous Systems*, 91, 83–100.

Page, B. R.; Ziaeeefard, S.; Moridian, B. and Mahmoudian, N. (2017). "Learning autonomous systems — An interdisciplinary project-based experience". *2017 IEEE Frontiers in Education Conference (FIE)*.

Paull, L.; Tani, J.; Ahn, H.; Alonso-Mora, J.; Carlone, L.; Cap, M. and Censi, A. (2017). "Duckietown: An open, inexpensive and flexible platform for autonomy education and research". *2017 IEEE International Conference on Robotics and Automation (ICRA)*.

Portugal D. and Rocha R.P. (2013). "Distributed multi-robot patrol: A scalable and fault-tolerant framework", *Robotics and Autonomous Systems*, 61(12), 1572–1587.

Portugal, D.; Iocchi, L. and Farinelli, A. (2018). "A ROS-Based Framework for Simulation and Benchmarking of Multi-robot Patrolling Algorithms". *Robot Operating System (ROS)*, 3–28.

Rekleitis, I.; Sim, R.; Dudek, G. and Milios, E. (2001). "Collaborative exploration for the construction of visual maps". *Proceedings of the IEEE-RSJ Int. Conf. On Intelligent Robots and Systems (IROS'01)*, Wailea, Hawaii.

RoboCup Standard Platform League (2018) NAO Rule Book. <https://spl.roboocup.org/wp-content/uploads/downloads/2018/07/NAO-Rule-Book-2018.pdf>

Rocha, R.; Dias, J. and Carvalho, A. (2005). "Cooperative Multi-Robot Systems: a study of Vision-based 3-D Mapping using Information Theory". *Robotics and Autonomous Systems*, 53(3-4), 282-311.

Rosman, G.; Paull, L. and Rus, D. (2017). "Hybrid control and learning with coresets for autonomous vehicles". *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Salomon D. (2007). "A concise introduction to data compression". *Springer*.

Saraiva, A.; Fonseca Ferreira, N.M. and Valente, A. (2018). "Scalable Task Cleanup Assignment for Multi-agents". *Proceedings of CLAWAR 2018, 21st International Conference on Climbing and Walking Robots and Support Technologies for Mobile Machines*, 10–12 Sept, Universidad Tecnológica de Panamá, Panamá.

Schultz, A.; Adams, W. and Yamauchi, B. (1999). "Integrating exploration, localization, navigation and planning with a common representation". *Autonomous Robots*, 6, 293-308.

Sheng, W.; Yang, Q.; Tan, J. and Xi, N. (2006). "Distributed multi-robot coordination in area exploration". *Robotics and Autonomous Systems*, 54, 945-955.

Sim, R. and Little, J. (2006). "Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters". *Intelligent Robots and Systems*, 2006 IEEE/RSJ International Conference, 2082-2089.

Smith, R. C. and Cheeseman, P. (1986). "On the representation and estimation of spatial uncertainty". *The international Journal of Robotics Research*, 56-68.

Smith, R.; Self, M. and Cheeseman, P. (1990). "Autonomous robot vehicles". Chapter Estimating uncertain spatial relationships in robotics, *Springer-Verlag New York, Inc.*, New York, NY, USA, 167-193.

- Solanas, A. and Garcia, M. (2004). "Coordinated multi-robot exploration through unsupervised clustering of unknown space". *Proceedings of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (IROS'04)*, Sendai, Japan, 717-721.
- Stachniss, C.; Grisetti, G. and Burgard, W. (2005). "Information gain-based exploration using rao-blackwellized particle filters". *Proceedings of Robotics: Science and Systems (RSS'05)*, Cambridge, MA, USA.
- Thrun, S.; Burgard, W. and Fox, D. (2005). "Probabilistic Robotics". *Cambridge: The MIT Press*, 667.
- Tovar, B.; Munoz-Gomez, L.; Murrieta-Cid, R.; Alencastre-Miranda, M.; Monroy, R. and Hutchinson, S. (2006). "Planning exploration strategies for simultaneous localization and mapping". *Robotics and Autonomous Systems*, 54(4), 314-331.
- Vig, L. and Adams, J. A. (2006). "Multi-robot coalition formation". *IEEE Trans. on Robotics*, 22(4), 637-649.
- Wurm, K. M.; Stachniss, C. and Burgard, W. (2008). "Coordinated multi-robot exploration using a segmentation of the environment". *Proceedings of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (IROS'08)*, Nice, France.
- Zadeh, L. A. (1965). "Fuzzy Sets". *Information and Control*, 8(3), 338-353.
- Zhang, L.; Meng, X. J. and Chen, Y. W. (2009). "A fastslam algorithm based on the auxiliary particle filter with stirling interpolation". *Proceedings of the 2009 IEEE International Conference on Information and Automation*, 167-172.
- Ziv J. and Lempel A. (1977). "A universal algorithm for sequential data compression". *IEEE Transactions on information theory*, 23(3), 337-343.
- Zlot, R.; Stentz, A.; Dias, M. B. and Thayer, S. (2002). "Multi-robot exploration controlled by a market economy". *Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA'02)*, Washington DC, USA.