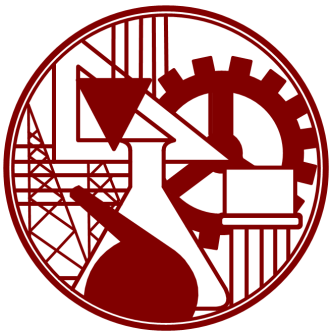
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de

Engenharia de Electrónica e Telecomunicações e de Computadores



Integração de Reacção e Deliberação em Agentes Inteligentes

Carlos António Batista Lopes Junior

(Licenciado)

Trabalho Final de Mestrado para obtenção do grau de Mestre em

Engenharia de Redes de Comunicação e Multimédia

Orientador:

Professor Doutor Luís Filipe Graça Morgado

Júri:

Presidente: Professor Doutor Paulo Manuel Trigo Cândido da Silva
Vogal: Professor Doutor Arnaldo Joaquim Castro Abrantes

Dezembro de 2014

Resumo

A integração de reacção e deliberação é um dos aspectos centrais em modelos de agentes inteligentes, nomeadamente em modelos de agente híbridos. O conceito de agente inteligente híbrido, surgiu nos anos 90, numa tentativa de combinar o melhor de dois mundos, após o percurso da inteligência artificial ter passado pelos paradigmas hierárquico e reactivo. Este tipo de agente é composto principalmente por duas camadas: reactiva e deliberativa. A camada reactiva reage a estímulos do ambiente enquanto a deliberativa tem um carácter pró-activo, utilizando um modelo interno para gerar planos de alto nível.

Nesta dissertação propõe-se um modelo de agente inteligente híbrido constituído por uma camada adicional de carácter adaptativo. Essa camada tem por objectivo disponibilizar um nível de competência, no qual o agente tem capacidade de aprender com a experiência, complementando as competências nos níveis reactivo e deliberativo.

A arquitectura do modelo de agente proposto tem uma organização vertical, onde somente a camada reactiva interage com o ambiente. Esta é responsável por adquirir a percepção e gerar uma acção resultante da composição de vários comportamentos. Já a camada adaptativa recebe a percepção da camada reactiva e efectua uma discretização, que irá suportar a aprendizagem resultante da interacção com o ambiente, de modo a complementar a camada reactiva nas suas limitações. Por último, a camada deliberativa recebe a percepção do nível hierárquico inferior, com base na qual gera uma representação interna do mundo para suporte dos mecanismos deliberativos.

Deste modo, pretende-se ter um modelo agente capaz de responder às necessidades de operação em tempo real em cenários complexos e dinâmicos.

Palavras-chave: Agentes inteligentes, modelos de agentes híbridos, aprendizagem por reforço.

Abstract

The integration of reaction and deliberation is one of the main aspects in intelligent agent models, namely in hybrid agent models. The concept of hybrid intelligent agents emerged during the 90's, in attempt to combine the best of both worlds after the area of artificial intelligence passed through the hierarchical and reactive paradigms. These kinds of agents are mainly composed of two layers: reactive and deliberative. The reactive layer reacts to stimuli while the deliberative layer has a proactive character using an internal model to generate higher level plans.

In this document a hybrid intelligent agent model is proposed, composed by an additional layer of adaptive character. This layer aims to provide a level of competence in which the agent has ability to learn from experience, complementing the abilities in reactive and deliberative layers.

The proposed model's architecture has a vertical organization, in which only the reactive layer interacts with the environment. This layer is responsible for acquiring perception and generating actions as a result of the composition of many behaviors. The adaptive layer receives the perception from the reactive layer and performs a discretization which will support learning resulting from interaction with the environment, so it complements the reactive layer in their limitations. Lastly, the deliberative layer receives the perception from a hierarchical lower level, in which generates a world inner representation to support the deliberative mechanisms.

In this way, the aim is to obtain an agent model capable of responding to real-time requirements of operation in complex and dynamic environments.

Keywords: Intelligent agents, hybrid agent models, reinforcement learning.

Agradecimentos

Inicialmente gostava de agradecer ao Professor Luís Morgado pela dedicação e profissionalismo ao orientar-me durante todas as fases de desenvolvimento desta dissertação. Fornecendo, de forma sábia, sugestões úteis para que todo o processo se concretizasse.

Ao meu grande amigo Diogo Lopes, que ao longo do curso mostrou-se ser uma das pessoas mais altruístas que conheço, dispensando o seu tempo e esforço a ajudar quem necessitasse. Um amigo dedicado e leal que, mesmo sem saber, muito me ensinou pelas suas acções e atitudes. Ele foi, certamente, uma grande ajuda ao longo do desenvolvimento deste trabalho na partilha de opiniões e discussão de ideias.

A todos os meus amigos "iselianos" que me acompanharam durante no percurso nesta fase da minha vida, e que de algum modo, me ajudaram a crescer como pessoa, aluno e profissional.

A todos os professores do ISEL pela sabedoria partilhada e pela vossa participação no meu percurso académico.

Aos meus pais, Carlos e Mira, que sem eles nada disto seria possível, eles que sempre acreditaram em mim, mesmo sem um acompanhamento constante da minha vida académica, sei que apoiam incondicionalmente as minhas decisões.

Ao meu filho, Simão, que é sem dúvida a minha alegria de viver e a minha maior motivação para que pudesse cumprir o meu objectivo, na esperança de lhe proporcionar uma vida sem grandes dificuldades para que continue forte e saudável como sempre foi. Espero que, de algum modo, no futuro, esta fase da minha vida o possa inspirar e motivar a seguir os seus sonhos e a acreditar ser capaz de concretizar os seus próprios objectivos desde que haja trabalho e dedicação.

E agradeço a todos, a quem não mencionei o nome, que participaram e me ajudaram, de alguma forma, na minha caminhada.

Um muito obrigado a todos.

Índice Geral

1 INTRODUÇÃO 1

1.1 CONTEXTO 2

1.2 MOTIVAÇÃO 2

1.3 OBJECTIVOS 2

1.4 ORGANIZAÇÃO DO DOCUMENTO 3

2 ENQUADRAMENTO TEÓRICO 5

2.1 ARQUITECTURA DE AGENTES INTELIGENTES 5

2.1.1 Agentes Reactivos 6

2.1.2 Agentes Deliberativos 7

2.1.3 Agentes Híbridos 8

2.2 APRENDIZAGEM POR REFORÇO 10

2.3 PROCESSOS DE DECISÃO DE MARKOV 12

2.4 CONCLUSÃO 13

3 TRABALHO RELACIONADO 15

3.1 REAL-TIME CONTROL SYSTEM (RCS) 15

3.1.1 4D/RCS Arquitectura de Modelo de Referência 16

3.2 INTERRAP 25

3.2.1 Representação do conhecimento 27

3.2.2 Interacção entre camadas 28

3.2.3 Tomada de decisão 28

3.2.4 Restrições de tempo real 30

3.3 CONCLUSÃO 32

4 MODELO DE AGENTE PROPOSTO 33

4.1 ORGANIZAÇÃO GERAL DO MODELO PROPOSTO 33

4.2 SUBSISTEMA REACTIVO 37

4.2.1 Coordenador Sensorial 37

4.2.2 Controlo Reactivo 39

4.2.3 Coordenador de Acção 40

4.3 SUBSISTEMA ADAPTATIVO 43

4.3.1 Coordenador Sensorial 43

4.3.2 Estrutura Cognitiva 44

4.3.3 Controlo Adaptativo 45

4.3.4 Coordenador de Acção 47

4.4 Subsistema Deliberativo 47
4.4.1 Coordenador Sensorial *48*
4.4.2 Estrutura cognitiva *49*
4.4.3 Controlo Deliberativo *50*
4.4.4 Coordenador de Acção *51*
4.5 Conclusão 51

5 CONCRETIZAÇÃO EXPERIMENTAL 53

5.1 Caracterização do Ambiente 53
5.2 Restrições de operação 54
5.3 Definição de comportamentos reactivos 55
5.4 Percepção, representação e aprendizagem 58
5.5 Percepção e representação deliberativa 60
5.6 Utilidade e política comportamental deliberativa 66
5.7 Conclusão 68

6 RESULTADOS EXPERIMENTAIS 69

6.1 Caso experimental 1 69
6.1.1 Subsistema Reactivo *70*
6.1.2 Subsistema Adaptativo *71*
6.1.3 Subsistema Deliberativo *75*
6.2 Caso experimental 2 79

7 CONCLUSÃO 83

7.1 Trabalho futuro 84
7.1.1 Quadtree iterativa *84*
7.1.2 Aprendizagem em ambiente contínuo *85*
7.1.3 Optimizações de processamento deliberativo *85*
7.1.4 Outro mecanismo de raciocínio *86*
7.2 Considerações Finais 86

8 BIBLIOGRAFIA 89

9 ANEXOS 91

9.1 Ambiente de Desenvolvimento 91

Índice de Figuras

Figura 2.1- Organização das primitivas segundo o paradigma híbrido 8

Figura 2.2 – Organização horizontal 9

Figura 2.3 – Organização vertical..... 9

Figura 2.4 - Interacção Agente-Ambiente na aprendizagem por reforço 10

Figura 3.1 - Estrutura interna básica do ciclo de controlo do 4D/RCS (Albus, 2002)... 17

Figura 3.2 - Arquitectura 4D/RCS para um veículo individual (Albus, 2002) 19

Figura 3.3 - Nó RCS típico da arquitectura 4D/RCS 22

Figura 3.4 - Arquitectura InteRRaP (Muller, 1993) 27

Figura 4.1 - Organização geral do modelo de agente proposto 34

Figura 4.2 - Responsabilidade de cada componente do modelo proposto..... 35

Figura 4.3 - Subsistema Reactivo 37

Figura 4.4 - Organização estrutural dos componentes de Coordenação Sensorial..... 38

Figura 4.5 - Organização dos comportamentos 40

Figura 4.6 - Organização estrutural da Acção Motora 41

Figura 4.7 - Coordenar as Acções Motoras 42

Figura 4.8 - Subsistema Adaptativo 43

Figura 4.9 - Organização do Controlo Adaptativo 45

Figura 4.10 - Subsistema Deliberativo 48

Figura 4.11 - Relação entre os modelos do sistema deliberativo 49

Figura 5.1 - Total da soma de todos os vectores 56

Figura 5.2 - Comparação entre o vector da soma ponderada e da soma entre os vectores: do alvo mais próximo (a) e da soma de todos os alvos (b)..... 57

Figura 5.3 - Discretização linear em grelha..... 59

Figura 5.4 - Discretização da Acção 59

Figura 5.5 - Processo de inserção de um ponto na Quadtree 61

Figura 5.6 - Representação do ambiente discretizado utilizando a *Quadtree* 62

Figura 5.7 - Área de conhecimento do agente para obtenção da área da *Quadtree* 63

Figura 5.8 - Áreas vizinhas numa *Quadtree* 65

Figura 5.9 - Algoritmo de iteração de valor para o cálculo da utilidade 67

Figura 6.1 - Ambiente com sala e com paredes semelhante ao gridworld de Bianchi (2004) 69

Figura 6.2 - Orientação do agente para o alvo 70

Figura 6.3 - Aprendizagem Q com método exploratório *ε-greedy* 72

Figura 6.4 - Aprendizagem Q com o método exploratório baseado na heurística 72

Figura 6.5 - Aprendizagem *Dyna-Q* com exploração *ε-greedy* 73

Figura 6.6 - Aprendizagem *Dyna-Q* com exploração heurística 73

Figura 6.7 - Discretização não linear em *Quadtree* para o caso experimental 1 76

Figura 6.8 - Discretização não linear em *Quadtree* em tempo real 77

Figura 6.9 - Ambiente de teste para o caso experimental 2 80

Figura 6.10 - Situação ocorrida com a falta de exploração do ambiente 81

Figura 9.1 - Interface de visualização da PSA 92

Índice de Tabelas

Tabela 3.1 – Restrições de reacção na arquitectura 4D/RCS 25

Tabela 6.1 - Comparação entre aprendizagens e políticas de avaliação..... 73

Tabela 6.2 - Número de iterações para um determinado ε 79

"If I am walking with two other men, each of them will serve as my teacher. I will pick out the good points of the one and imitate them and the bad points of the other and correct them in myself."

Confúcio

1 Introdução

O conceito de agente inteligente híbrido surgiu durante os anos 90, numa tentativa de ultrapassar as limitações dos modelos de agente reactivos e deliberativos, de forma a conseguir integrar, num único tipo de agente, ambas as componentes, reactiva e deliberativa.

A integração de reacção e deliberação em agentes inteligentes é a característica principal dos agentes inteligentes híbridos. Estes são rápidos a reagir a perturbações do ambiente e são implementados de forma modular, mas também são dotados de mecanismos de raciocínio que são suportados por um modelo interno do ambiente. É por esse motivo que se considera que este tipo de arquitectura reúne as melhores características dos agentes reactivos e deliberativos.

Das características presentes em agentes inteligentes híbridos nomeadamente, autonomia, pró-actividade e reactividade, a reactividade está directamente relacionada com a operação em tempo real. Segundo *Murphy* (2000) qualquer agente que seja implementado segundo o paradigma reactivo é comum ser implementado tendo a noção de comportamento um papel fundamental, pois a sua concretização, no processo de implementação, favorece a decomposição, modularidade e teste incremental permitindo, assim, uma alta coesão e um baixo acoplamento entre módulos. Dado o comportamento ser composto por regras de estímulo-resposta, o tempo de reacção a alterações no ambiente é reduzido devido à baixa complexidade computacional dos comportamentos reactivos. Sendo assim, a reactividade permite uma operação em ambiente dinâmicos e com restrições de tempo-real.

A deliberação presente no agente inteligente híbrido é suportada por um modelo interno do ambiente. Por norma, o processo de deliberação é ordenado e sequencial pois inicialmente é necessário percepcionar o ambiente e construir uma representação interna, depois planear as directivas para concretizar o objectivo e, por último, colocar em prática a primeira directiva. Esse processo segue a sequência das primitivas do paradigma hierárquico (*Percepcionar, Planear, Agir*) e repete-se após a sua execução, percepcionando a consequência da acção no ambiente, replaneando e voltando a agir. Por esse motivo, a deliberação é um processo computacionalmente pesado mas permite encontrar a solução óptima caso exista.

Ao integrar processos reactivos e deliberativos, os agentes inteligentes híbridos reúnem características que favorecem a sua utilização em ambientes complexos, dinâmicos e com necessidades de operação em tempo real. É por esse motivo que, nesta dissertação, se pretende estudar formas de integrar os processos de reacção e deliberação em agentes inteligentes.

1.1 Contexto

Esta dissertação insere-se no contexto de um relatório científico desenvolvido para a obtenção do grau de Mestre em Engenharia de Redes de Comunicação e Multimédia no Instituto Superior de Engenharia de Lisboa no decorrer do ano lectivo de 2013/2014.

1.2 Motivação

A operação de agentes em ambientes complexos e dinâmicos varia segundo o tipo de agente que é utilizado para o efeito. Os agentes reactivos, dado a sua possibilidade de reagir a perturbações do ambiente, são mais utilizados para a operação em ambientes dinâmicos. Contudo, se o ambiente for complexo e caso seja necessário obter uma solução óptima, a utilização de um agente deliberativo é preferível. Este último requer uma grande capacidade computacional e a sua viabilidade pode ser comprometida dado o tempo de resposta necessário.

Do que atrás foi exposto, a motivação para esta dissertação passa por conseguir combinar os dois tipos de agente mencionados anteriormente, num agente inteligente híbrido capaz de operar, de forma satisfatória, em ambientes complexos e dinâmicos com restrições de operação em tempo real.

1.3 Objectivos

Pelo atrás mencionado, os objectivos desta dissertação são:

- Estudar algumas arquitecturas híbridas e as suas formas de integrar a componente reactiva e a componente deliberativa num agente inteligente híbrido;
- Implementar uma arquitectura híbrida que contém as principais características deste tipo de agentes;

- Adicionar uma camada intermédia para auxiliar as camadas reactiva e deliberativa através de mecanismos de aprendizagem;
- Permitir que o modelo proposto tenha a capacidade de operar em ambientes dinâmicos, lidar com recursos computacionais limitados e obedecer às restrições temporais para a operação em tempo real.

1.4 Organização do documento

A presente dissertação encontra-se organizada nos seguintes capítulos:

Capítulo 1: Introdução – O presente capítulo expõe uma introdução à temática de agentes inteligentes híbridos, a motivação para o desenvolvimento desta dissertação assim como os objectivos do seu desenvolvimento.

Capítulo 2: Enquadramento Teórico – Apresenta um enquadramento teórico dos temas mais relevantes abordados na construção de um protótipo de um agente inteligente híbrido.

Capítulo 3: Trabalho Relacionado – Apresenta a descrição de algumas arquitecturas de agentes inteligentes híbridos que foram alvo de estudo e utilizadas como apoio para o desenvolvimento de um protótipo.

Capítulo 4: Modelo de Agente Proposto – Apresenta o modelo proposto de arquitectura de agente inteligente híbrido e os aspectos base para o desenvolvimento do mesmo.

Capítulo 5: Concretização Experimental – Apresenta, de modo específico, aspectos relacionados com a integração da reactividade e deliberação no contexto do modelo proposto.

Capítulo 6: Resultados Experimentais – Capítulo onde são apresentados os testes e validações efectuados ao modelo de agente implementado.

Capítulo 7: Conclusão – Apresenta as conclusões resultantes do estudo do trabalho realizado, assim como possíveis ideias para trabalho futuro.

2 Enquadramento Teórico

No decorrer do presente capítulo, é feita uma introdução aos temas abordados nesta dissertação, dando ênfase aos pontos relevantes dos tipos de agente que estão na base do surgimento de agentes inteligentes híbridos. São abordados, também, modelos de aprendizagem e de raciocínio, nomeadamente a aprendizagem por reforço e os processos de decisão de *Markov* (PDM), que servem de base à concretização do modelo de agente proposto.

2.1 Arquitectura de Agentes Inteligentes

Num sentido geral, um agente pode ser definido como sendo uma entidade capaz de actuar autonomamente, no sentido de concretizar os objectivos para o qual foi projectado. Para isso, percebe o meio envolvente (ambiente) através de sensores e age sobre o mesmo através de actuadores. Segundo *Wooldridge* (2002), para que um agente possa ser considerado inteligente deve ter as seguintes características: autonomia, reactividade, pró-actividade e sociabilidade. Tais características permitem ao agente operar num ambiente e adaptar-se a possíveis perturbações sem a intervenção de humanos ou outros agentes (autonomia), responder atempadamente a alterações do ambiente (reactividade), tomar iniciativa de agir de forma orientada à concretização dos seus objectivos (pró-actividade) e interagir com outros agentes (sociabilidade). Adicionalmente, os agentes devem ter em consideração o efeito do tempo no sistema. Devem saber gerir os recursos disponíveis para cumprir as restrições temporais sem comprometer a satisfação da resposta (capacidade de operação em tempo-real).

Um agente pode interagir como o ambiente de modo a obter informação sobre o mesmo e criar um modelo do mundo. Essa interacção permite ao agente aprender com a experiência, através de aprendizagem por reforço, onde cada acção efectuada pelo agente resulta numa recompensa e, eventualmente, numa transição de estado. Com essa informação o agente pode criar um modelo do mundo no intuito de ganhar experiência através de simulações internas, ou utilizá-la para a geração de uma política comportamental com base em mecanismos deliberativos, nomeadamente processos de decisão de *Markov*.

Apresentam-se de seguida, de forma mais detalhada, alguns dos conceitos referidos.

2.1.1 Agentes Reactivos

Os agentes reactivos são agentes simples que se baseiam em regras *estímulo-resposta* para reagir a estímulos do ambiente. A principal característica dos agentes deste tipo é serem rápidos a reagir e por tal serem indicados para a utilização em ambientes que requerem operação em tempo real. Estes agentes, na sua forma primária, não dispõem de qualquer representação interna do mundo ou de outros agentes, as suas acções são o resultado da activação das regras ou de comportamentos pré-definidos.

Os agentes reactivos surgiram para suprir as necessidades de reacção em ambientes dinâmicos cujo tempo de reacção e a tolerância a falhas são factores em consideração para obtenção de comportamentos inteligentes.

Segundo *Murphy* (2000, p.108) todas as acções, segundo o paradigma reactivo, são realizadas por meio de comportamentos que correspondem a um mapeamento directo de percepções sensoriais a um conjunto de acções motoras utilizadas para cumprir uma tarefa. Cada comportamento é independente de outros comportamentos e realiza o acoplamento de duas componentes primitivas do paradigma reactivo, *Percepcionar* e *Agir*.

Existem duas arquitecturas, mais divulgadas na literatura, representativas de agentes reactivos: a arquitectura de *Subsunção* (Brooks, 1986) e a arquitectura de *Esquemas comportamentais* (Arkin, 1998).

A arquitectura de Subsunção é uma arquitectura que decompõe comportamentos inteligentes complexos em vários módulos de comportamentos mais "simples". É caracterizada por ter uma organização por níveis de competências, onde os níveis são constituídos por comportamentos, que por sua vez são um conjunto de módulos que definem a resposta aos estímulos provenientes dos sensores e geram acções de controlo para os actuadores para a realização de uma tarefa. Cada comportamento é uma máquina de estados aumentada (Murphy, 2000) sendo a sua operação assíncrona. Os módulos dos níveis mais altos podem inibir o comportamento gerado pelos níveis abaixo. Esta arquitectura não possui estado interno[1] porque segundo Brooks (1991, p.1) "o mundo é o seu melhor modelo", no entanto, alguns comportamentos podem ter uma representação persistente local necessária para despoletar comportamentos (Murphy, 2000, p.115).

[1] Representações persistentes do mundo

Já a arquitectura de *Esquemas Comportamentais* utiliza uma abordagem baseada em campos de potencial para especificar comportamentos, cujas respostas são representadas por vectores podendo ser combinadas através da soma vectorial para o surgimento de comportamentos emergentes. As arquitecturas baseadas em campos de potencial estão sujeitas a exibirem problemas de mínimos locais[2]. Estes podem ser ultrapassados com a utilização de alguns métodos, como por exemplo a soma de vectores aleatórios de pequena magnitude ou o método *Navigation Templates* (Slack, 1990) que permite a implementação de heurísticas simples para a criação de campos vectoriais tangenciais. Outra abordagem é a utilização de métodos mais pesados computacionalmente, como funções heurísticas.

As metodologias de campo de potencial têm algumas vantagens, como por exemplo, podem ser utilizadas numa representação contínua, permitem a combinação de comportamentos através da soma vectorial e podem ser parametrizadas fazendo variar a sua área de influência.

2.1.2 Agentes Deliberativos

Os agentes deliberativos são agentes complexos que contêm uma representação simbólica e explícita do ambiente e, eventualmente, de outros agentes. Estes agentes conseguem tomar decisões com base em raciocínio lógico baseado em manipulação simbólica. Segundo o paradigma hierárquico são compostos pelas três primitivas principais: *Percepcionar*, *Planear* e *Agir*.

Os agentes deliberativos utilizam a primitiva "Planear" para decidir ou gerar planos de alto nível para guiar as acções do agente. Dependendo do mecanismo utilizado para o raciocínio, estes agentes conseguem encontrar a solução óptima, caso exista, para um determinado objectivo, mas requerem grande processamento e tempo de resposta. Esse facto faz com que tenham dificuldades em reagir em ambientes dinâmicos, onde a mudança do ambiente é mais rápida do que o processamento do modelo interno, resultando em decisões ou acções não satisfatórias.

As arquitecturas deliberativas são em grande medida inspiradas pela psicologia humana para representar o raciocínio e por isso grande parte das arquitecturas representativas abordam conceitos psicológicos.

[2] Mínimo de uma determinada região do campo de potencial, não sendo um mínimo global.

São várias as arquitecturas que utilizam o modelo BDI proposto por *Bratman* (1987) para a organização de uma arquitectura deliberativa. Este modelo é organizado em crenças, desejos e intenções. As crenças representam a informação que o agente tem em relação ao ambiente. Os desejos correspondem aos estados do mundo pretendido e as intenções representam desejos escolhidos para concretização ou formas de concretização desses desejos.

2.1.3 Agentes Híbridos

Os agentes híbridos são formados por duas ou mais camadas, nomeadamente uma reactiva e outra deliberativa. Estes agentes combinam o melhor das duas arquitecturas descritas anteriormente, pois dispõem da reactividade fornecida pela arquitectura reactiva e a pró-actividade fornecida pela arquitectura deliberativa. Por tal, conseguem elaborar planos de alto nível ao mesmo tempo que conseguem reagir, rapidamente, a alterações do ambiente.

Este tipo de arquitectura segue o paradigma híbrido (Murphy, 2000, p.257) que corresponde às três primitivas principais mas organizadas por funcionalidade, onde inicialmente tem-se o *Planear* como principal primitiva representativa da deliberação, e o *Percepcionar* e *Agir* relacionados segundo o paradigma reactivo. A organização geral deste tipo de arquitectura é apresentada na figura seguinte.

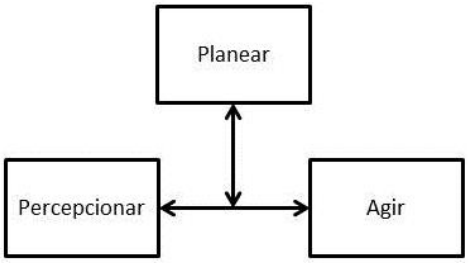


Figura 2.1- Organização das primitivas segundo o paradigma híbrido

Uma arquitectura híbrida é organizada em camadas, normalmente dispostas segundo uma organização hierárquica, onde cada camada lida com um nível diferente de abstracção. No entanto, existem duas disposições principais para as camadas das arquitecturas híbridas: horizontal e vertical.

A disposição horizontal (Figura 2.2) permite que todas as camadas possam interagir com o ambiente, ao passo que na disposição vertical (Figura 2.3) apenas a primeira camada tem essa possibilidade sendo que o fluxo da informação ocorre entre camadas de forma sequencial.

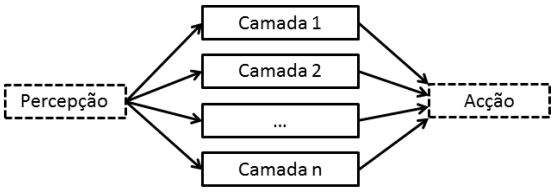


Figura 2.2 – Organização horizontal

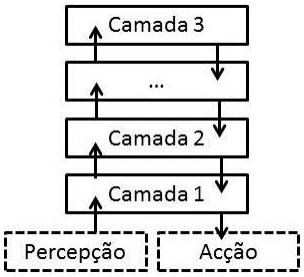


Figura 2.3 – Organização vertical

Estes dois tipos de organização têm vantagens e desvantagens. Na organização horizontal a grande vantagem é existir a possibilidade de ter várias camadas a interagir com o ambiente. No entanto, essa multiplicidade de interacções pode resultar numa falta de coerência no comportamento do agente e por isso é normal existir um "mediador" que permite seleccionar qual a camada que irá ter o controlo do agente. Mesmo assim, o mediador irá introduzir um ponto de falha do qual depende a tomada de decisão do agente. A organização vertical reduz o problema das interacções simplificando o número de interacções entre as camadas, pois o fluxo da informação passa de forma sequencial entre elas. Contudo, não é tolerante a falhas devido ao facto de que uma falha, em qualquer camada, pode comprometer o desempenho do agente.

Durante a década de 90 várias arquitecturas de agentes inteligentes híbridos foram propostas, no entanto, há duas arquitecturas que representam melhor os tipos de organização descritos anteriormente: *Touring Machine* (Ferguson, 1992) e *InteRRaP* (Muller, 1993; Muller, 1996).

A arquitectura *Touring Machine* é uma arquitectura organizada segundo o modelo horizontal composta por três camadas de controlo (reacção, planeamento e modelação) que comunicam com os subsistemas de percepção e acção. Cada camada opera de forma concorrente e independente em relação às outras, contribuindo com uma acção ou com uma instrução de comunicação segundo o seu nível de abstracção e a capacidade disponível para a concretização da tarefa. As camadas de controlo estão organizadas num estrutura de controlo que realiza a mediação entre elas para seleccionar qual deverá ter o controlo e para resolver conflitos entre as mesmas. A camada reactiva é responsável por reagir rapidamente a alterações do ambiente, para as quais as camadas acima não têm recursos disponíveis apropriados para responder. A camada de planeamento gera e executa planos para ajudar o

agente a atingir os objectivos a longo prazo. Já a camada de modelação contém uma representação simbólica do estado cognitivo do agente e de outros agentes da sociedade.

A arquitectura *InteRRaP* e outras arquitecturas híbridas irão ser estudadas detalhadamente mais à frente por se considerar terem pontos de interesse para a arquitectura proposta.

2.2 Aprendizagem por Reforço

A aprendizagem por reforço é um conjunto de técnicas utilizadas para permitir que um agente possa aprender com uma interacção directa com o ambiente de modo a maximizar um sinal numérico de reforço (Sutton *et. al.*, 1998). O princípio base deste tipo de aprendizagem é permitir que o agente aprenda quais as acções a efectuar sem saber, à partida, quais as melhores acções. A aprendizagem dá-se através do reforço (recompensa) recebido das acções seleccionadas, cujo âmbito abarca a recompensa imediata e as recompensas subsequentes. Estas duas características são descritas na literatura como exploração por tentativa e erro e recompensa diferida.

A aprendizagem por reforço distingue-se da aprendizagem supervisionada[3] por estabelecer que a aprendizagem é obtida através da interacção com o ambiente (ver Figura 2.4) de modo a atingir o objectivo. Para isso o agente deve percepcionar o estado em que se encontra, efectuar uma acção que pode resultar na alteração desse estado, obter a recompensa pela acção seleccionada e aprender a partir dessa experiência.

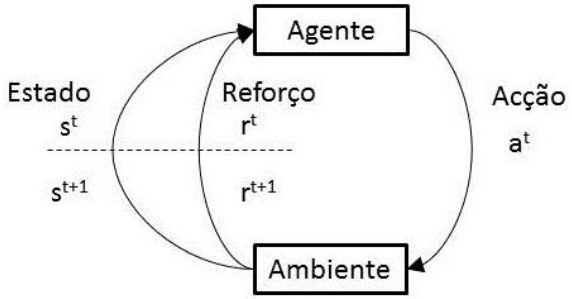


Figura 2.4 - Interacção Agente-Ambiente na aprendizagem por reforço

Existe um problema central neste tipo de aprendizagem que corresponde ao compromisso entre a exploração e o aproveitamento. Para obter o máximo de recompensa, o agente deve

[3] Aprendizagem por exemplos fornecidos por um supervisor externo detentor de conhecimento

seleccionar as acções efectuadas no passado que maximizaram a recompensa, no entanto, para obter conhecimento deve seleccionar acções que ainda não foram seleccionadas. Sendo assim, é necessário um compromisso entre aproveitar o conhecimento adquirido das experiências passadas e explorar novos caminhos para maximizar a recompensa futura. Esse equilíbrio pode ser concretizado através das políticas de selecção de acção.

Os principais elementos presentes na aprendizagem por reforço são: a política, função de recompensa, as funções valor (de estado $V(s)$ ou de estado-acção $Q(s,a)$) e, opcionalmente, o modelo. A política corresponde a uma especificação do comportamento aprendido pelo agente num determinado tempo, ou seja, um mapeamento dos estados percepcionados do ambiente em acções a serem efectuadas nesses estados. A função de recompensa define o objectivo na problemática da aprendizagem por reforço, ou seja, mapeia cada estado percepcionado num valor numérico correspondente à valorização de realizar uma acção num determinado estado. As funções valor especificam o que é bom a longo prazo. O valor do estado é a recompensa média expectável nesse estado e o valor do estado-acção é a recompensa expectável para cada par estado-acção. O último elemento, o modelo, é uma representação interna para simulação do ambiente de forma a ser possível o agente prever o estado seguinte, estando num determinado estado e efectuando uma acção.

A aprendizagem por reforço é uma das áreas da inteligência artificial mais maduras, tendo passado por três fases: a da tentativa e erro, controlo óptimo e diferença temporal. Pela exploração dos algoritmos inerentes a cada fase, *Cris Watkins* (Watkins,1989) agregou conceitos adquiridos para criar a aprendizagem Q (*Q-learning*).

A aprendizagem Q é representada pela expressão de actualização de Q(s,a) indicada.

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \, máx_{a'} \, Q(s',a') - Q(s,a)) \qquad \textbf{(1)}$$

Esta expressão explicita a forma deste mecanismo actualizar a função Q (função valor-acção) sem a necessidade de ter um modelo interno. A actualização da função Q depende apenas do seu valor actual somado com diferença entre o valor Q do estado actual e o estado seguinte obtido se efectuar a melhor acção ($máx_{a'} \, Q(s',a') - Q(s,a)$) associado a um factor de ponderação de ganhos futuros γ mais a recompensa r recebida da transição para esse estado. O valor α corresponde a um regulador de propagação da aprendizagem.

Apesar deste mecanismo de aprendizagem não necessitar de modelo, tem a desvantagem de estar limitado no conhecimento futuro, o que poderá dificultar a aprendizagem num ambiente complexo.

No contexto do modelo proposto, a aprendizagem por reforço permite obter conhecimento do ambiente com base na experiência para auxiliar a resposta reactiva do agente ao mesmo tempo que aguarda por uma resposta deliberativa.

2.3 Processos de Decisão de Markov

Um Processo de Decisão de *Markov* (PDM) é um modelo de decisão sequencial (Putterman, 2005) utilizado para modelar processos onde as transições entre estados são probabilísticas. Nele, o conjunto de acções disponíveis, as recompensas e as probabilidades de transição dependem apenas do estado e da acção actual e não dos estados ocupados e das acções escolhidas no passado (propriedade de *Markov*).

A resolução de um Processo de Decisão de *Markov* baseia-se no princípio de optimalidade de *Bellman* (1957, p.8) para seleccionar sequências de acções que permitem ao sistema actuar optimamente. Nesse sentido, consegue operar em horizontes finitos e infinitos abarcando a incerteza nas suas decisões, numa tentativa de maximizar a recompensa esperada descontada no tempo, ou seja a utilidade.

Os processos que conseguem ser modelados através dos Processos de Decisão de *Markov* designam-se de "Markovianos". Esses processos permitem a existência de diversas políticas comportamentais, que correspondem a um conjunto de acções decididas e executadas ao longo do tempo. Define-se por política óptima, a política que maximiza o retorno num determinado estado.

Para gerar uma política comportamental é necessário bastante processamento, o desafio passa por efectuar várias iterações do Processo de Decisão de *Markov* em tempo real para que seja utilizado para a formulação de uma política de suporte ao comportamento do agente.

No contexto do modelo proposto, o mecanismo de raciocínio baseado em PDM permite gerar uma resposta de mais alto nível, que leva em conta a maximização da utilidade para cada estado, para a geração de uma política comportamental do agente.

2.4 Conclusão

Neste capítulo apresentaram-se as características de agentes inteligentes híbridos e a descrição dos principais níveis de arquitectura envolvidos. Introduziu-se a noção de aprendizagem por reforço e de Processo de Decisão de *Markov* dado que estas abordagens foram escolhidas para a concretização do modelo de agente proposto. No próximo capítulo, serão abordadas em concreto duas arquitecturas de agente vocacionadas para o desenvolvimento de agentes híbridos.

3 Trabalho Relacionado

Neste capítulo são estudadas algumas arquitecturas de agente inteligente híbrido e as suas respectivas formas de integrar reacção e deliberação num agente capaz de operar em tempo real. São analisados os principais componentes de cada arquitectura e as suas respectivas formas de lidar com manipulação do conhecimento, interacção entre camadas, tomada de decisão e aprendizagem.

3.1 Real-Time Control System (RCS)

A arquitectura RCS (Albus et. al., 2002; Huang, 2011) é um modelo de referência de criação de sistemas de controlo inteligentes para a operação em tempo real que, baseada em técnicas de engenharia de *software* bem fundamentadas, permite lidar com a complexidade inerente deste tipo de sistemas.

O RCS foi inspirado no modelo teórico do cerebelo (Albus, 1975), parte do cérebro responsável pela coordenação da motricidade fina e controlo do movimento consciente, para o desenvolvimento de manipuladores[4] interactivos. No entanto, durante as últimas décadas o RCS evoluiu para uma arquitectura de controlo em tempo real para sistemas inteligentes.

O RCS estabelece um modelo de controlo hierárquico para a organização da complexidade do sistema, onde cada controlo partilha um modelo genérico. O foco desta arquitectura são controlos inteligentes capazes de se adaptar em ambientes operacionais incertos e não-estruturados[5].

Esta arquitectura também fornece uma metodologia que sugere uma abordagem específica para suportar a elaboração de um conjunto de módulos para a implementação das várias funções de um sistema inteligente.

De todas as versões que surgiram do RCS ir-se-á descrever apenas a última versão (RCS-4) por se considerar que o seu estudo é uma mais-valia para a implementação do protótipo de agente híbrido.

[4] Dispositivo mecânico para o controlo remoto de objectos
[5] Estocásticos e dinâmicos

3.1.1 4D/RCS Arquitectura de Modelo de Referência

O 4D/RCS trata-se de um modelo de referência da quarta versão do RCS (RCS-4). O 4D/RCS foi criado pela divisão de sistemas robóticos da NIST[6], no início da década de 90, para sistemas de veículos não tripulados (Albus, 2002). Consiste numa arquitectura hierárquica de multi-resolução de ciclos de controlo realimentados que integra o comportamento reactivo com funções deliberativas, constituindo assim um sistema híbrido.

A composição hierárquica do 4D/RCS segue, analogamente, a estrutura hierárquica militar onde cada nível tem um conjunto de responsabilidades e deveres. Cada nível é composto por um controlo com funções reactivas e deliberativas, que recebe objectivos e prioridades dos níveis superiores e decompõem em sub-objectivos e prioridades para os níveis subordinados. Deste modo, a reactividade, o planeamento e tomada de decisão estão distribuídos por toda a hierarquia.

O âmbito da concepção desta arquitectura foi pensado para um cenário multi-agente composto por vários veículos terrestres não tripulados e totalmente autónomos, com supervisão humana, para a formação de uma força de combate militar (Albus, 2002). No entanto, o cenário multi-agente assim como a comunicação entre eles não será abrangido no âmbito do estudo desenvolvido nesta dissertação cingindo-se ao estudo da estrutura de um único agente.

A Figura 3.1 ilustra a estrutura básica do ciclo de controlo de um nó da hierárquica RCS. A organização interna desse nó é dividida em: *Processamento Sensorial*; *Modelo do Mundo*; e *Geração de Comportamentos*.

[6] NIST - National Institute of Standards and Technology

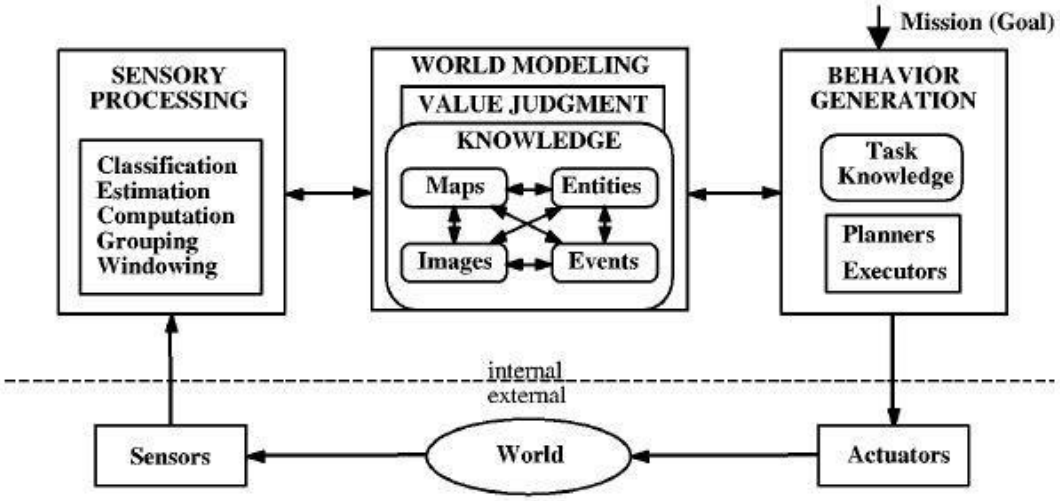


Figura 3.1 - Estrutura interna básica do ciclo de controlo do 4D/RCS (Albus, 2002)

O *Processamento Sensorial* (*Sensory Processing*) é um conjunto de processos que operam sobre o sinal dos sensores para detectar, medir e classificar informação referente a entidades e eventos recebidos dos sensores, tendo como resultado informação útil do mundo.

O *Modelo do Mundo* (*World Modelling*) corresponde a uma representação interna do mundo construída e mantida através de um conjunto de processos. As suas principais funções são:

- Manter uma base de conhecimento (imagens, mapas, entidades, eventos e relações entre eles);
- Manter uma estimação do estado do mundo para ser utilizado na previsão da resposta sensorial e no planeamento de acções futuras;
- Prever observações sensoriais baseadas na estimação do estado do mundo e também em sinais que serão utilizados pelo *Processamento Sensorial* para configurar parâmetros;
- Simular resultados de possíveis planos baseados em estimações do estado do mundo e das acções planeadas. Os resultados simulados são verificados pelo *Juízo de Valor* (*Value Judgement*) para seleccionar o melhor plano a executar.

É de realçar que o *Conhecimento* (*Knowledge*) presente na Figura 3.1 corresponde a uma estrutura de dados com informação estática e dinâmica do ambiente. Essa informação é necessária para dar suporte aos módulos *Geração de Comportamento* (*Behavior Generator*), *Processamento Sensorial* e ao *Juízo de Valor*.

A informação estática e dinâmica referida anteriormente, corresponde a uma memória de curto e longo prazo. A memória de curto prazo corresponde a uma representação simbólica do mundo que é armazenada enquanto for relevante para o foco de atenção corrente[7]. A memória de longo prazo faz uma representação simbólica permanente de todos os objectos, eventos, classes, relações e regras que são conhecidas pelo sistema inteligente.

Ainda associado ao *Modelo do Mundo*, existe o *Juízo de Valor* que corresponde a um conjunto de processos que avaliam situações percepcionadas e planeadas, determinam importâncias, atribuem prioridades, geram recompensas e punições, calculam o nível de recursos a ser alocado para cada tarefa, atribuem valores ao objectos e eventos reconhecidos, para além de activarem o módulo *Geração de Comportamento* para seleccionar objectivos e definir prioridades para os comportamentos.

O módulo *Geração de Comportamento* permite planear e controlar acções que são utilizadas para alcançar um estado desejado. Este selecciona ou gera planos utilizando o módulo *Conhecimento da Tarefa* (*Task Knowledge*) para definir o que usar, recursos utilizados e informação necessária. Juntamente com as funções do *Juízo de Valor* e informação em tempo real proveniente do *Modelo do Mundo* tem a possibilidade de encontrar os melhores recursos e escalonamento de acções para o agente.

3.1.1.1 Representação do conhecimento

Na arquitectura 4D/RCS a representação do conhecimento é mantida na Base de Conhecimento (*Knowledge*) no *Modelo do Mundo*, distribuída entre vários nós, sob forma de imagens, mapas, objectos, agentes, situações, relações, conhecimento das competências das tarefas, das leis da natureza e das relações entre elas.

A Base de Conhecimento armazena três tipos de informação: Experiência imediata, Memória de curto-prazo e Memória de longo-prazo.

A Experiência imediata consiste numa representação sensorial directa que corresponde a uma informação transiente em forma de valores actuais dos sensores. Essa informação pode corresponder a entidades, eventos, ponteiros, atributos ou variáveis de estado, ou mesmo imagens e sinais.

[7] Corresponde a um conjunto de eventos e entidades a serem preservadas temporariamente.

A Memória de curto-prazo consiste numa memória temporária que armazena uma lista de eventos e entidades participantes para processamento, análise, reconhecimento ou detecção de padrões temporais. Esta memória somente é mantida durante o foco de atenção, caso este se altere, a memória é reescrita sobrepondo-se à informação antiga.

A Memória de longo-prazo consiste num repositório de informação persistente que pode ser mantida por tempo indeterminado. Esta memória contém qualquer tipo de informação que seja considerada importante pelo *Juízo de Valor*.

A informação presente na Base de Conhecimento é utilizada para auxiliar as funções de *Geração de Comportamento* e de *Processamento Sensorial*.

3.1.1.2 Interacção entre camadas

Esta arquitectura não está dividida em camadas mas sim em níveis hierárquicos de ciclos de controlo de multi-resolução no tempo e no espaço, o que significa que cada nível pode ter mais do que um ciclo de controlo que corresponde a uma funcionalidade no seu nível de abstracção. Por exemplo, num veículo autónomo pode-se ter subsistemas para a mobilidade, comunicação e missões, sendo esses subsistemas compostos por outros subsistemas mais específicos (ver Figura 3.2).

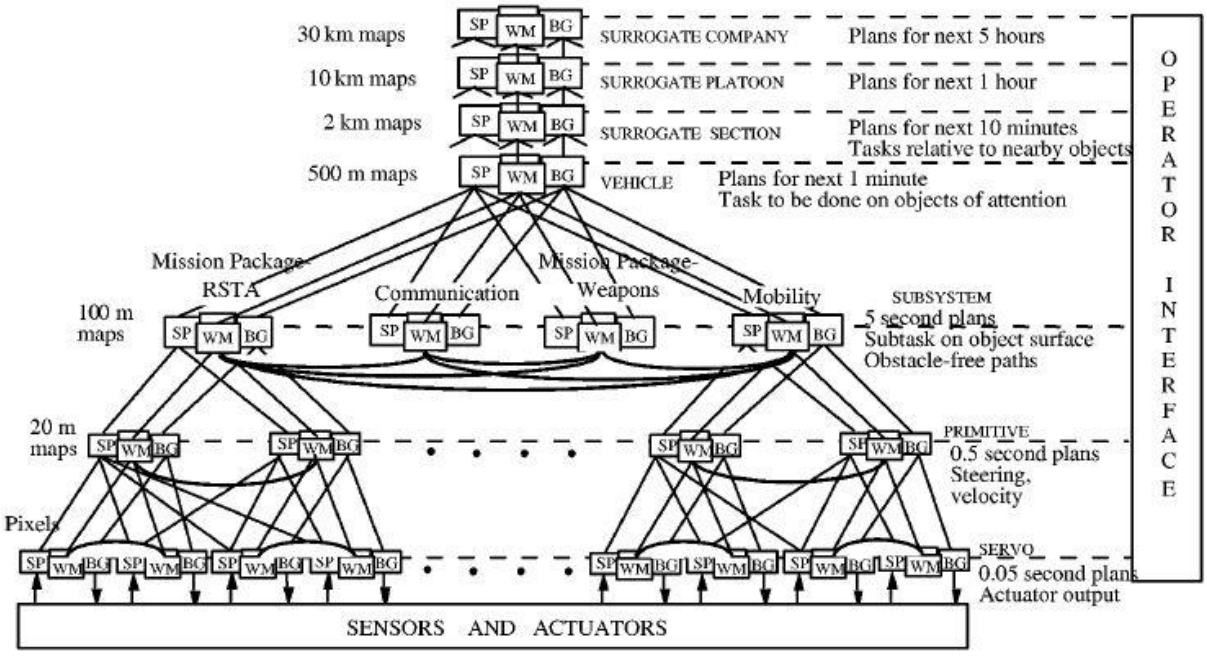


Figura 3.2 - Arquitectura 4D/RCS para um veículo individual (Albus, 2002)

Um nó da hierarquia é composto por módulos de *Processamento Sensorial*, *Modelo do Mundo* e *Geração de Comportamento*. O *Processamento Sensorial* é uma hierarquia baseada no agrupamento de sinais e pixéis em entidades e eventos. Já o módulo de *Geração de Comportamento* é uma hierarquia baseada na decomposição de tarefas e atribuição de tarefas a unidades operacionais. Estas duas hierarquias estão separadas pela hierarquia dos processos do *Modelo do Mundo* que oferece um *buffer* de comunicação entre estas duas hierarquias.

A interface de comunicação do *Modelo do Mundo* com o *Processamento Sensorial* permite comparar observações com previsões, mas para isso é necessário que as previsões estejam ao mesmo nível de abstracção e no mesmo quadro de coordenadas que as observações do *Processamento Sensorial* de cada nível. Por outro lado, a interface do *Modelo do Mundo* com o módulo *Geração de Comportamento* necessita de suportar a decomposição de tarefas e o planeamento. Sendo assim as representações do *Modelo do Mundo* devem estar na mesma gama de intervalos e na mesma resolução no tempo e no espaço das observações, e estar no mesmo sistema de coordenadas das tarefas que estão a ser decompostas em cada nível.

Para satisfazer as duas restrições mencionadas acima, o 4D/RCS permite que qualquer entidade de qualquer nível do *Processamento Sensorial* possa transformar o fluxo de informação para mapas de qualquer nível do módulo *Geração de Comportamento* e para isso, é necessário haver troca de fluxos de informação entre Modelos de Mundo de níveis diferentes.

A Figura 3.2 ilustra a hierarquia de nós para um único veículo. Pretende-se mostrar que as comunicações entre os Modelos do Mundo ocorrem entre diferentes níveis (setas verticais) ou entre Modelos do Mundo do mesmo nível (setas horizontais arqueadas). A comunicação horizontal entre os Modelos do Mundo serve para partilhar informação da Base de Conhecimento para sincronizar tarefas relacionadas.

Na Figura 3.2, pode observar-se do lado direito a presença de uma *Interface do Operador* (*Operator Interface*), essa interface permite dar acesso ao Operador aos vários níveis da hierarquia. Ainda do lado direito existe uma quantificação temporal para a execução dos planos, esse valor corresponde ao horizonte de planeamento e irá ser descrito mais adiante. Do lado esquerdo da hierarquia, encontram-se os intervalos dos mapas para cada nível do *Modelo do Mundo*.

Ao observar detalhadamente a Figura 3.2, verifica-se que o veículo tem três níveis correspondentes à secção, ao pelotão e à companhia, respectivamente. Esses níveis correspondem à cadeia de comandos de substituição e fornecem quatro importantes funções. Primeiro, providencia uma estimativa do que os seus superiores poderiam comandá-lo fazer se estivessem em comunicação directa. Em segundo lugar, permite que qualquer veículo possa assumir o posto de qualquer um dos seus superiores, caso necessário. Em terceiro lugar, fornece uma interface natural para os humanos para os níveis de secção, pelotão e companhia para interagir com o veículo no nível relevante para que a tarefa seja abordada. Por último, permite que cada veículo tenha um nó separado para lidar com cada tarefa de nível superior.

3.1.1.3 Tomada de decisão

Segundo o *Albus* (2002, p.13) o 4D/RCS disponibiliza uma plataforma, denominada de *Interface do Operador*, para que este possa ser totalmente manipulado pelos soldados, ou ser totalmente autónomo com supervisão humana. No caso da última vertente, este necessita que seja fornecido um objectivo de alto nível ao sistema (missão) para que o sistema tome decisões de forma a organizar um batalhão para cumprir os objectivos da missão. Na Figura 3.2 observa-se do lado direito da imagem a presença da *Interface do Operador*. Essa interface tem ligação com os principais constituintes do nó, representado pela linha tracejada.

A tomada de decisão na arquitectura 4D/RCS é igualmente distribuída por todos os nós da hierarquia a vários níveis de abstracção e vem sobre forma de comandos de tarefas, estes compostos por objectivos decompostos, planos e prioridades, passados pelo módulo de *Geração de Comportamento* superior ou por intervenção humana, caso seja o nível mais alto.

O processo de geração de comportamentos recebe tarefas e planos presentes nos comandos de tarefa dos níveis superiores, e executa comportamentos para cumprir determinada tarefa. A estrutura interna num processo de geração de comportamentos consiste num Planeador e num conjunto de Executores (ver Figura 3.3).

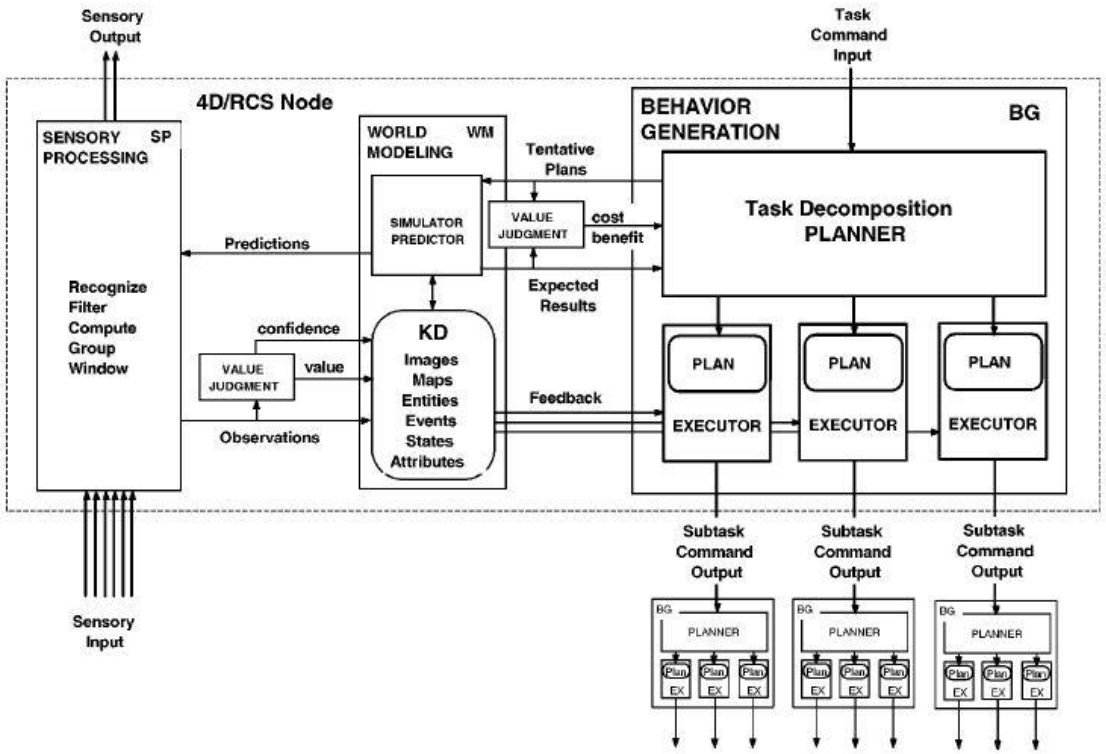


Figura 3.3 - Nó RCS típico da arquitectura 4D/RCS

Anteriormente mencionou-se a existência de um módulo de *Conhecimento da Tarefa* no *Geração de Comportamento*. Esse *Conhecimento da Tarefa* trata-se de um conjunto de competências e capacidades necessárias para executar a tarefa, e são obtidas através da aprendizagem ou definidas na implementação. Quando associado à informação presente nos comandos das tarefas, recebidos de níveis superiores, permite activar os processos de geração de comportamentos para executar tarefas.

O Planeador é o módulo responsável pela decomposição das tarefas em planos coordenados para os processos de geração de comportamentos subordinados. Este é composto por três subprocessos: Atribuidor de Trabalho (*Job Assignor*), Escalonador (*Scheduler*) e Selector de Planos (*Plan Selector*).

O Atribuidor de Trabalho é o processo que actua como supervisor da unidade de *Geração de Comportamento*. O Escalonador e o Executor formam pares correspondentes a subprocessos subordinados dentro da mesma unidade de *Geração de Comportamento*. Sendo assim, o Atribuidor de Trabalho vai supervisionar os planos dos Escalonadores para que os respectivos Executores o possam executar.

O Selector de Planos é um processo que funciona em parceria com o Simulador de Planos (*Plan Simulator*) do *Modelo do Mundo* e com o Avaliador de Planos (*Plan Evaluator*) do *Juízo de Valor* para seleccionar o melhor plano de todos para a execução no subprocesso do Executor dessa unidade de *Geração de Comportamento*. O Simulador de Planos prevê os resultados dos planos alternativos e o Avaliador de Planos processa os resultados das simulações dos planos usando uma fórmula de custo/benefício. Esses resultados são passados para o Selector de Planos para poder seleccionar o melhor plano.

Na estrutura interna do processo de geração de comportamentos, existe o Executor que é um subprocesso dentro do módulo de *Geração de Comportamento* que executa parte do plano seleccionado, coordenando acções quando necessário e corrigindo erros entre os resultados planeados e a evolução do estado do mundo reportado pelo *Modelo do Mundo*. Cada subprocesso do Executor encerra um ciclo de controlo realimentado do nó a que ele pertence.

No processo de tomada de decisão desta arquitectura, convém mencionar como é classificada a informação relevante a ser tratada. Nesse sentido, existem duas perspectivas: *top-down* e *bottom-up.*

Na perspectiva *top-down* a informação relevante é definida pelos objectivos comportamentais[8]. O sistema inteligente é guiado por objectivos e prioridades de alto nível para focar a sua atenção nos objectos especificados pela tarefa, utilizando os recursos identificados pelo *Conhecimento da Tarefa* para concluí-la com sucesso. Objectivos e percepções vindas de níveis superiores podem gerar expectativas de objectos e eventos a encontrar durante a evolução da tarefas, algo que pode ajudar a concretizar o objectivo.

Na perspectiva *bottom-up* a informação relevante surge de forma inesperada. Nesse sentido, as funções do *Processamento Sensorial*, em cada nível, detectam erros ao comparar o que era expectável com o que é observado gerando sinais de erros (normalmente ocorre nos níveis mais baixos) que são interpretados pelas regras de controlo presentes no *Geração de Comportamento* que, por sua vez, geram acções correctivas para que o processo volte a seguir o plano. Caso as regras de controlo dos níveis mais baixos não sejam capazes de corrigir a diferença então estes são filtrados para níveis superiores onde os planos são revistos e os

[8] Estado desejado que o comportamento é suposto atingir ou manter

objectivos restruturados. Nesse sentido os níveis de controlo de mais baixo nível são os primeiros a agir.

3.1.1.4 Restrições de tempo real

Esta arquitectura utiliza um mecanismo de planeamento em tempo real para conseguir cumprir as necessidades de dar resposta a eventuais alterações no ambiente, em tempos relativamente curtos.

O Planeamento do 4D/RCS está distribuído pelos vários níveis da hierarquia com diferentes horizontes de planeamento em cada nível (Figura 3.2). Essa característica permite que cada nível tenha um ponto temporal no futuro até onde planear e quanto menor o nível, menor o horizonte de planeamento, assim é possível ter um rápido planeamento num nível de detalhe inferior.

No entanto, o horizonte de planeamento não é o suficiente para suprir as necessidades de tempo real, porque num ambiente dinâmico e estocástico pode haver a necessidade de reagir a uma alteração do ambiente e o plano deixar de ser válido, sendo assim necessário replanear.

Na arquitectura 4D/RCS o replaneamento ocorre várias vezes durante um ciclo de planeamento. Consequentemente o intervalo de replaneamento é inferior ao horizonte de planeamento, assim, para se replanear a uma velocidade mais rápida do que o ciclo de planeamento, é imperativo limitar a quantidade de informação do *Modelo do Mundo* que é actualizada entre cada ciclo de planeamento e diminuir a procura necessária para a criação de novos planos.

Como referido anteriormente, é no módulo de *Geração de Comportamento* que estão agregadas a componente deliberativa e reactiva do nó. Por isso, para replanear é necessário que esse processo seja despoletado por uma percepção sensorial, identificada pela parte reactiva do módulo de *Geração de Comportamento*. Visto que a reactividade é obtida através de ciclos realimentados, assim que é percepcionada uma alteração no ambiente o despoletar do replaneamento terá uma latência associada, designada por latência de reacção.

A latência de reacção corresponde ao atraso mínimo do ciclo de realimentação reactiva de cada nível. Sendo assim, pode-se compor uma relação entre o horizonte de planeamento, o intervalo de replaneamento e a latência de reacção, mostrado na Tabela 3.1.

Tabela 3.1 – Restrições de reacção na arquitectura 4D/RCS

	Level	Planning horizon	Replan interval	Reaction latency
1	Servo	50ms	50ms	5 ms
2	Primitive	500ms	50ms	50ms
3	Subsystem	5 s	500ms	200ms
4	Vehicle	50 s	5 s	500ms
5	Section	10 min	50 s	2s
6	Platoon	Ih	5 min	5 s
7	Company	5h	30 min	10 s
8	Battalion	24h	2h	20 s

A Tabela 3.1 mostra uma diferença no tempo de resposta entre cada nível para cada tipo de intervalo. É de notar que há um desfasamento temporal entre estes intervalos. Os níveis mais baixos têm tempos curtos para conseguirem lidar com situações inesperadas. Os níveis mais altos da hierarquia têm tempos maiores para conseguirem lidar com a componente mais deliberativa, assim têm mais tempo para formular planos mais complexos.

Nesse sentido, o objectivo da hierarquização por níveis é manter a quantidade de recursos necessários para a geração de comportamentos dentro de limites controláveis.

3.2 InteRRap

Segundo *Jorg Muller* (1993, p.8) esta arquitectura foi proposta, como uma extensão ao modelo RATMAN[9], para cumprir os requisitos da modelação de uma sociedade de agentes dinâmicos. A sua principal característica é a possibilidade de combinar padrões de comportamentos com mecanismos de planeamento. Os padrões de comportamentos conferem ao agente a reactividade necessária para lidar com alterações do ambiente. Enquanto a possibilidade de formular planos confere ao agente a possibilidade de efectuar tarefas mais sofisticadas.

Segundo as duas organizações base de arquitecturas híbridas referidas anteriormente (Secção 2.1.3), esta arquitectura está organizada de forma vertical e cada camada representa níveis de

[9] RATMAN - Rational Agents Testbed for Multi Agent Networks. Plataforma de desenvolvimento de arquitecturas de agentes inteligentes baseados em lógica.

abstracção distintos que, por sua vez estão divididas em duas unidades principais: controlo e base de conhecimento.

Na Figura 3.4 apresenta-se a estrutura base da arquitectura *InteRRaP*. Nela pode-se constatar a existência de duas unidades principais em cada nível. A unidade do lado esquerdo corresponde ao controlo e a do lado direito, à base de conhecimento. A base de conhecimento contém a representação do agente e do ambiente a diferentes níveis de abstracção. Já o controlo confere ao agente a funcionalidade que cada nível está responsável, isto é, o primeiro nível está responsável por definir uma interface entre o agente e o ambiente lidando com aspectos de percepção, actuação e comunicação. O segundo nível, denomina-se por nível comportamental e é responsável pelo comportamento do agente. Segue-se o nível de planeamento que elabora e selecciona planos a serem utilizados pelo agente. Por fim, o nível cooperativo, este é responsável pelos aspectos sociais do agente.

Um aspecto importante na arquitectura *InteRRaP* é a interacção entre as camadas, esta interacção dá-se em duas passagens: uma orientada de baixo para cima (*bottom-up activation*) e outra de cima para baixo (*top-down execution*). Estas orientações previnem conflito ou problemas entre camadas uma vez que cada camada tem um nível de competência definido que caso seja extrapolado pela tarefa a ser executada, activa a camada superior, passando-lhe o controlo, para que possa tratar da tarefa não conseguida. No entanto, caso essa tarefa seja do nível de competência dessa camada então trata dessa tarefa fazendo uso das competências das camadas abaixo, caso existam. Os aspectos da interacção entre camadas serão abordados com mais detalhe posteriormente.

Outro aspecto a realçar é o facto de que cada camada presente nesta arquitectura implementa duas funcionalidades específicas: uma responsável pelo reconhecimento da situação e activação do objectivo, e outra pelo planeamento e escalonamento. A primeira mapeia a base de conhecimento, do nível em questão, e objectivos actuais para um novo conjunto de objectivos. Já a segunda, é responsável pela selecção do plano a executar baseado na informação actual sobre os planos, objectivos e bases de conhecimento do nível em questão.

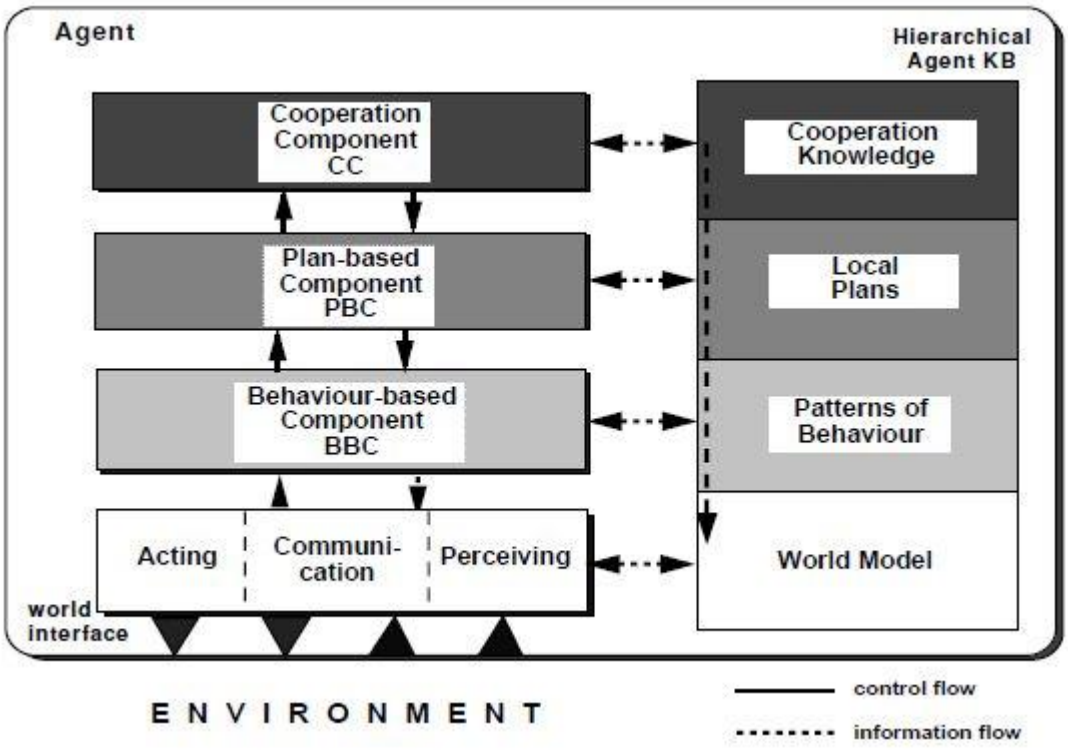


Figura 3.4 - Arquitectura InteRRaP (Muller, 1993)

3.2.1 Representação do conhecimento

O conhecimento que o agente tem do mundo encontra-se presente na arquitectura *InteRRaP* nas bases de conhecimento. Estas bases de conhecimento estão organizadas hierarquicamente. Cada camada da arquitectura contém a sua base de conhecimento correspondente.

A primeira base de conhecimento corresponde à camada mais baixa e contém as crenças do agente sobre o que acredita ser o estado actual do mundo. A segunda corresponde ao conhecimento comportamental e contém as acções primitivas e os padrões de comportamentos. O conhecimento comportamental também contém conhecimento necessário para o controlo comportamental para manter os padrões de comportamento. A terceira base de conhecimento é a de planeamento local, e contém planos locais e conhecimento específico para planear. Por fim, a quarta base de conhecimento contém conhecimento e estratégias para cooperação. Este conhecimento refere-se a um conjunto de planos utilizados para a coordenação de múltiplos agentes.

3.2.2 Interacção entre camadas

A Interacção entre camadas pode ser visualizada na Figura 3.4 através das setas não tracejadas entre as camadas. A orientação é em duas direcções, uma de baixo para cima (*bottom-up)* e outra de cima para baixo (*top-down*). No sentido *bottom-up* o fluxo da informação inicia-se na interface com o ambiente e pode (ou não) terminar na camada cooperativa. Isto porque caso a situação seja simples de resolver então as camadas abaixo da camada cooperativa resolverão.

No sentido *bottom-up*, situações mais simples, como alterações no ambiente ou algumas mensagens entre agentes, são tratadas na camada comportamental onde são activadas reacções para essas novas situações. No entanto, se a situação for complexa e necessitar de alguma deliberação, então o controlo passa para a camada acima, camada de planeamento. O fluxo de controlo parará caso a camada de planeamento consiga lidar com a situação, caso contrário, o controlo é passado para a camada de coordenação que resolverá a situação fazendo uso de planos conjuntos ou enviando mensagens de coordenação de actividade em comum a outros agentes.

A passagem de controlo entre camadas requer que a camada saiba reconhecer as suas próprias limitações para, caso não tenha competências para lidar com a situação, passar o controlo à camada acima. Segundo *Muller* (1993) este mecanismo de autoconhecimento das limitações e de passagem de controlo é designado como um mecanismo de controlo orientado à competência.

Após a informação fluir até a última camada, e esta lidar com a situação, a solução é difundida no sentido contrário (*top-down*) pela hierarquia de controlos. Nesse caso a camada cooperativa retorna um plano conjunto que é decomposto em planos únicos sincronizados, e os protocolos de negociação são decompostos em planos locais parciais. Esses planos são passados para a camada comportamental onde são interpretados para corresponderem aos padrões de comportamentos executáveis. Por fim, esses padrões de comportamento activam um conjunto de acções primitivas e capacidades de comunicação disponibilizadas pela interface com o ambiente que permite efectuar acções e enviar mensagens para o exterior.

3.2.3 Tomada de decisão

A tomada de decisão nesta arquitectura é feita em duas dimensões hierárquicas: uma descritiva e outra executiva (Muller, 1993). Na dimensão descritiva encontra-se a priorização

do objectivo que incorpora mecanismos da arquitectura de subsunção (Brooks, 1986) e que será detalhada mais adiante. Na dimensão executiva tem-se a expansão do objectivo, onde a decisão é tomada para dividir um objectivo em sub-objectivos, padrões de comportamentos executáveis e acções primárias.

Um agente com a arquitectura *InteRRaP*, contém um conjunto de objectivos a serem cumpridos. Esses objectivos podem ser acedidos por ambos os decisores da vertente descritiva e executiva para serem tratados ou decompostos. No entanto, o agente necessita escolher qual o objectivo que deverá tratar primeiro. Essa escolha é efectuada com recurso à priorização dos objectivos o qual detalhar-se-á de seguida.

3.2.3.1 Priorização do objectivo

A priorização do objectivo é usado como mecanismo de decisão para determinar qual o objectivo a seguir. Utiliza uma função de prioridade para associar uma prioridade a cada objectivo. A prioridade do objectivo leva em consideração duas componentes: uma estática e uma dinâmica.

A componente estática da priorização do objectivo define-se como prioridade estática. *Muller* segue o modelo proposto por *Maslow*[10], para atribuir uma ordem prioritária aos objectivos do agente. *Maslow* classifica a importância das necessidades humanas numa pirâmide com cinco níveis hierárquicos, e considera que as necessidades presentes no nível mais baixo da pirâmide têm maior prioridade em relação aos níveis superiores e devem ser satisfeitos antes de serem satisfeitas as necessidades humanas superiores.

Ao contrário da pirâmide de *Maslow*, a ordem atribuída por *Muller* apenas atinge o quarto nível pois considera que o último nível da pirâmide, que corresponde à criatividade, não é relevante para a arquitectura.

Os quatro níveis de priorização estática dos objectivos, em ordem decrescente de prioridade, são:

- Físico: Corresponde ao objectivos primários como, evitar colisões. Estes objectivos têm prioridade máxima.

[10] Psicólogo Americano da década de 40

- Segurança relevante: são objectivos para o agente cumprir a tarefa e resolver conflitos com outros agentes;
- Social: corresponde aos objectivos sociais como: ajuda a outros agentes ou passagem de informação;
- Optimização: corresponde aos objectivos básicos, visto que, o agente tem de reflectir sobre os seus comportamentos e como optimizá-los. Estes são os objectivos de mais baixa prioridade.

Este tipo de atribuição de prioridade é muitas vezes desejado mas poderá ter alguns problemas caso haja conflitos entre objectivos de níveis diferentes. Daí, alguma falta de flexibilidade no mecanismo de priorização para algumas situações.

É de realçar que existem dois tipos de relações entre objectivos de níveis diferentes. O primeiro é de satisfação pois o objectivo de nível superior deve ser satisfeito após a satisfação do inferior. O segundo é o de supressão pois objectivos superiores podem ter prioridades maiores num determinado contexto e é nesse sentido que entra a priorização dinâmica.

A priorização dinâmica na arquitectura *InteRRaP* é expressa em "importância relativa" que corresponde a um grau de satisfação. Ambas são inversamente proporcionais pois quando um objectivo tem uma importância relativa alta, o seu grau de satisfação é baixo.

A prioridade dinâmica é dependente do domínio do problema pois a sua formulação é dependente de critérios heurísticos como restrições no tempo, disponibilidade e escassez de recursos ou estado corrente do processamento do objectivo.

Uma decisão da escolha do objectivo é tomada quando são combinados os dois tipos de prioridades (estática e dinâmica) através de uma função de prioridade que reflecte os objectivos prioritários num determinado momento.

3.2.4 Restrições de tempo real

A arquitectura *InteRRaP* lida com as restrições de tempo real (Muller, 1993, p.53) através de um mecanismo denominado de planeamento dinâmico.

O planeamento dinâmico permite comutar entre planeamento e reacção, de modo a conseguir lidar com eventos não previstos. Uma das formas de lidar é omitindo a revisão do plano quando ocorrem alterações no ambiente. Para isso o agente necessita ter padrões de

comportamentos que saibam lidar com situações excepcionais. Outra forma é através da interacção flexível entre os módulos da unidade de controlo, que permite que os eventos não previstos sejam tratados no nível apropriado. Este mecanismo não seria possível sem a implementação de duas funcionalidades: uma responsável pelo reconhecimento da situação e activação do objectivo, e outra pelo planeamento e escalonamento.

A primeira funcionalidade é repartida em duas funções: reconhecimento da situação e activação do objectivo.

O reconhecimento da situação é um processo incremental que permite ao agente identificar necessidades de actividade. Situações críticas em termos de tempo de resposta como, por exemplo, colisão, são reconhecidas somente na camada comportamental através da informação proveniente do modelo do mundo. Se houver mais tempo disponível, essas situações podem ser melhoradas pela análise de possíveis efeitos nos objectivos do agente ou na interacção com os objectivos de outros agentes.

A activação do objectivo surge após o reconhecimento da situação e pode ser um processo rápido e simples, como na camada comportamental para a activação dos padrões de comportamentos, ou mais complexa, como na camada de planeamento local e na camada cooperativa para a activação de objectivos no processo de geração de planos e activação da negociação, respectivamente.

A segunda funcionalidade corresponde ao planeamento e escalonamento. Essa funcionalidade permite coordenar processos nas camadas vizinhas através da comunicação de obrigações. Essas obrigações seguem no sentido, entre camadas, de cima para baixo e cada camada após comunicar a obrigação aguarda pela resposta da execução da obrigação emitida. Essa resposta permite saber se a camada acima continua a processar informação ou se pára o processamento.

3.3 Conclusão

Neste capítulo, descreveu-se a estrutura e a funcionalidade de dois tipos de arquitecturas: *RCS* e *InteRRaP*. Estudaram-se conceitos e abordagens que foram úteis para a concretização do modelo proposto.

No capítulo seguinte, apresentar-se-á o modelo proposto e far-se-á referência aos principais conceitos e à abordagem utilizada no contexto das arquitecturas estudadas.

4 Modelo de Agente Proposto

Neste capítulo apresenta-se o modelo de agente proposto, referindo os principais componentes da sua arquitectura e especificando as respectivas funcionalidades, de modo a fazer-se a ponte entre os temas estudados e a definição de um modelo conceptual de agente.

4.1 Organização geral do modelo proposto

Tal como foi descrito nos capítulos anteriores, uma arquitectura de agente inteligente híbrido é organizada, normalmente, em camadas, sendo uma delas a camada reactiva que é responsável pelo fornecimento de respostas rápidas a estímulos do ambiente, e pela camada deliberativa que utiliza uma representação interna do mundo para gerar planos de mais alto nível com recurso a mecanismos de raciocínio. No entanto, nenhuma das arquitecturas estudadas referem a utilização de um processo específico para a aprendizagem ou de um subsistema onde a aprendizagem esteja disponível para utilização. Embora que, na arquitectura RCS, se refira que a aprendizagem pode surgir como um resultado da intercomunicação do *Processamento Sensorial*, *Modelo do Mundo* e *Juízo de Valor* (Albus, 2002). Nesse sentido, identificou-se que o agente necessita de se adaptar a possíveis alterações do ambiente, especialmente quando a camada reactiva não consegue lidar com a situação presente e a camada deliberativa não tem uma resposta para fornecer, dentro das restrições temporais, dado o seu tempo de processamento. Assim, surge a necessidade de ter um nível intermédio que consegue adaptar-se a perturbações do ambiente, auxiliando o nível reactivo nas suas limitações enquanto aguarda uma resposta do nível superior, nomeadamente no que se refere a ultrapassar óptimos locais.

Assim, propõe-se um modelo de agente inteligente que contém uma camada adaptativa que se encontra entre a camada reactiva e deliberativa. Esta vem trazer mais um nível de competência, que auxilia a camada reactiva nas suas limitações, com a aprendizagem, e fornece suporte à camada superior através da abstracção que utiliza.

A figura seguinte ilustra a organização geral do modelo de agente proposto.

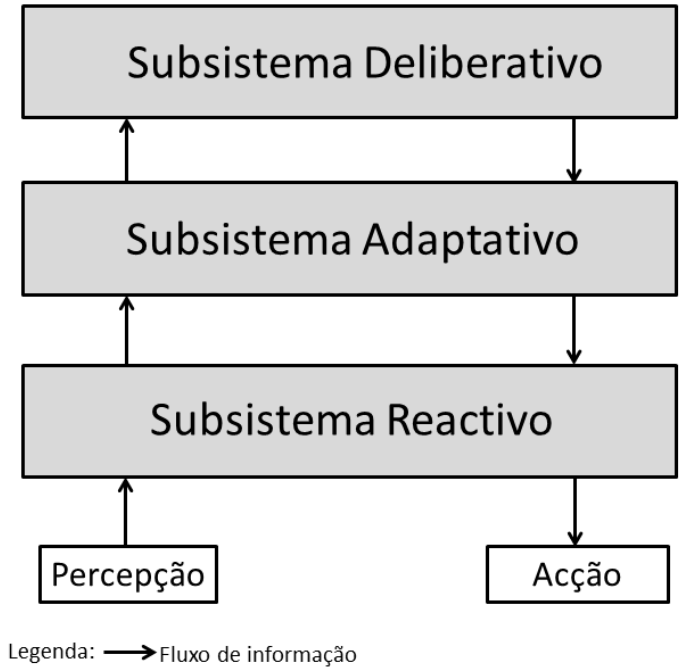


Figura 4.1 - Organização geral do modelo de agente proposto

A arquitectura ilustrada na Figura 4.1 é composta por três camadas (Reactiva, Adaptativa, e Deliberativa) organizadas em subsistemas. Cada subsistema é composto por um módulo *Coordenador Sensorial*, por um módulo *Controlo* (de cada camada) e por um módulo *Coordenador de Acção*. Para além disso, os subsistemas adaptativo e deliberativo mantêm modelos internos, formando a *Estrutura Cognitiva* de cada camada. O módulo Coordenador de Acção é partilhado pelas várias camadas como será descrito mais à afrente.

A organização em camadas desta arquitectura vai ao encontro da maioria das arquitecturas híbridas estudadas, no entanto, diferencia-se de arquitecturas como *Touring Machine* (Ferguson, 1992) e *InteRRaP* (Muller, 1993; Muller, 1996) por ter uma camada responsável pela aprendizagem.

Inspirado na arquitectura *RCS*, cada subsistema da arquitectura proposta possui um Coordenador Sensorial multi-resolução, que recebe a percepção abstraída pelo nível hierárquico abaixo e distribui essa informação para o Controlo (na camada reactiva) ou para os respectivos modelos (nas camadas adaptativa e deliberativa).

A Estrutura Cognitiva contém os modelos do mundo a cada nível de abstracção, para que possa auxiliar o processo de aprendizagem e o processo de formulação de planos. É neste

componente que se encontra o *Juízo de Valor* que é responsável por retornar os valores de recompensa ou penalização pela concretização de uma acção.

O Controlo é o componente central de cada subsistema, pois é responsável pela interacção com os restantes componentes, uma vez que utiliza a informação percepcionada pelo Coordenador Sensorial e os modelos presentes na Estrutura Cognitiva para gerar uma acção que é disponibilizada para o Coordenador de Acção.

O Coordenador de Acção é o componente responsável pela selecção da acção gerada por cada uma das camadas. O método de selecção de acção é semelhante ao da arquitectura de Subsunção proposta por *Brooks* (1986), onde as acções das camadas superiores suprimem a resposta das camadas inferiores.

A figura seguinte ilustra, de forma resumida, as principais responsabilidades de cada componente presente no modelo proposto.

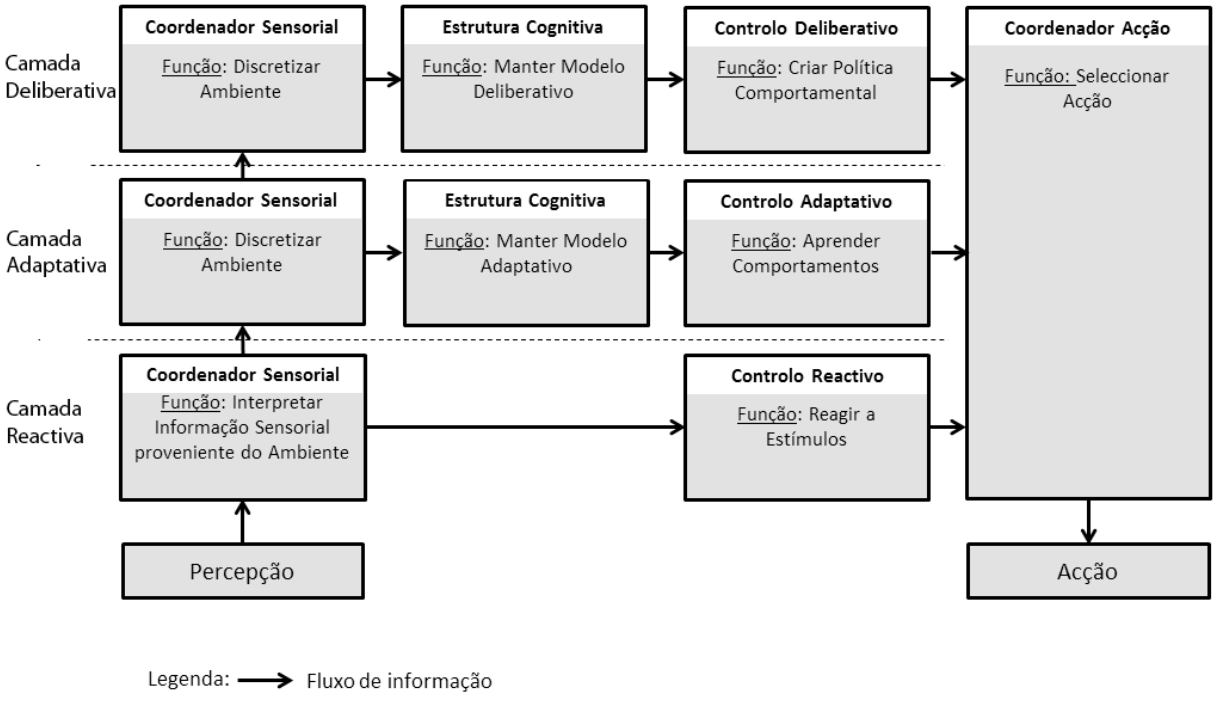


Figura 4.2 - Responsabilidade de cada componente do modelo proposto

No modelo de agente proposto ilustrado na Figura 4.2, cada camada corresponde a um nível de competência. O nível reactivo reage a estímulos, para isso, utiliza campos de potencial para representar a influência de um motivador sobre o agente, sendo que, um motivador, corresponde a um estado interno que despoleta um comportamento no agente. Já o nível

adaptativo aprende comportamentos, utilizando um processo de aprendizagem para assimilar conhecimento resultante da interacção com o ambiente. Por último, o nível deliberativo gera uma política comportamental, com base num mecanismo de raciocínio baseado em utilidade, para determinar a política comportamental para cada estado do modelo interno.

O fluxo de controlo segue a vertente sequencial, o que a diferencia de outras arquitecturas como é o caso da *InteRRaP*, onde cada camada é executada em paralelo (Lind, 2002, p.54) e utiliza a activação *bottom-up* para entregar o controlo a níveis superiores e a execução *top-down* para receber instruções de execução. Na arquitectura proposta, o fluxo de activação percorre todas as camadas, iniciando-se na camada reactiva e terminando na camada deliberativa. Após a activação de cada camada, os controlos respectivos retornam acções motoras que serão utilizadas, posteriormente, pelo Coordenador de Acção para seleccionar a acção a executar.

A manipulação do conhecimento é organizada em modelos que se encontram na Estrutura Cognitiva e armazenam informações simbólicas sobre o mundo. A camada reactiva não possui qualquer comunicação com os modelos da Estrutura Cognitiva, uma vez que lida com aspectos de âmbito exclusivamente reactivo.

O grau de autonomia do agente define como o agente interage com o ambiente e será repartido entre as três camadas. Na camada reactiva a acção será gerada com base em campos de potencial (Arkin, 1998). Nas camadas adaptativa e deliberativa a acção resultante terá a experiência passada e a utilidade em conta no processo de formulação da política comportamental.

O processo de tomada de decisão para a escolha do objectivo baseia-se no mecanismo de raciocínio escolhido e por tal, a tomada de decisão segue a política que estiver definida para o agente segundo o cálculo da utilidade. Enquanto o cálculo da utilidade está a ser processado pela camada deliberativa, o agente segue a política fornecida pela camada adaptativa. Caso ainda não tenha informação para utilizar na geração de acção da camada adaptativa, segue a acção gerada pela camada reactiva.

4.2 Subsistema Reactivo

O subsistema reactivo é responsável por fornecer uma resposta rápida aos estímulos percepcionados do ambiente. Por tal, é divido em três componentes principais, cujo objectivo é processar esse estímulo e retornar uma acção a ser utilizada pelo agente sobre o ambiente.

A figura seguinte ilustra o subsistema reactivo e os componentes presentes no mesmo.

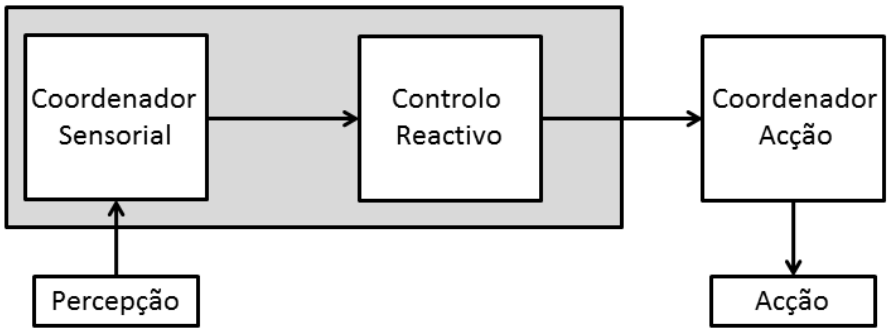


Figura 4.3 - Subsistema Reactivo

O subsistema reactivo é composto por três componentes: Coordenador Sensorial, Controlo Reactivo e Coordenador de Acção. Cada um dos componentes presentes neste subsistema insere-se no âmbito operacional de geração de acção do agente, uma vez que lida com as alterações do ambiente num contexto temporal de curto-prazo. É essa perspectiva que serve de base ao subsistema que será desenvolvido de seguida.

4.2.1 Coordenador Sensorial

O componente Coordenador Sensorial está presente em todas as camadas do modelo de agente. Cada camada tem acesso à percepção da camada inferior e processa a informação para o nível de abstracção correspondente. Assim, o Coordenador Sensorial está organizado tal como apresentado na figura seguinte.

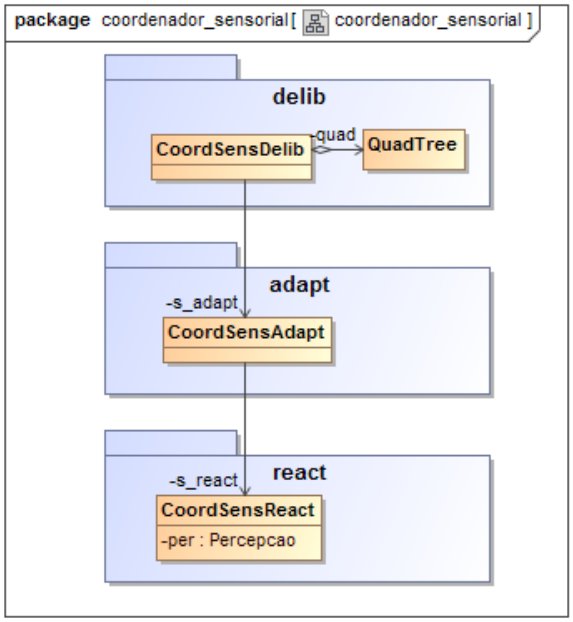


Figura 4.4 - Organização estrutural dos componentes de Coordenação Sensorial

O Coordenador Sensorial Reactivo é responsável por obter a percepção do ambiente e retirar a informação que será utilizada para gerar uma reacção a essa percepção. Este é o único componente a receber a percepção directamente do ambiente. As camadas hierarquicamente superiores utilizam a informação percepcionada por esta camada.

É no Coordenador Sensorial Reactivo que são definidos os campos de potencial que caracterizam os comportamentos reactivos de agente. O campo de potencial representa um campo de força exercido no seu espaço envolvente (atractor ou repulsor) que orienta os objectos por ele influenciados.

Assim, a percepção recebida no Coordenador Sensorial Reactivo contém a informação do ambiente e permite que o motivador seja representado como uma taxia, ou seja, um impulso orientado representado por um vector.

A percepção obtida por este subsistema permite que sejam identificadas informações relevantes da interacção do agente com o ambiente, nomeadamente a colisão com um obstáculo, a recolha de alvos e a posição actual do agente. Essa informação não é relevante para o nível reactivo mas será para os níveis superiores.

4.2.2 Controlo Reactivo

Como referido anteriormente (secção 2.1.1, p.6), os agentes reactivos são agentes simples compostos por regras de estímulo-resposta cuja característica principal é reagir a estímulos externos de forma rápida. Podem ser organizados em comportamentos que, quando combinados, resultam em comportamentos emergentes. É nesse sentido, que se concretizou este componente do subsistema reactivo.

O Controlo Reactivo é responsável por gerar uma resposta ao estímulo recebido, utilizando uma abordagem baseada em campos de potencial, para obter os vectores dos motivadores que se encontram no ambiente. Segundo *Murphy* (2000) na descrição da arquitectura baseada em esquemas comportamentais, que utiliza as metodologias de campos de potencial, um comportamento primitivo é um comportamento composto por um esquema perceptual e um esquema motor que realiza um mapeamento de um estímulo numa resposta. Já o comportamento abstracto corresponde a um comportamento que é formado por vários comportamentos combinados. No contexto da arquitectura desenvolvida nesta dissertação, os comportamentos primitivos traduzem-se por *reacções*, que são comportamentos activos pelos motivadores; e os comportamentos abstractos traduzem-se na arquitectura por *comportamentos compostos* que correspondem a agregados de reacções sendo a sua resposta resultante de uma combinação das reacções que compõem o comportamento.

Assim uma reacção tem a possibilidade de detectar um estímulo proveniente do ambiente e gerar uma resposta a esse estímulo, de modo que cada motivador presente no ambiente tenha uma resposta correspondente. Já o comportamento composto tem a possibilidade de combinar todos os comportamentos que o compõem de forma a fundi-los numa única resposta, este conceito denomina-se de fusão comportamental.

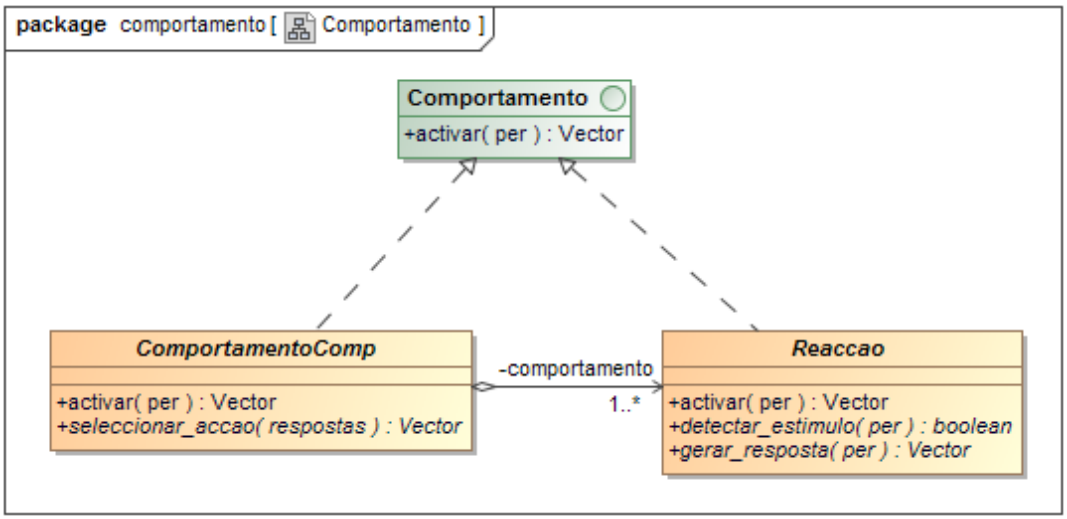


Figura 4.5 - Organização dos comportamentos

A Figura 4.5 ilustra a organização dos comportamentos na arquitectura proposta. Neste caso, verifica-se que a reacção e o comportamento composto são ambos comportamentos que retornam vectores. Verifica-se que no comportamento composto, a selecção de acção recebe as respostas provenientes das várias reacções que o compõe e retorna apenas um vector corresponde à soma vectorial dos mesmos. Este será utilizado pelo Controlo Reactivo para a geração de uma Acção Motora.

4.2.3 Coordenador de Acção

O Coordenador de Acção é um componente transversal a todos os subsistemas, uma vez que recebe acções motoras de todas as camadas. A sua grande diferença entre cada subsistema é a informação que recebe dessa camada proveniente das acções motoras, a qual irá favorecer ou não a selecção de uma determinada acção.

Para melhor se perceber, a Acção Motora é um tipo de acção criada para suprir as necessidades de selecção de acção. Nela podem-se armazenar informações que são necessárias para o processo de tomada de controlo por uma das camadas. A organização da Acção Motora está apresentada na figura seguinte.

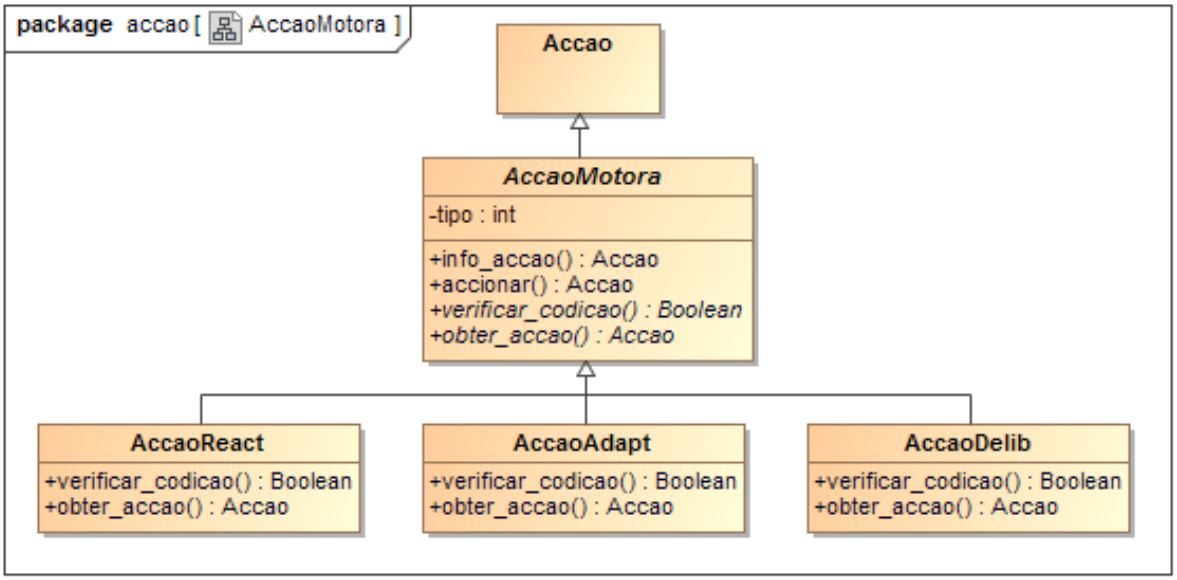


Figura 4.6 - Organização estrutural da Acção Motora

Tal como apresentado na Figura 4.6, existem três tipos de acções motoras: Reactiva, Adaptativa e Deliberativa. Uma Acção Motora é uma Acção que é activada após ser confirmada a existência de uma determinada condição, cuja validação coloca em prática a acção presente na mesma. Ou seja, cada tipo de acção motora tem a acção que foi retornada pela camada que a criou e a sua utilização depende da condição de activação e da sua posição na hierarquia.

Para o subsistema reactivo a condição verificada é a existência de motivadores no ambiente, visto que, em cada ciclo é gerado uma acção que orienta o agente para o motivador, então significa que a acção reactiva existe sempre, desde que existam motivadores no ambiente. Portanto, o agente mesmo não tendo qualquer acção disponível das camadas hierarquicamente superiores, terá a acção deste subsistema para efectuar. No caso da não existência de motivadores no ambiente, nenhuma acção é retornada.

As condições de activação para as restantes camadas serão abordadas nos seus respectivos subsistemas.

O processo de coordenar as acções provenientes das várias camadas é realizado pelo coordenador de acção tal como indicado na figura seguinte.

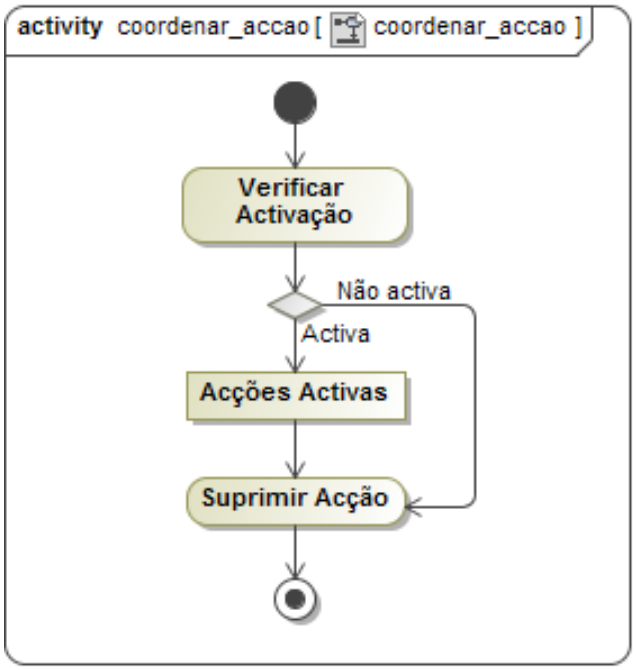


Figura 4.7 - Coordenar as Acções Motoras

A Figura 4.7 ilustra o processo de coordenação de acção. Por cada Acção Motora presente no Coordenador de Acção é verificado a existência da sua condição de activação. Caso essa condição exista então é adicionada a Acção Motora ao conjunto de acções activas que serão utilizadas no processo de supressão de acção para a escolha de uma acção a ser efectuada.

No processo de verificação de activação da camada, são verificadas duas condições: a activação segundo a parametrização utilizada e a condição de activação de cada camada. Toda a parametrização referida no decorrer desta dissertação encontra-se concretizada num ficheiro de configuração. Já o processo de supressão leva em conta o nível hierárquico da Acção Motora e os valores nela presentes para a escolha dessa acção, por exemplo, valor Q ou política.

Deste modo, o Coordenador de Acção gere as acções fornecidas pelas camadas considerando a hierarquização adoptada e as condições de selecção.

4.3 Subsistema Adaptativo

O Subsistema Adaptativo é responsável por aprender a partir da interacção com o ambiente e retornar uma acção que leva em conta a aprendizagem realizada. Para que o processo de aprendizagem ocorra, o agente necessita de manter uma representação interna do conhecimento e, por esse motivo, contém mais um componente em comparação com o subsistema anterior, a Estrutura Cognitiva. A organização deste subsistema encontra-se ilustrada na figura seguinte.

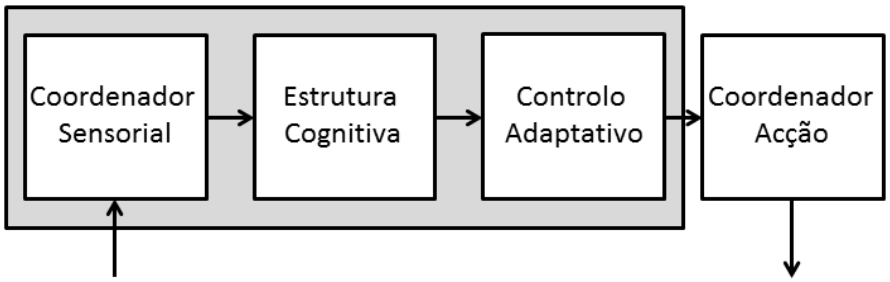


Figura 4.8 - Subsistema Adaptativo

O Subsistema Adaptativo encontra-se no âmbito táctico de geração de acção, uma vez que a acção retornada por esta camada leva em conta a informação aprendida sobre o ambiente que é útil para a utilização a médio-prazo.

4.3.1 Coordenador Sensorial

O Coordenador Sensorial Adaptativo é responsável por discretizar o ambiente para a obtenção de uma representação do mundo em estados abstractos. Estes dividem o ambiente uniformemente numa representação em duas dimensões, diminuindo a complexidade espacial.

A representação estrutural do Coordenador Sensorial ilustrada na Figura 4.4, denota a utilização da percepção presente na camada reactiva para a geração de um nível superior de abstracção do ambiente na camada adaptativa. Esta abstracção facilita a assimilação do conhecimento proveniente da exploração do mesmo.

A alteração da representação espacial para uma representação discreta, gera uma necessidade, identificada no processo de aprendizagem, de abstrair também as acções possíveis do agente. Tanto o processo de discretização do estado como o da acção serão abordados mais adiante no capítulo de concretização experimental.

4.3.2 Estrutura Cognitiva

A Estrutura Cognitiva corresponde à base de conhecimento da arquitectura proposta. É neste componente que estão armazenados os modelos utilizados para o processo de aprendizagem e mecanismo de raciocínio. Este contém um Juízo de Valor que, à semelhança da arquitectura RCS estudada, indica o valor de recompensa pelas transições entre cada estado.

No subsistema adaptativo a Estrutura Cognitiva contém dois modelos que auxiliam os processos de aprendizagem disponíveis: Modelo Aprendizagem, e Modelo *Dyna*.

O Modelo de Aprendizagem corresponde a uma estrutura de dados utilizada para armazenar os valores Q que são utilizados durante o processo de aprendizagem. Estes valores correspondem ao valor expectável para um determinado estado Q (correspondente ao par estado-acção). É neste modelo que é efectuada a actualização do valor Q para os estados que o agente passa a conhecer quando efectua uma determinada acção (actualização da função *Q(s,a)* apresentada na secção 2.2).

O Modelo de Aprendizagem tem um mecanismo capaz de lidar com complexidade espacial, uma que vez que possui um mecanismo de esquecimento que realça o carácter táctico deste subsistema, removendo valores Q que se encontrem há mais tempo armazenados na estrutura de dados. Este processo remove da estrutura de dados os valores *Q(s,a)* de estados que foram explorados há mais tempo. Para ser considerado informação "antiga" é iterado, a cada passo do agente, o valor de memorização onde é reduzido até um determinado limiar que quando atingido, remove da estrutura de dados o valor *Q(s,a)* para esse estado "antigo". No entanto, este processo apenas remove os valores de *Q(s,a)* que não correspondem a colisões, o que significa que o agente mantém, de forma persistente, a informação das colisões.

O Modelo *Dyna* corresponde ao modelo interno utilizado pela aprendizagem *Dyna*-Q, pois este processo de aprendizagem necessita de um modelo interno para que possa iterar internamente a informação aprendida de modo a convergir muito mais rapidamente do que a aprendizagem Q. Este modelo é composto por uma função de transição e uma função de recompensa que armazenam informações úteis para uma representação interna do mundo.

É de realçar que durante o processo de iteração efectuado pela aprendizagem *Dyna*-Q o mecanismo de esquecimento não é activado, pelo que as iterações efectuadas por este processo não contam para o processo de esquecimento da informação.

Na Estrutura Cognitiva existe um componente denominado de Juízo de Valor que é responsável pelo processo de atribuição de reforço (recompensa pela acção) para o movimento do agente. Ao Juízo de Valor está associado um custo de movimentação que corresponde a uma penalização por qualquer acção efectuada. Ao custo de movimentação é somado uma penalização pelo afastamento do agente ao alvo.

Deste modo, ambas as atribuições de valores pela transição compõem a geração de reforço para o agente desenvolvido nesta dissertação.

4.3.3 Controlo Adaptativo

O Controlo Adaptativo é o componente responsável por conferir ao agente a capacidade de aprender a partir da interacção com o ambiente. A aprendizagem decorre da atribuição de um valor de utilidade a cada par estado-acção. Esse valor, correspondente ao valor de retorno expectável. É através da exploração que o agente vai conhecendo o ambiente e actualizando os modelos, de modo a serem utilizados por outros processos, nomeadamente a aprendizagem *Dyna*-Q e os Processos de Decisão de *Markov* (camada deliberativa).

A figura seguinte ilustra a organização estrutural deste componente.

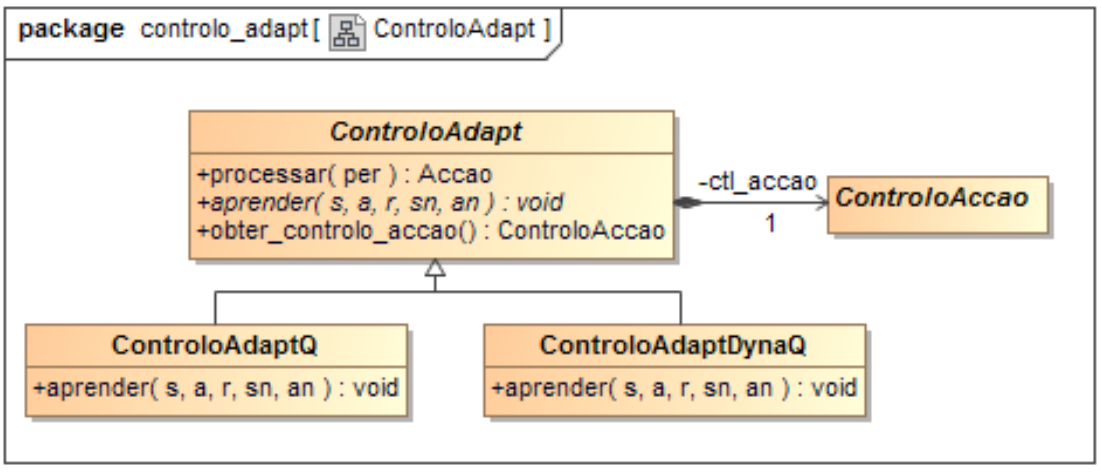


Figura 4.9 - Organização do Controlo Adaptativo

Na Figura 4.9 verifica-se que na organização estrutural do Controlo adaptativo existem dois processos de aprendizagem: a aprendizagem Q e a aprendizagem *Dyna*-Q. Também verifica-se a existência de um Controlo de Acção, esse controlo corresponde a uma generalização das políticas de selecção de acção presentes na arquitectura.

Tal como apresentado, o processo de aprendizagem presente no Controlo Adaptativo pode ser de dois tipos: Aprendizagem Q e Aprendizagem *Dyna*-Q. Ambas as aprendizagens implementadas para este trabalho são aprendizagens *off-policy*, o que significa que existe uma separação entre as políticas que são utilizadas para aproveitar o conhecimento obtido (política comportamental) e para explorar o ambiente (política de selecção de acção). Assim, o processo de aproveitamento passa pela actualização da função *Q(s,a)* para o estado anterior (ver expressão na Secção 2.2). Isto significa que a propagação de valor entre estados corresponde sempre ao valor da melhor acção possível. Já o processo de exploração pode utilizar outra política para explorar o ambiente, como por exemplo, a escolha da melhor acção conhecida até ao momento para o estado actual com uma certa probabilidade de escolha entre essa acção e outra aleatória.

Apesar das semelhanças, as duas aprendizagens mencionadas são diferentes no que toca à utilização de um modelo interno do mundo pois, ao contrário da aprendizagem Q, a aprendizagem *Dyna*-Q utiliza um modelo interno para efectuar simulações do cálculo do valor Q, resultando, por isso, numa convergência mais rápida.

Na Figura 4.9 está representado um Controlo de Acção agregado ao Controlo Adaptativo. Esse Controlo corresponde aos tipos de políticas de selecção de acção (método exploratório) possíveis para esta arquitectura: a política *ε-greedy* e a política baseada na heurística.

A política *ε-greedy* caracteriza-se por ser uma política que aproveita a informação aprendida com uma probabilidade de $1 - \varepsilon$, onde o ε corresponde ao valor da probabilidade utilizada para a exploração do ambiente. Deste modo, é possível utilizar o conhecimento adquirido com a aprendizagem mas também explorar o ambiente para conhecer o resultado de novas acções possíveis.

A política baseada na heurística é uma política caracterizada por ter uma exploração "guiada", o que significa que a selecção de acção é restringida às acções mais próximas, de um valor, que ainda não foram seleccionadas. Esse valor corresponde à heurística, e refere-se à orientação da acção obtida da camada reactiva, sendo definida antes do fluxo de controlo chegar ao Controlo Adaptativo. Para que a selecção de acção se concretize, é verificado o valor Q para cada acção possível no estado actual. O valor Q, nesta política de selecção de acção, é utilizada para identificar as acções que não têm valor Q associado, permitindo que essas possam ser seleccionadas consoante a sua aproximação à heurística. Se todas as acções

possíveis já tiverem um valor Q associado, então é utilizada uma política *greedy* para a escolha da acção, ou seja, determinada pelo maior valor Q.

Após todo este processo, é criada uma Acção Motora para ser retornada para o Coordenador de Acção deste subsistema. Visto que o processo funde-se com o âmbito do Coordenador de Acção, optou-se por especificar este processo na Secção abaixo.

4.3.4 Coordenador de Acção

O Coordenador de Acção, como referido na Secção 4.2.3, é responsável por seleccionar uma acção de todas as acções recebidas através do processo de supressão de acção. No entanto, no âmbito da estrutura desta dissertação optou-se por explicar os aspectos específicos de cada subsistema em separado.

Para a geração da Acção Motora no Controlo Adaptativo é verificado se o agente conhece esse motivador. Para saber se conhece, é consultado o Modelo de Aprendizagem e verificado quais os valores Q presentes na estrutura de dados que correspondem ao estado actual do agente. No caso de existirem valores Q para o estado actual do agente, então é adicionado a indicação de que a acção motora deve estar activa. Caso contrário, a acção encontra-se desactiva.

Assim que o Coordenador de Acção recebe a Acção Motora deste subsistema, verifica durante o processo de supressão se a acção retornada por este contém valores Q para o estado actual do agente, através da confirmação de activação da acção. Caso não existam valores Q, significa que o agente não conhece este estado e por tal não considera esta acção para ser efectuada. No entanto, caso tenha informação então essa acção sobrepõe-se à acção reactiva.

Com este processo, o Coordenador de Acção consegue garantir que o comportamento do agente leva em consideração o conhecimento adquirido para esse estado.

4.4 Subsistema Deliberativo

O último subsistema desta arquitectura corresponde ao Subsistema Deliberativo, sendo o mais alto da hierarquia do agente inteligente híbrido. Este é responsável por definir uma política comportamental baseada em Processos de Decisão de *Markov* (PDM). A organização deste subsistema assemelha-se à organização do subsistema adaptativo, uma vez que utiliza um

mecanismo de raciocínio que necessita da representação interna do mundo. Daí verificar-se, novamente, a presença da Estrutura Cognitiva.

Segue-se, na figura seguinte, a ilustração da organização conceptual deste subsistema.

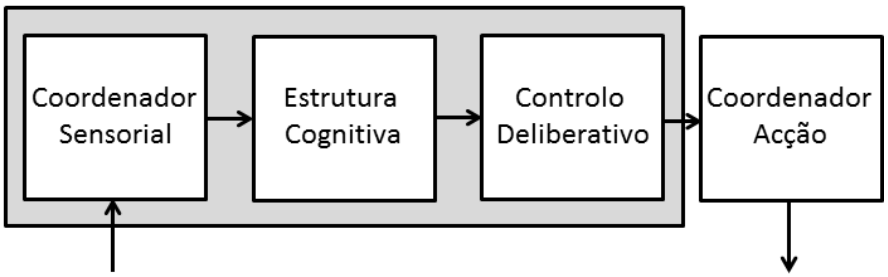


Figura 4.10 - Subsistema Deliberativo

Tal como representado na Figura 4.10, o subsistema deliberativo recebe uma percepção da camada abaixo e manipula-a para o nível de abstracção que lhe compete para, posteriormente, ser utilizada para gerar uma política comportamental de âmbito estratégico, uma vez que as acções retornadas levam em conta a concretização dos objectivos a longo-prazo. É nesta perspectiva que se descreve o subsistema a seguir.

4.4.1 Coordenador Sensorial

O Coordenador Sensorial Deliberativo é um componente que recebe uma percepção fornecida pelo Coordenador Sensorial da camada abaixo e utiliza essa informação para abstrair o ambiente anteriormente discretizado pelo nível inferior. Os estados abstractos resultantes formam áreas de dimensão variável, sendo discretizado em função dos óptimos locais presentes no ambiente segundo uma decomposição em *Quadtree*. Resultando numa disposição não linear das áreas ao longo do ambiente.

O termo *Quadtree* refere-se a um tipo de estrutura de dados hierárquica cuja propriedade comum é baseada na decomposição recursiva do espaço (Samet, 1984). Normalmente, ela é utilizada para subdividir recursivamente um espaço de duas dimensões em quatro quadrantes ou regiões. A sua representação é em árvore onde cada nó da árvore contém quatro nós filhos, que podem conter outros quatro nós filhos ou permanecerem vazios, dependendo da identificação desse nó como nó folha, ou seja, nós que se encontram no extremo da ramificação da árvore.

A representação baseada neste conceito de *Quadtree* foi utilizada para discretizar o ambiente num nível de abstracção superior, onde cada região (denominada de área) pode ter dimensões diferentes segundo um conjunto de pontos relevantes considerados para o cálculo, os quais correspondem aos objectivos e aos óptimos locais. Deste modo, os pontos relevantes correspondem a pontos de interesse e por tal, pretende-se dar maior "foco de atenção" à área que contém o ponto relevante, daí quanto mais próximo desse ponto maior o nível de detalhe e por tal menor será a dimensão da área. Assim, o cálculo da *Quadtree* para a obtenção desse resultado corresponde a dividir sucessivamente as áreas até que a resolução máxima seja atingida (Restrição 5 da Secção 4.3).

4.4.2 Estrutura Cognitiva

A Estrutura Cognitiva do subsistema deliberativo é responsável por armazenar os modelos a serem utilizados neste subsistema. Nesse sentido, possui modelos internos que armazenam informações sobre o mundo e que auxiliam o cálculo dos Processos de Decisão de *Markov* (PDM). Deste modo, existem os seguintes modelos neste componente:

- Modelo Representativo
- Modelo Deliberativo
- Modelo PDM

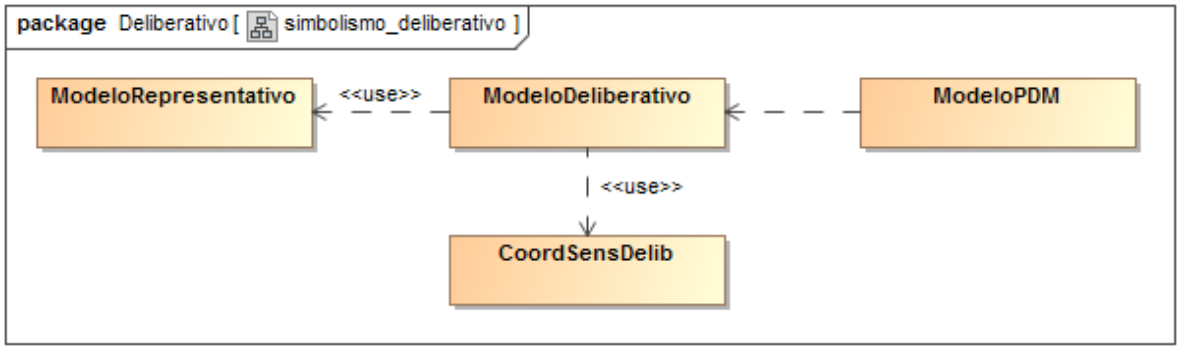


Figura 4.11 - Relação entre os modelos do sistema deliberativo

A Figura 4.11 ilustra a relação entre os modelos presentes no subsistema deliberativo.

O agente, através da exploração do ambiente, gerada pelas camadas reactiva e adaptativa, começa a representar o espaço explorado e a informação associada a cada estado no Modelo Representativo. Esse modelo faz uma representação do ambiente com base na informação que ele vai adquirindo. As principais funções deste modelo são: 1) armazenar a informação que o

agente tem do ambiente (que pode corresponder à realidade ou não) e 2) suportar o processo de discretização pela *Quadtree*.

É no Modelo Deliberativo que, após a exploração do agente, é utilizada a representação do ambiente do Modelo Representativo e a decomposição em *Quadtree* fornecida pelo Coordenador Sensorial Deliberativo para obter uma representação abstracta de mais alto nível, baseada em estados abstractos dispersos não linearmente no espaço. A discretização presente neste modelo é recalculada para cada óptimo local encontrado no ambiente.

Por último, para encontrar uma política comportamental para a representação abstracta do ambiente do Modelo Deliberativo, é utilizado o Modelo PDM, que representa o ambiente segundo um processo de decisão de *Markov*, decomposto num conjunto de estados, acções, modelo de probabilidade de transição e recompensas, para o cálculo da utilidade e o posterior cálculo da política.

4.4.3 Controlo Deliberativo

O Controlo Deliberativo é o componente que utiliza as funcionalidades dos componentes presentes neste subsistema para a geração de uma política comportamental baseada em Processos de Decisão de *Markov* (PDM). Como referido na secção 2.3, o objectivo de um PDM é maximizar a utilidade da acção a longo-prazo. Essa característica reforça o carácter estratégico deste subsistema pois a política obtida corresponderá à política com a maior utilidade expectável.

O Controlo Deliberativo é activado quando existe informação sobre o ambiente. Essa informação é recebida à medida que são encontradas as posições dos óptimos locais presentes no ambiente. Por cada nova posição encontrada, a informação é utilizada para criar a representação abstracta do ambiente segundo a discretização em *Quadtree*, compondo o Modelo Deliberativo. Este modelo serve de base para a geração do Modelo PDM, pois é através deste que é retirada a informação para organizá-lo segundo um Processo de Decisão de *Markov*. Uma vez criado, calcula-se a utilidade para todos os estados presentes no modelo, utilizando o processo de iteração de valor para o cálculo da política óptima.

Após todo este processo e à semelhança da descrição efectuada na secção anterior, gera-se a Acção Motora. Processo este que será explicado na secção seguinte.

4.4.4 Coordenador de Acção

A geração da Acção Motora deliberativa é iniciada no Controlo Deliberativo que, após ter uma política disponível, activa a Acção Motora e adiciona a melhor acção para o estado onde o agente se encontra. A atribuição de acção somente ocorre após a convergência da utilidade. Caso ainda não seja possível obter uma política, então nenhuma acção é atribuída à Acção Motora e o seu estado mantém-se inactivo.

O valor da utilidade é adicionado à Acção Motora para auxiliar o processo de selecção de acção. Ela é utilizada para confirmar se o valor de utilidade para um estado existe. Uma vez que é possível o agente ter um valor de utilidade para um estado mas a acção por ela encontrada não corresponder à melhor acção possível.

Para a acção deliberativa ser seleccionada, o valor da utilidade desse estado deve ser superior a zero. Caso contrário, essa acção é descartada e o agente é controlado pelas acções das camadas abaixo.

Assim, com o mecanismo adoptado, o Coordenador de Acção escolhe as acções superiores na hierarquia de agente híbrido, desde que respeitem certas condições estipuladas para cada camada.

4.5 Conclusão

Neste capítulo apresentou-se o modelo de agente proposto, apresentando a organização geral do modelo e a arquitectura dos principais subsistemas, fazendo referência aos principais conceitos abordados. Segue-se a concretização do modelo de agente proposto, fazendo a ponte entre os conceitos mencionados neste capítulo e a abordagem escolhida para a implementação do protótipo.

5 Concretização Experimental

Neste capítulo apresentam-se os aspectos específicos relacionados com a implementação do modelo de agente proposto descrito no capítulo anterior, mencionando-se as características do ambiente e as restrições impostas para o bom funcionamento do protótipo.

5.1 Caracterização do Ambiente

No contexto do modelo de agente proposto, o ambiente tem um papel fundamental no processo de aquisição de conhecimento, pois é a partir da interacção entre o agente e o ambiente, que o agente adquire conhecimento do meio envolvente. Deste modo, pretende-se classificar o ambiente segundo as características mencionadas por *Russell* e *Norvig* (2010, p.42-45) adaptadas para o contexto de utilização, de modo a mostrar as opções de implementação para a lidar com ambientes cujas características correspondem às mesmas de uma operação em tempo real. Listam-se de seguida essas características e as opções de implementação:

Parcialmente observável – O agente é colocado num ambiente o qual vai conhecendo à medida que vai explorando, apenas obtendo informação próxima da posição onde se encontra, por tal, os sensores do agente não detectam todos os aspectos relevantes do ambiente para a tomada de decisão. Assim, o agente necessita de manter o estado interno do mundo que se reflecte nos modelos presentes na Estrutura Cognitiva;

Não-Determinístico – O próximo estado do agente não é determinado pelo estado actual do agente e pela selecção da acção efectuada. Contudo, realça-se o facto de, na camada deliberativa, o agente obter a estrutura do ambiente indirectamente através da identificação dos óptimos locais, pois é com base nessa informação que é efectuada a discretização não linear em *Quadtree*. Assim, o grau de incerteza entre a transição de estados é considerado pelo mecanismo de PDM através da representação da probabilidade de transição entre áreas adjacentes na orientação da acção efectuada;

Sequencial – A decisão corrente afecta todas as decisões futuras. Na arquitectura proposta esta característica é mais visível durante a formação do modelo PDM, isto porque quando o agente está a seguir a acção da camada reactiva existe a possibilidade de atingir óptimos

locais, e essa situação despoleta uma nova discretização do ambiente na camada deliberativa e consequentemente o agente irá voltar a calcular a política para a nova discretização, descartando o processamento anteriormente realizado;

Dinâmico – O ambiente pode-se alterar enquanto o agente está a deliberar. No entanto, esta característica cai numa das restrições presentes neste documento, uma vez que a detecção de alterações no ambiente por parte do agente, quando se encontra perante um ambiente dinâmico, resulta na necessidade de alteração dos modelos deliberativos;

Contínuo – O ambiente é contínuo, sendo as respectivas dimensões representadas no domínio real. O agente lida com esta situação abstraindo, através da discretização, os estados do ambiente. Assim, todas as camadas abstraem, à excepção da camada reactiva que não abstrai e retorna acções no domínio contínuo;

Conhecido – Esta característica, segundo *Russell* e *Norvig* (2010, p.44), está mais relacionada com o estado do conhecimento do agente sobre as leis do ambiente do que propriamente com o ambiente. Assim, o agente conhece quais as acções que pode fazer e quais as recompensas por efectuar certa acção. Por exemplo, o agente sabe que se colidir com o obstáculo a recompensa é negativa e se apanhar um alvo é positiva.

5.2 Restrições de operação

A concretização do modelo proposto utiliza uma plataforma de simulação de agentes (PSA – Ver anexo) para simular a operação de um agente inteligente híbrido, quando presente num ambiente com as características referidas na secção anterior. No entanto, o sistema implementado necessita obedecer a um conjunto de restrições de modo a poder cumprir os objectivos propostos. As restrições consideradas em termos de cenários experimentais são as seguintes:

1. O tamanho do passo do agente deve corresponder ao valor máximo da largura de um estado discreto na grelha linear, o que para a implementação adoptada corresponde ao valor 1. Essa restrição garante que o agente não entra em ciclos infinitos quando a acção dependa da camada adaptativa, porque para o método exploratório baseado na heurística, o agente selecciona a acção que se encontra mais próxima da heurística definida. Assim, a selecção de acção pode seleccionar duas acções diferentes

(contrárias) para o mesmo estado caso o tamanho do passo não permita que o agente aprenda o resultado da acção efectuada;

2. A informação dos objectivos deve ser disponibilizada ao agente. Essa informação irá garantir que, na camada reactiva, o agente obtenha um vector para todos os alvos presentes no ambiente e que, na camada deliberativa, se possa criar o modelo PDM uma vez que é necessário o fornecimento dessa informação por parâmetro;
3. A acção deverá ser discretizada em 4 direcções (Cima, Baixo, Esquerda e Direita). Esta restrição garante que o agente aprenda a transitar correctamente entre estados discretos porque, sendo um estado discreto representado por um quadrado e sendo a acção efectuada na camada reactiva contínua, é pouco provável que uma acção na diagonal transite, na grande maioria das vezes, para o estado que se encontra na diagonal do estado actual;
4. O dinamismo do ambiente leva a que o agente, quando detecta alterações no ambiente, inviabilize o uso dos modelos deliberativos, impossibilitando a geração de planos de mais alto nível, devido à falta de estabilidade dos alvos para a convergência da política. Assim o agente passa a ser guiado somente pelas camadas reactiva e adaptativa.
5. A resolução máxima de decomposição da *Quadtree* deve ser sempre igual ao valor mínimo de um estado discreto. O objectivo desta restrição é evitar incompatibilidade de aplicação do mecanismo de discretização de alto nível;
6. Um elemento (alvo ou obstáculo) deve preencher o valor mínimo de um estado discreto. Esta restrição irá garantir a possibilidade de identificar o tipo de elemento que se encontra no estado discreto, impossibilitando a existência de dois elementos num único estado;
7. A colisão não é evitada. Esta restrição utiliza uma das características da plataforma de simulação de agentes (PSA) para identificar uma colisão. Assim, utiliza essa informação como condição de aprendizagem na camada adaptativa.

5.3 Definição de comportamentos reactivos

Um comportamento, como referido anteriormente, surge como uma resposta do agente aos estímulos do ambiente de modo a concluir uma tarefa. O agente é estimulado pelos motivadores presentes no ambiente que exercem uma força atractora sobre o agente. Os

motivadores, no contexto da implementação adoptada, correspondem aos alvos presentes no ambiente.

O agente ao navegar pelo ambiente encontra-se sobre a influência dos alvos em qualquer posição do ambiente. Para que tal ocorra, a área de influência de cada alvo é superior ao diâmetro máximo do ambiente. A distância que o agente se encontra do alvo traduz-se na influência que o alvo tem sobre o agente. Essa influência assume um carácter linear ao longo de todo ambiente, o que significa que a variação da magnitude do vector resultante da taxia de atracção a um alvo é inversamente proporcional à distância do agente ao alvo e, por isso, quanto mais distante o alvo da posição actual do agente, menor a magnitude do vector.

Tal como visto na secção 4.2.2, um comportamento pode ser uma reacção ou um comportamento composto. Assim, a concretização de uma reacção assume que este pode ser de dois tipos: Aproximar e Seguir. O Aproximar corresponde a uma reacção ao estímulo de todos os alvos presentes no ambiente, pois após a detecção de alvos no ambiente, são encontrados e somados os vectores de todos os alvos. O resultado é uma aproximação do agente ao centro da posição de todos os alvos. A Figura 5.1 representa o vector resultante da soma vectorial.

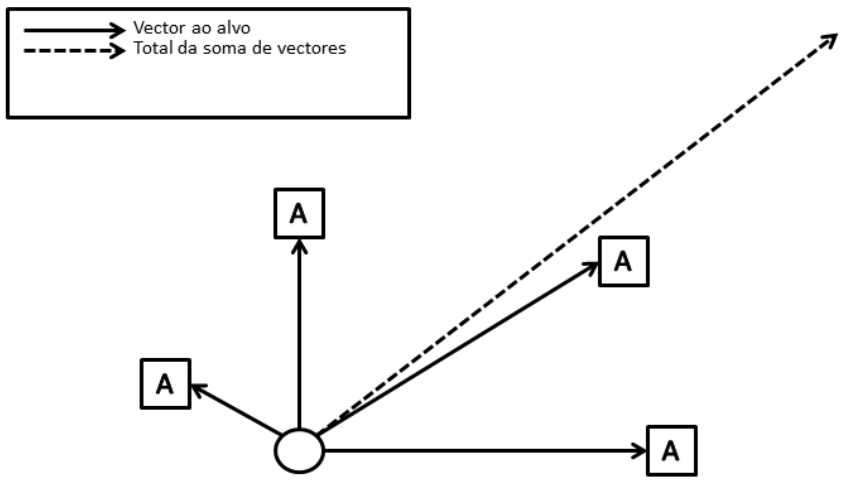


Figura 5.1 - Total da soma de todos os vectores

A segunda reacção implementada, nesta dissertação, é a reacção de Seguir. Esta reacção identifica o alvo mais próximo através da comparação das magnitudes de cada vector. O resultado é o agente seguir o alvo mais próximo da posição actual em que se encontra.

Nesta situação existe a necessidade de combinar estes dois comportamentos. É nesse sentido que o comportamento composto surge pois há uma necessidade de combinar os dois vectores existentes. Assim, implementou-se um comportamento composto que efectua a fusão entre as duas reacções. No entanto, pretende-se dar maior relevância ao comportamento Seguir porque direcciona o agente para o alvo mais próximo, como tal, a fusão efectuada entre os vectores resultantes das reacções corresponde a uma soma ponderadas entre os mesmos. A Figura 5.2 ilustra a formação do comportamento emergente mostrando a diferença entre a soma "normal" dos mesmos dois vectores.

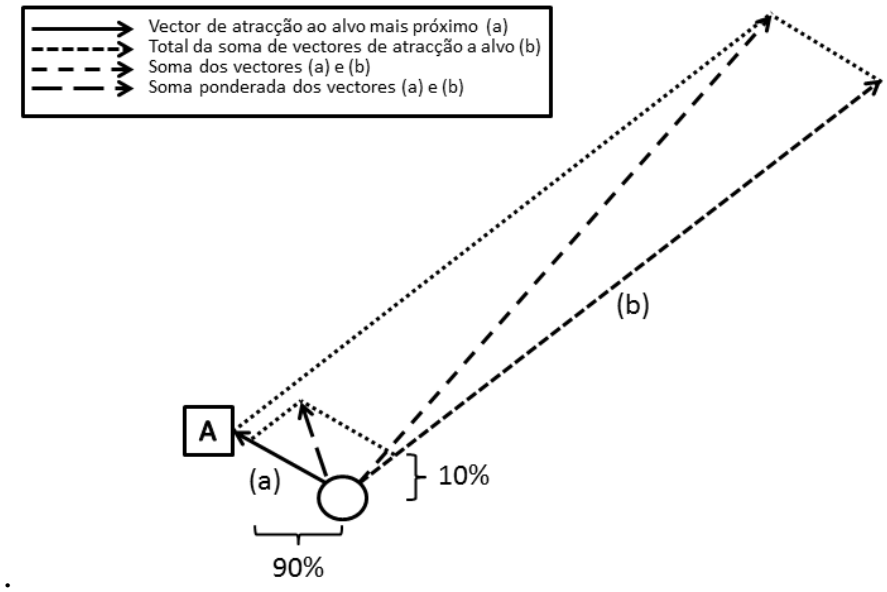


Figura 5.2 - Comparação entre o vector da soma ponderada e da soma entre os vectores: do alvo mais próximo (a) e da soma de todos os alvos (b).

É de notar na Figura 5.2, que o vector da soma ponderada é mais orientado para o alvo do que o vector resultante da soma entre o vector (a) e (b). A ponderação atribuída aos vectores neste exemplo é de 90% para o vector do alvo mais próximo e 10% para o vector resultante da soma de todos os alvos do ambiente. Assim, é possível ter um comportamento parametrizado sendo mais ou menos orientado para o alvo mais próximo dependendo da ponderação atribuída.

O facto de nesta camada o agente estar somente sobre a influência de campos de potencial atractores e o facto de combinar vectores através da soma ponderada, reduz a possibilidade de ocorrência do problema do mínimo local descrito por *Murphy* (2000, p.133) porque, neste caso, para além de não existirem campos de potencial repulsores, que poderiam fazer com a magnitude do vector resultante da soma de vectores fosse zero, também é sempre

seleccionado um alvo considerado o mais próximo, mesmo que todos os alvos estejam à mesma distância, sendo a este dado uma ponderação superior. No entanto, o facto mencionado não influencia a perturbação do agente por óptimos locais, os quais podem vir a não ser identificados, por exemplo, o obstáculo origina um óptimo local na camada reactiva que o agente não identifica.

Após ser obtido o vector correspondente ao comportamento emergente resultante da fusão dos dois vectores provenientes das reacções, é gerada a acção do agente que leva em consideração a orientação fornecida pelo comportamento e atribui uma velocidade, ou passo, de modo a concretizar a deslocação do agente.

5.4 Percepção, representação e aprendizagem

A aprendizagem ocorre através da assimilação da informação proveniente do ambiente. Essa informação necessita de ser manipulada de forma a ser armazenada para futura utilização do agente. Nesse sentido, existe uma discretização do ambiente e das acções possíveis do agente, para que possa ser utilizada quando necessária sem ocupar demasiados recursos, nomeadamente em termos de memória.

O nível de competência adaptativo corresponde à camada adaptativa. É nela, que ocorre o processo de aprendizagem e o primeiro nível de discretização do ambiente. Estes dois processos estão relacionados, uma vez que o processo de aprendizagem inicia-se com a discretização do ambiente para a posição actual do agente.

O processo de discretização do ambiente corresponde a uma redução da complexidade do ambiente. Para se conseguir discretizar um ambiente contínuo (representado pela PSA por valores reais) foi necessário arredondar os valores reais para valores inteiros e somá-los ao valor correspondente ao centro da posição do agente, uma vez que a representação do agente no ponto (1,1) na PSA, corresponde a posicioná-lo graficamente na posição (1.5, 1.5), ver Figura 5.3. Esse processo, permite que durante o processo de transição de estados, a transição seja considerada quando o centro do agente ultrapassa o limite de um estado discreto.

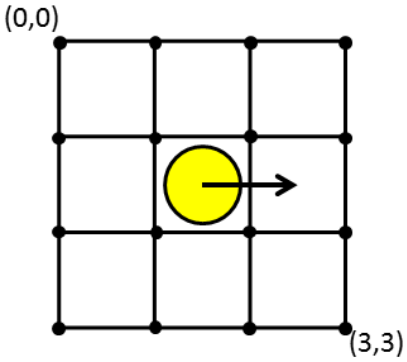


Figura 5.3 - Discretização linear em grelha

É de notar que a discretização referida anteriormente corresponde a uma discretização linear em grelha, pois todos os estados têm a mesma dimensão e estão organizados numa grelha bidimensional.

Na secção 5.2, a restrição 3 refere a necessidade de utilização de quatros acções discretas para que a aprendizagem nesta camada seja bem efectuada. Nesse sentido, foi necessário discretizar as acções do agente, uma vez que quando o agente está a ser controlado pela camada reactiva, as acções retornadas contêm a orientação obtida da soma de vectores dos alvos presentes no ambiente, por tal, são acções contínuas. Assim, discretizou-se o valor real fornecido pela orientação de uma acção contínua em múltiplos de $\frac{2\pi}{N}$, onde N corresponde ao valor 4 (correspondente às quatro acções). O resultado encontra-se ilustrado na figura seguinte.

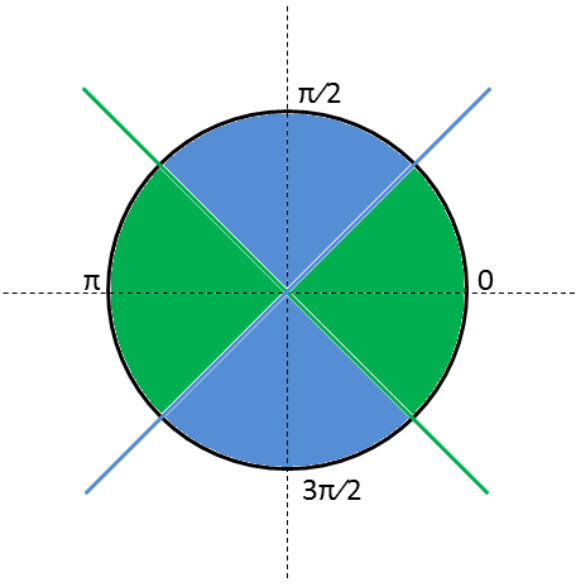


Figura 5.4 - Discretização da Acção

Com ambas as discretizações expostas, o agente passa a lidar com informação de âmbito discreto. Estes processos correspondem a uma abstracção de informação, uma vez que permite que o agente tenha informação reduzida sobre o ambiente, para que consiga lidar com a complexidade de um ambiente contínuo e reter a informação considerada relevante.

O processo de aprendizagem do agente necessita que uma das seguintes condições seja cumprida. A primeira corresponde à identificação de transições, uma vez que a existência de transições permite associar um valor de utilidade, denominado de valor Q, para um estado quando efectua uma certa acção. Outra condição é a colisão com os obstáculos pois, quando ocorre uma colisão com um obstáculo, não há transição de estado, apenas existe uma recompensa negativa por essa tentativa de transição. Consequentemente, o estado seguinte do agente mantém-se o mesmo.

A assimilação da informação proveniente do mundo pelo agente, corresponde ao armazenamento numa estrutura de dados que associa o estado e a acção ao valor Q obtido pela transição de estado. Esse valor é armazenado para cada transição possível entre estados discretos do ambiente, ou para cada estado onde uma determinada acção gera uma colisão com um obstáculo.

Na arquitectura de agente proposta, o Controlo Adaptativo actualiza um dos modelos da camada deliberativa através de uma condição de verificação de recolha de alvos. Esta condição é contraditória aos objectivos do subsistema adaptativo, uma vez que o próprio é responsável por aprender a evitar obstáculos. No entanto, esta verificação é útil para a camada acima, pois à medida que o agente aprende, actualiza o modelo da camada deliberativa que contém a representação dos alvos, removendo-o do modelo caso haja recolha. Assim, o modelo da camada deliberativa mantém-se consistente com a informação presente no ambiente.

5.5 Percepção e representação deliberativa

A discretização de mais alto nível abstrai o ambiente segundo a posição percepcionada dos obstáculos e a posição obtida dos alvos. A discretização utiliza uma decomposição em *Quadtree* de modo a dar foco aos pontos considerados relevantes, reduzindo o nível de detalhe em pontos não relevantes.

A concretização da *Quadtree* inicia-se com a criação de uma árvore que contém a área correspondente à zona onde o agente se encontra e a posição dos alvos presentes no ambiente. Por cada ponto adicional que se queira inserir é verificado se o mesmo se encontra dentro da área. Se sim, então é verificado se já se alcançou a restrição da resolução máxima que condicionará a inserção do ponto na estrutura (restrição 5). Caso se verifique que a resolução máxima ainda não foi alcançada, então subdivide-se o nó em quatro quadrantes (novos nós), caso este ainda não tenha sido subdividido, e repete-se o mesmo processo para cada nó da árvore. A figura seguinte ilustra o processo mencionado mas aplicado no contexto do trabalho.

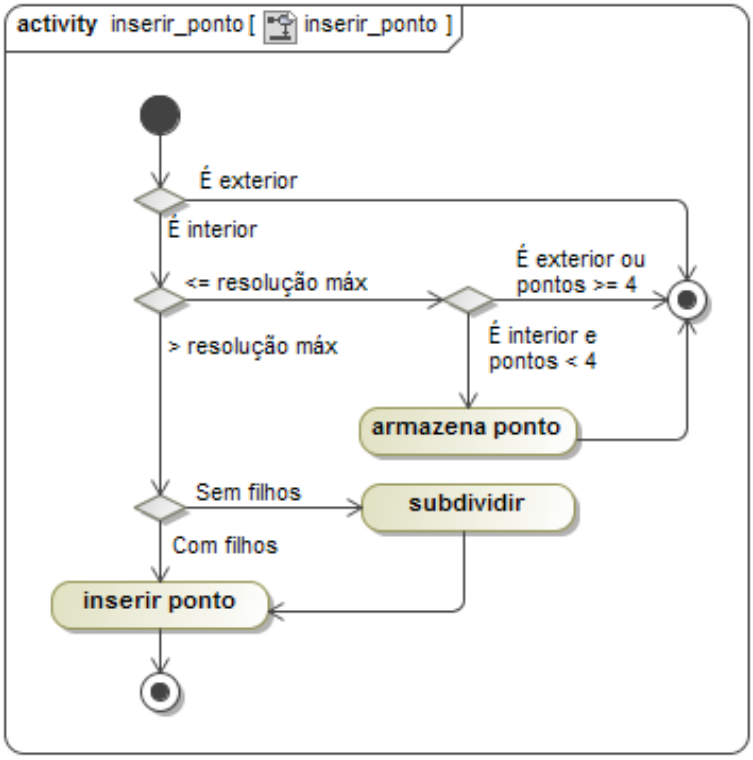


Figura 5.5 - Processo de inserção de um ponto na Quadtree

Após o processo de inserção de todos os pontos relevantes na estrutura obtém-se a *Quadtree* que será utilizada para a discretização do ambiente. A figura seguinte ilustra um exemplo do ambiente discretizado.

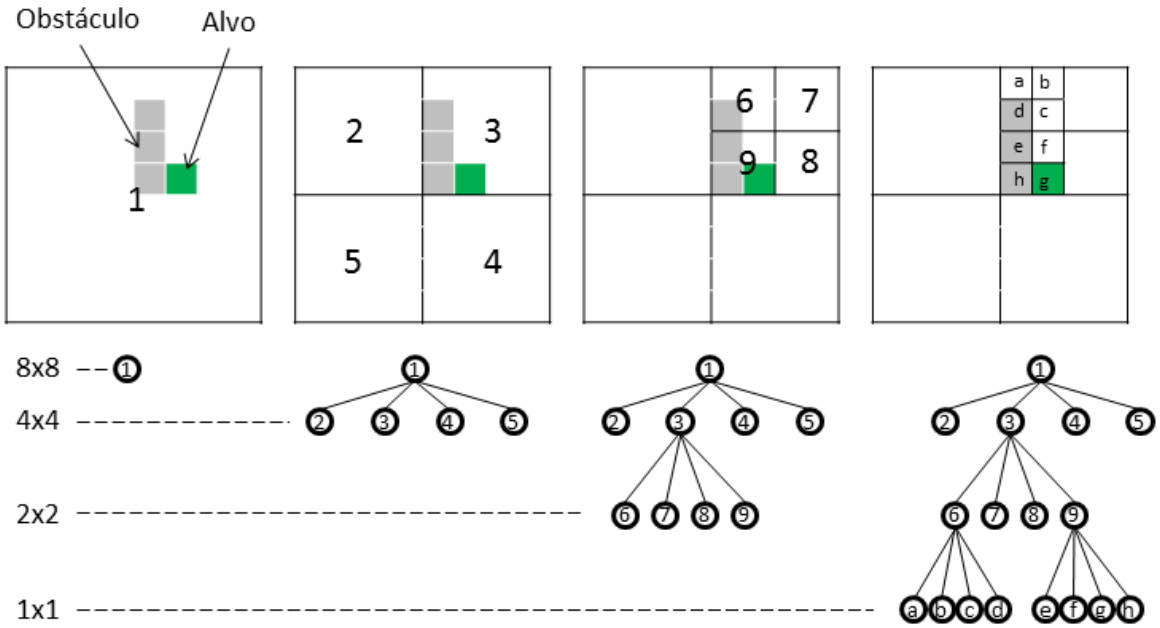


Figura 5.6 - Representação do ambiente discretizado utilizando a *Quadtree*

É de notar a correspondência entre o ambiente discretizado pelo processo da *Quadtree* e a formação da estrutura em árvore. Na Figura 5.6, verifica-se que para cada nível da *Quadtree* a dimensão da área decomposta é reduzida a metade, e por tal cada nível é composto por áreas da mesma dimensão, a este processo denomina-se decomposição regular da *Quadtree* (Samet, 1984). Para ambientes cuja dimensão não corresponde a uma potência de dois, a decomposição da *Quadtree* pode não formar áreas de mesma dimensão para o mesmo nível de decomposição. Neste caso um ponto relevante poderia não preencher uma área tal como especificado na restrição 6 presente na secção 5.2. Desde modo, durante o processo de obtenção das áreas da *Quadtree* o mecanismo de fornecimento de áreas subdivide as áreas que caem nessa condição em áreas de dimensão de 1x1.

A secção 4.4, refere que a camada deliberativa contém modelos deliberativos. Esses modelos permitem armazenar informação simbólica, suportar a discretização e auxiliar o processo de cálculo da política para o ambiente. Os modelos presentes nessa camada são: o Modelo de Representativo, o Modelo Deliberativo e o Modelo PDM.

O Modelo de Representativo tem duas funções fundamentais. A primeira função corresponde à representação interna do ambiente, atribuindo uma classificação interna aos estados que vai conhecendo à medida que explora o ambiente. A segunda deriva da primeira e corresponde à delimitação da área a ser utilizada pela decomposição em *Quadtree*.

A classificação dos estados explorados é realizada com base na recompensa recebida pela transição ou não de estado. Isto significa que recompensas negativas, quando não existe transição de estado, resultam numa classificação do estado de "obstáculo". As transições com recompensas positivas resultam numa classificação em "alvo" e as restantes em "vazio". As classificações a "vazio" permitem saber quais os estados que o agente efectivamente conhece. O Modelo Representativo pode ser preenchido com valores que não correspondem à realidade (caso a classificação seja mal feita), no entanto também permite ser corrigida caso o agente volte a passar pelo mesmo estado, classificando-a correctamente. A segunda função do Modelo Representativo deriva da primeira, pois a classificação de cada estado visitado pelo agente auxilia o processo de criação da *Quadtree*, uma vez que delimita a área a ser utilizada para a formação da mesma. A figura seguinte mostra um exemplo deste processo.

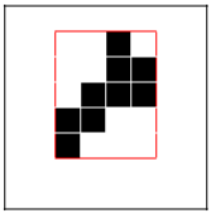


Figura 5.7 - Área de conhecimento do agente para obtenção da área da *Quadtree*

Na Figura 5.7, para representar o conhecimento do agente foram utilizados os quadrados a preto, onde estes podem ser qualquer tipo de elemento. É de notar que a área que abrange todo o conhecimento do agente (representada pelo contorno a vermelho) contém estados que não são conhecidos pelo agente. No entanto, essa situação não é relevante dado que o objectivo é a obtenção da área para o cálculo da *Quadtree*. A informação que será relevante para a *Quadtree* é o conhecimento que o agente tem sobre o mundo pois, esse sim, poderá influenciar o cálculo da *Quadtree* ao dar foco às zonas que contém pontos de interesse.

Qualquer alteração que ocorra no mundo necessita de ser actualizada no Modelo Representativo. Assim, quando o agente recolhe um alvo ou colide com um obstáculo este modelo é actualizado para que a informação se mantenha coerente com o ambiente. Nesse sentido, este modelo é actualizado juntamente com os modelos da camada adaptativa, uma vez que a informação utilizada para representar o mundo é a mesma utilizada para actualizar os modelos da camada abaixo. No entanto, o Modelo Representativo é abordado no subsistema deliberativo pelo facto de a sua funcionalidade ser de uso exclusivo deste subsistema.

Outro modelo mencionado é o Modelo Deliberativo. Este utiliza o Modelo Representativo para representar o mundo com base num modelo de transição e num modelo de recompensa baseado na discretização em *Quadtree* para, posteriormente, ser utilizado pelo Modelo PDM. O modelo de transição corresponde à transição entre áreas na *Quadtree*, enquanto o modelo de recompensas corresponde às recompensas por essas mesmas transições.

O Modelo Deliberativo é actualizado sempre que se recolhem alvos ou ocorre uma colisão com um obstáculo, e por cada um deles é solicitado ao Coordenador Sensorial Deliberativo que forneça uma discretização não linear do ambiente em *Quadtree* sendo passado como parâmetro a área conhecida e os pontos relevantes fornecidos pelo Modelo Representativo. Assim que o Modelo Deliberativo obtém a representação em *Quadtree* para o conhecimento corrente, são actualizados os modelos de transição e de recompensa para cada área presente na *Quadtree* que não seja um obstáculo. O processo de actualização do modelo de transição é através da identificação das áreas vizinhas para cada área analisada numa determinada direcção (acções discretas). Já o modelo de recompensa é actualizado através da verificação das respectivas recompensas para essas transições entre áreas da *Quadtree*, por exemplo, uma transição para um estado que contenha um obstáculo origina uma recompensa negativa, para um alvo uma recompensa positiva e para as restantes uma recompensa nula.

Durante o processo de actualização dos modelos (de transição e recompensa), é efectuada uma contagem das transições com base na intersecção entre a área sucessora e a aresta da área actual numa determinada direcção, por exemplo, na Figura 5.8 é ilustrada uma representação de uma área com três áreas vizinhas (a, b e c) do lado direito. Cada uma dessas áreas tem uma relação com a área que está a ser analisada, ou seja, a área *a* tem dimensão de 4x4 mas apenas metade da sua dimensão é que tem uma intersecção com a área principal, ocupando duas unidades da dimensão de 8x8. Assim, a contagem corresponde ao número de unidades de intersecção. Esse processo será útil para o cálculo da probabilidade de transição entre áreas que será utilizado para a geração do Modelo PDM, onde a probabilidade de transição advém da divisão do valor de contagem pela dimensão da área de intersecção, ou seja, para a área *a* a probabilidade é igual a 0,25. O cálculo da probabilidade de transição é realizado de acordo com a seguinte fórmula:

$$\mathrm{P(área)} = \frac{valor\ de\ contagem}{dimensão\ da\ área\ de\ intersecção} \qquad (2)$$

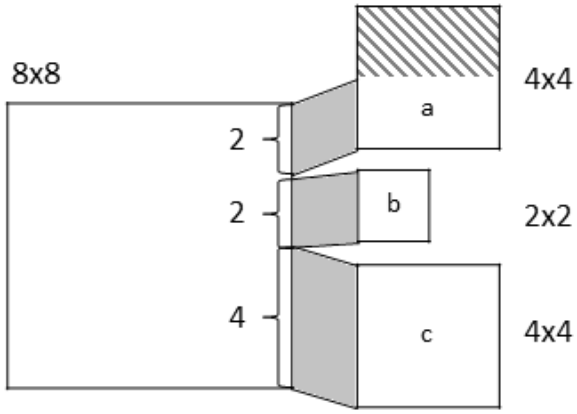


Figura 5.8 - Áreas vizinhas numa *Quadtree*

O último modelo mencionado é o Modelo PDM. Este modelo permite a construção do ambiente segundo a estrutura dos Processos de Decisão de *Markov* (PDM), onde o ambiente é representado em termos de comportamento e recompensas possíveis para o agente através da seguinte organização estrutural:

- Conjuntos de estados;
- Conjuntos de acções;
- Modelo de probabilidade de transição[11];
- Modelo de recompensa;
- Estados sucessores.

Uma vez que a representação do ambiente é baseada na *Quadtree*, então o conjunto de estados corresponde ao conjunto de áreas que formam a *Quadtree*. O conjunto de acções corresponde às acções possíveis em cada estado, referidas nas restrições da Secção 5.2. O modelo de probabilidade de transição corresponde à probabilidade descrita anteriormente para cada área vizinha de um certo estado numa determinada direcção (acções possíveis), excepto quando se trata de um estado absorvente (objectivo), nesse caso a probabilidade de transição é igual a zero. O modelo de recompensa corresponde às recompensas obtidas por cada transição, tal como referido anteriormente. Os estados sucessores permitem, através da utilização do modelo de transição do Modelo Deliberativo, saber quais os estados sucessores para uma determinada área e identificar o estado absorvente.

[11] Termo usado para evitar confusão com o modelo de transição mencionado anteriormente

Assim, o Modelo PDM auxilia o processo de cálculo da utilidade e política num Processo de Decisão de *Markov*.

5.6 Utilidade e política comportamental deliberativa

O Modelo PDM descrito na secção anterior corresponde ao modelo utilizado para o cálculo da função de utilidade do Processo de Decisão de *Markov*, que gera uma política comportamental para o agente tendo em consideração os estados abstractos presentes na discretização não linear em *Quadtree*. Este processo inicia-se com o cálculo da utilidade, e posteriormente o cálculo da política referente ao mapeamento de uma acção para cada estado do ambiente.

Para o cálculo da utilidade, o processo foi alterado de modo a ser utilizado no contexto de operação em tempo real, de modo a dividir a carga computacional entre vários ciclos do agente. Assim, o cálculo da utilidade é armazenado internamente para cada passo do agente enquanto estiver a ser realizado para uma *Quadtree*. A equação utilizada para o cálculo da utilidade deriva da equação de *Bellman* (1954) e denomina-se de equação de actualização de valor, representada na fórmula abaixo:

$$U(s)_{k+1} \leftarrow \max_{a \in A(s)} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U(s')_k] \qquad \textbf{(3)}$$

onde $T(s,a,s')$ corresponde ao modelo de probabilidade de transição, $R(s,a,s')$ ao modelo de recompensas e $U(s)_k$ à soma das recompensas expectáveis acumuladas quando no estado s se actua optimamente, o que significa que os valores dos estados podem ser determinados em função dos estados sucessores.

A resolução da fórmula acima é ilustrada no algoritmo da iteração de valor, adaptado para o contexto deste trabalho, mostrado na figura seguinte.

Função utilidade (modelo) **retorna** a função utilidade

entradas: *modelo*, o ModeloPDM composto pelo conjunto de estados *S*, acções *A(s)*, modelo de probabilidade de transição *T(s,a,s')* e modelo de recompensas *R(s,a,s')*
NUM_ITERACOES, número de ciclos pretendidos
γ, factor de desconto
ε, erro máximo na utilidade de cada estado

variáveis locais: *U*, Função de utilidade *U(s)*
δ, alteração máxima na utilidade de qualquer estado numa iteração
i, iteração

variáveis de instancia:
U', vector de utilidades para cada estado do conjunto, inicializado a zero

$\Delta' \leftarrow \varepsilon (1-\gamma)/\gamma$
$i \leftarrow 0$
repetir $i <$ *NUM_ITERACOES*
  $U \leftarrow U'$; $\delta \leftarrow 0$
  Para cada estado *s* **em** *S* **fazer**
    $U'[s] \leftarrow \max_{a \in A(s)} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U[s']]$
    $\delta \leftarrow \max(\delta, |U'[s] - U[s]|)$
  Se $\delta < \Delta'$ **então**
    retorna *U'*
  $i \leftarrow i + 1$
retorna *Nada*

Figura 5.9 - Algoritmo de iteração de valor para o cálculo da utilidade

O processo de cálculo da utilidade é parametrizado para permitir o controlo do número de iterações a ser efectuado por cada passo do agente. Enquanto a discretização do mundo se mantém a mesma, a cada passo do agente é iterado o vector de utilidades tantas vezes quanto for parametrizado e armazenado internamente para ser iterado novamente no passo seguinte. O processo de iteração termina quando a utilidade convergir ou quando nova discretização for criada, derivada de uma colisão ou recolha de alvos. Para convergir, o valor máximo da diferença de utilidade entre dois estados, de todos os estados do modelo ($|U'[s] - U[s]|$), deve ser inferior a $\varepsilon * ((1-\gamma))/\gamma$, onde ε é o erro máximo da utilidade para qualquer estado. Assim, é possível ter um resultado que distribui o peso do cálculo da utilidade por várias etapas de iteração de utilidade, de modo que a utilidade seja obtida obedecendo às restrições temporais de resposta em tempo real.

Quando a função de utilidade converge é verificado, para cada estado do Modelo PDM, qual é a acção que tem a maior utilidade. Essa acção é mapeada para cada estado do modelo, sendo deste modo encontrada a política óptima.

5.7 Conclusão

Neste capítulo apresentou-se os detalhes da implementação do modelo proposto, indicando as características do ambiente de utilização e a forma como o agente lida com alguns aspectos inerente de operação em tempo real. Definiu-se as restrições para a utilização do sistema implementado e descreveu-se as soluções encontradas para a concretização dos conceitos teóricos estudados.

Segue-se uma análise crítica da arquitectura implementada, retirando resultados qualitativos do comportamento geral do agente.

6 Resultados Experimentais

Na secção anterior descreveu-se detalhadamente a implementação da arquitectura de agente inteligente híbrido proposta para esta dissertação. Neste capítulo pretende-se analisar o comportamento do agente nas suas três vertentes possíveis: reactiva, adaptativa e deliberativa. De seguida verifica-se qual o comportamento do agente, como um todo, perante um ambiente complexo e com necessidade de operação em tempo real. Nesse sentido, criaram-se dois casos experimentais que utilizam ambientes diferentes: o primeiro é mais simples e com apenas um alvo; o outro é mais complexo e contém mais alvos.

6.1 Caso experimental 1

Para este caso experimental utilizou-se um ambiente estático, determinístico, e parcialmente observável. É semelhante ao *gridworld* proposto por *Bianchi* (2004) mas com a excepção de ser um ambiente contínuo. Este ambiente será utilizado para testar o comportamento do agente numa tentativa de explicar com base nas opções de implementação adoptadas. O ambiente definido é ilustrado na figura seguinte.

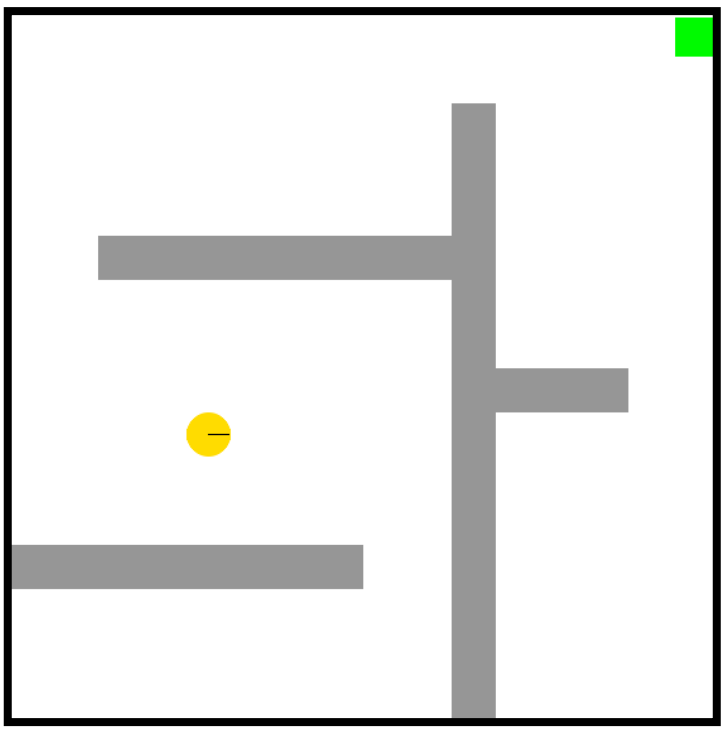


Figura 6.1 - Ambiente com sala e com paredes semelhante ao gridworld de Bianchi (2004)

Uma vez que a arquitectura está separada em três camadas, testou-se o agente nas três vertentes: reactiva, adaptativa e deliberativa. Segue-se o resultado obtido.

6.1.1 Subsistema Reactivo

O subsistema reactivo, como referido no decorrer desta dissertação, utiliza-se de campos de potencial para orientar o agente para os motivadores definidos, sendo gerados vectores orientados para o motivador que representam a influência dos motivadores sobre o agente, originando um comportamento do tipo taxia. Neste cenário concreto, os motivadores correspondem às posições dos alvos detectados.

Considerando o ambiente apresentado na Figura 6.1, o agente ao ser colocado num ambiente que não conhece (não tem acções adaptativas nem deliberativas utilizáveis), é controlado pela camada reactiva que orienta o agente para o alvo. A figura seguinte ilustra essa situação.

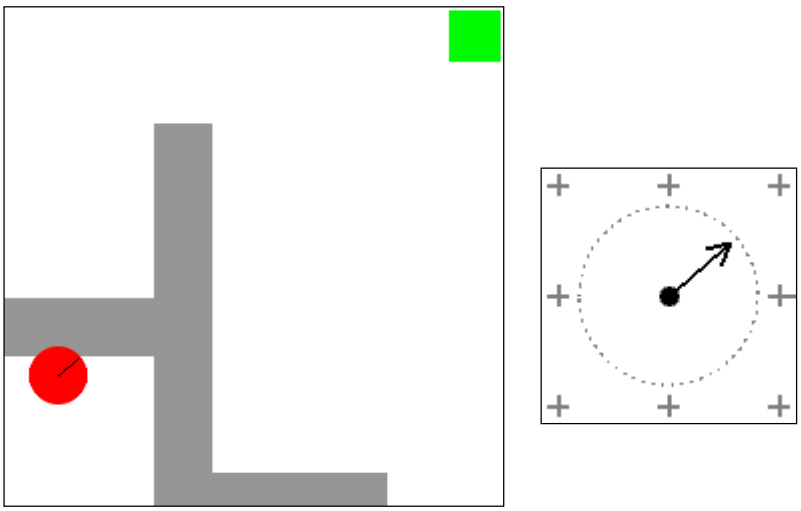


Figura 6.2 - Orientação do agente para o alvo

A Figura 6.2 ilustra a orientação do agente para o alvo bem como o vector de acção correspondente. É de realçar que a orientação do agente está a ser resultante de apenas um alvo, logo o vector resultante dos comportamentos Aproximar e Seguir, bem como o vector resultante da soma ponderada entre os mesmos têm todos a mesma orientação, daí ser ilustrado apenas um vector. Outra situação relevante corresponde ao facto de o agente avançar em direcção ao alvo mas não conseguir recolhê-lo, devido ao facto de existir uma barreira que impede o agente de se aproximar do alvo. Esse obstáculo, para o agente, corresponde a um óptimo local que o agente não o identifica e impede a aproximação ao alvo. No entanto,

definiu-se que a identificação e a resolução comportamental do agente face a um obstáculo não compete ao subsistema reactivo, sendo nesse sentido que a camada adaptativa auxilia a camada reactiva na sua limitação.

6.1.2 Subsistema Adaptativo

Para testar o subsistema adaptativo utilizaram-se os dois processos de aprendizagem e dois métodos exploratórios que foram implementados. As aprendizagens foram combinadas com cada política de selecção de acção (método exploratório) a fim de verificar semelhanças e diferenças no comportamento do agente. Assim, o método exploratório *ε-greedy* foi utilizado segundo o custo de movimentação descrito por *Watkins* (1989), onde cada movimento do agente tem um custo imediato, ou seja, uma penalização. Já o método exploratório baseado na heurística favorece a aproximação e penaliza o afastamento ao alvo. O favorecimento à aproximação é através da não penalização, ou seja, a recompensa igual a zero. Já ao afastamento é atribuído uma recompensa negativa sempre que o agente se afasta do alvo.

Posto isto, testou-se o comportamento do agente quando se adiciona a camada adaptativa à camada reactiva. Os dois níveis de competência quando combinados resultam em comportamentos diferentes para cada tipo de aprendizagem. Neste caso, testou-se, inicialmente, a aprendizagem Q para os dois tipos de métodos exploratórios.

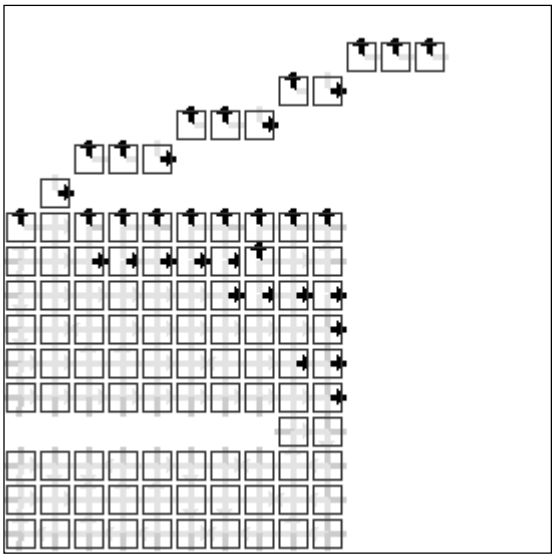


Figura 6.3 - Aprendizagem Q com método exploratório *ε-greedy*

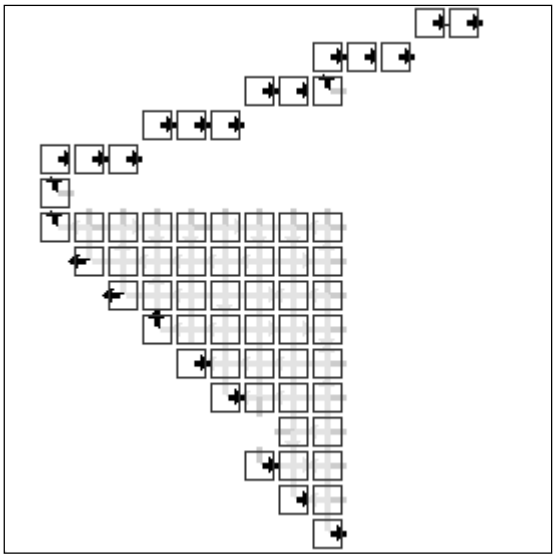


Figura 6.4 - Aprendizagem Q com o método exploratório baseado na heurística

As figuras acima ilustram os estados aprendidos (representado pelos quadrados) pelo agente quando utilizada a aprendizagem Q com dois tipos de métodos exploratórios. Nestes resultados verifica-se que a aprendizagem Q, para o método exploratório baseado na heurística, explorou muito menos do que o mesmo com o método *ε-greedy*. No entanto, o número de passos do agente até encontrar o alvo teve uma diferença significativa, onde a aprendizagem Q com *ε-greedy* resultou em 609 passos do agente até encontrar o alvo, enquanto que com a heurística o resultado foi 900 passos. Efectuou-se o mesmo teste mas, desta vez, com a aprendizagem *Dyna-Q*.

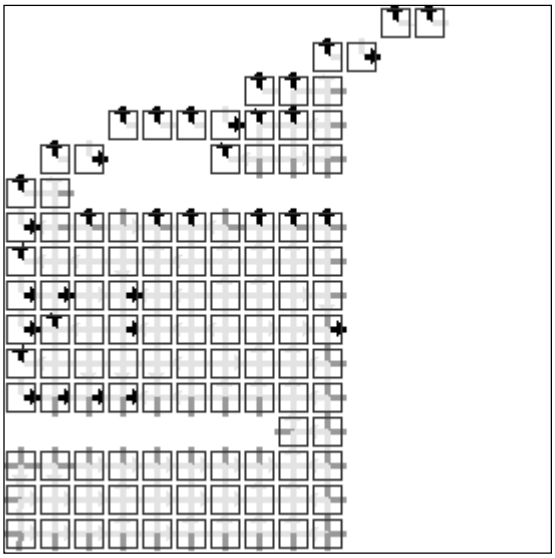


Figura 6.5 - Aprendizagem *Dyna-Q* com exploração *ε-greedy*

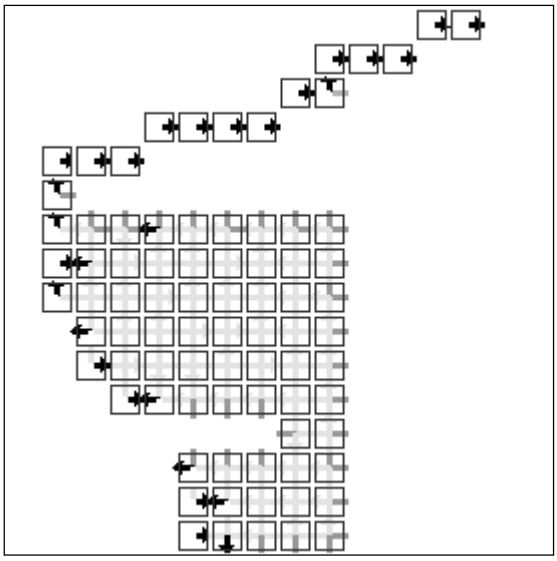


Figura 6.6 - Aprendizagem *Dyna-Q* com exploração heurística

As figuras acima ilustram os estados aprendidos pelo agente quando se utilizou a aprendizagem *Dyna-Q* com os dois métodos exploratórios implementados. Sabendo que a aprendizagem *Dyna-Q* efectua simulações internas para a iteração do valor *Q(s,a),* definiu-se o número de iterações igual a 100 para cada um dos testes. Nesta análise, verifica-se que a exploração com a heurística foi, novamente, menos abrangente dos que a exploração *ε-greedy.* No entanto, ao contrário do resultado anterior, a exploração heurística necessitou de menos passos para chegar ao alvo. A tabela abaixo mostra o resultado para os dois testes efectuados. Os valores indicados representam o número de passos para atingir o alvo.

Tabela 6.1 - Comparação entre aprendizagens e políticas de avaliação

	Política *ε-greedy*	Política heurística
Aprendizagem Q	609 passos	900 passos
Aprendizagem *Dyna-Q*	480 passos	317 passos

A aprendizagem *Dyna-Q* combinada com o método exploratório baseado na heurística teve um resultado melhor do que a exploração com a política *ε-greedy,* onde em 317 passos conseguiu recolher o alvo, isto corresponde a uma melhoria significativa face ao teste anterior.

Assim, verifica-se que a política de selecção de acção baseada na heurística depende do tipo de aprendizagem que se está a utilizar, uma vez que na Tabela 6.1 essa política tem os valores

extremos para os dois tipos de aprendizagem. Com a aprendizagem Q a heurística não favorece a exploração e nem favorece a rápida convergência, enquanto que com a aprendizagem *Dyna-Q* a convergência é muito mais rápida. Tal resultado pode ser explicado dada a rápida propagação de valor da aprendizagem *Dyna-Q* e o carácter exploratório da política de selecção de acção baseada na heurística, uma vez que o método exploratório baseado na heurística explora, inicialmente, os estados mais próximos da heurística que não conhece e a partir do momento que conhece todas as suas acções possíveis para esse estado, a sua exploração passa a ser segundo uma política *greedy,* o que significa que irá seleccionar as acções com melhor valor Q. Assim, quanto mais depressa aprender a desviar-se dos obstáculos e propagar esse valor para os estados vizinhos, mais depressa saberá quais são as melhores acções a serem efectuadas. Deste modo, torna-se compreensível o facto da política de selecção de acção baseada na heurística ter um melhor desempenho quando associado à aprendizagem *Dyna-Q*.

As figuras 6.3, 6.4, 6.5 e 6.6 ilustram dois níveis de competências (camada reactiva e adaptativa). A camada reactiva gera uma acção representada por um vector, cuja orientação da é um valor real. Já a camada adaptativa dispõe de um conjunto de acções discretas que são utilizadas como resposta fornecida por essa camada. Assim, quando estes dois níveis de competência são combinados, o comportamento do agente passa a ter acções discretas e acções contínuas. Quando o agente está sob o controlo da camada reactiva cada movimento é aprendido pela camada adaptativa através da discretização do estado e da acção do domínio contínuo. No entanto, se a acção da camada reactiva seguir uma orientação na diagonal, essa aprendizagem pode corresponder a uma aprendizagem incoerente no domínio discreto uma vez que o agente passa a conhecer transições entre estados discretos na diagonal. Um exemplo desta particularidade, encontra-se em todas as imagens quando o agente se aproxima do alvo, onde se verifica que a aprendizagem entre dois estados discretos é na diagonal.

A situação descrita acima torna-se particularmente relevante quando o agente se encontra junto a um obstáculo, pois pode fazer uma falsa correspondência entre o conhecimento adquirido e o ambiente real. Na Figura 6.2 apresentada na secção anterior, o agente ainda não conhece o ambiente e por tal é controlado pelas acções fornecidas pela camada reactiva. À medida que o agente navega pelo ambiente, a camada adaptativa discretiza o estado e a acção e aprende as transições entre estados. No entanto, ao colidir com um obstáculo, o agente recebe um reforço negativo alto e permanece no mesmo estado. A representação interna da

colisão continua a depender da discretização do estado e da acção, como tal, verifica a orientação actual do agente, discretiza-a e armazena a informação adquirida. Contudo, para a situação representada na Figura 6.2, a orientação actual do agente é discretizada para o ângulo 0.0, que corresponde a andar para a direita. Assim, o agente retém do mundo que ao estar no estado discreto actual e ao efectuar a acção discreta para a direita colide com um obstáculo, não sendo essa informação correcta. A aprendizagem incorrecta do ambiente resulta numa propagação incorrecta desse valor para os estados vizinhos que, dependendo do tipo de ambiente e do tipo de aprendizagem, pode resultar num comportamento do agente que evita voltar ao local mal identificado. Esta situação poderia ser minimizada com uma discretização diferente do estado e das acções, como por exemplo, discretização do estado em octógonos e a disponibilização de oito acções.

6.1.3 Subsistema Deliberativo

A camada deliberativa permite obter uma política comportamental baseada em Processos de Decisão de *Markov* (PDM). Essa política é obtida para uma representação do ambiente em áreas que compõem a *Quadtree*. A resposta fornecida por esta camada convém ser dada num tempo útil, de modo a utilizar o conhecimento obtido para auxiliar as camadas hierarquicamente abaixo no comportamento geral do agente.

Para análise do comportamento do agente quando está sob o controlo da camada deliberativa, efectuaram-se dois tipos de testes que utilizam um mecanismo deliberativo baseado em PDM para a obtenção da política comportamental, num ambiente discretizado em *Quadtree*. Inicialmente calculou-se a política para o ambiente totalmente observável e para o ambiente parcialmente observável, sendo este último calculado em tempo real. De seguida, colocou-se o agente num ambiente não-determinístico e analisou-se o comportamento do agente.

A figura seguinte ilustra o primeiro caso.

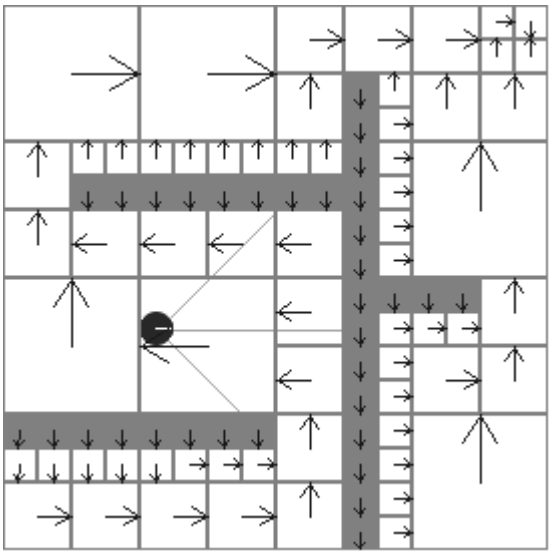


Figura 6.7 - Discretização não linear em *Quadtree* para o caso experimental 1

A Figura 6.7 ilustra o cálculo da política para uma discretização em *Quadtree* para um ambiente totalmente observável. O ambiente utilizado é um ambiente cujo número de estados discretos em grelha linear corresponde a um ambiente de dimensão 16x16, o que corresponde a uma potência de dois. Esse facto é relevante para a discretização pois, como se pode observar, a discretização em *Quadtree* é decomposta regulamente, isto é, cada nível da *Quadtree* é dividida em quatro partes igual (Samet, 1984).

Quando se pretende fazer uma discretização de um ambiente que não tem dimensão que seja múltipla de potência de dois, então a discretização já não corresponde a uma discretização como o da Figura 6.7. Essa situação ocorre quando o agente ainda está a conhecer o ambiente e a formar um modelo interno, tal como descrito na Secção 4.4.1. A área inicial corresponde à área que é formada pelos estados discretos que o agente conhece e os pontos relevantes que correspondem aos pontos classificados como alvo ou obstáculo. Assim, a figura seguinte ilustra uma discretização em tempo real para o cálculo da política.

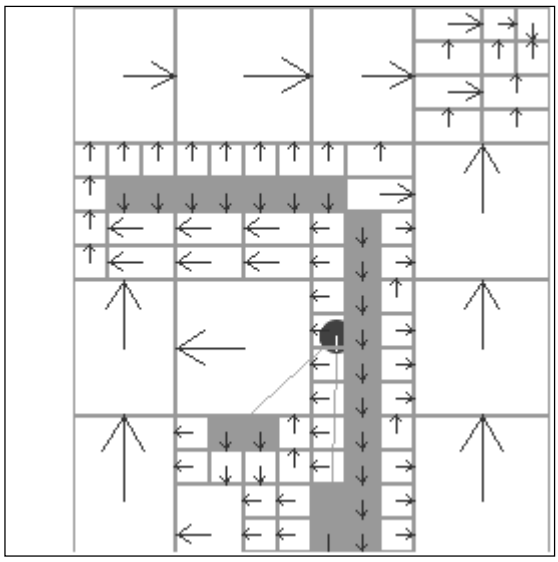


Figura 6.8 - Discretização não linear em *Quadtree* em tempo real

É de notar que a discretização que está a ser utilizada para o cálculo da política não corresponde a uma discretização com uma decomposição regular pois verifica-se a existência de áreas que não correspondem a um quadrado. Isso ocorre devido ao facto de o agente não conhecer o ambiente na sua totalidade e a área utilizada para a discretização não corresponder à dimensão total do ambiente, ou seja, a dimensão não é um múltiplo da potência de dois. Na Figura 6.8, observa-se também que os locais que o agente não conhece são considerados como espaços vazios, até que seja identificado a existência de um obstáculo. Para além disso, as áreas à volta do ambiente correspondem a obstáculos, logo, o agente estende a área de discretização para além da zona visível na PSA quando colide com os obstáculos posicionados na periferia do ambiente.

A zona que delimita o conhecimento que o agente tem do ambiente está representada na implementação do modelo proposto no Modelo Representativo. Este modelo é actualizado na camada adaptativa para ser utilizado pela camada deliberativa, por isso este modelo tem a mesma limitação que foi identificada na discretização da acção do subsistema anterior, ou seja, está propenso a falhas na classificação de um estado discreto.

A falha de classificação do ambiente torna-se mais visível quando se introduz o não determinismo no ambiente, ou seja, quando existe um grau de incerteza quanto ao estado seguinte resultante de uma determinada acção. A discretização não linear baseada na *Quadtree* dá foco aos pontos que correspondem a alvos e obstáculos, esses pontos são recebidos do Modelo Representativo que os identifica e armazena à medida que explora o

ambiente. Se uma má classificação de um obstáculo ocorre, isso significa que a *Quadtree* irá dar foco a uma zona que não tem relevância. Quando adicionado o não determinismo no ambiente, sempre que o agente se aproxima de um obstáculo a probabilidade de colidir acresce dado o facto do movimento do agente não ser determinista. Logo, mesmo que o agente efectue uma acção que consequentemente resultaria numa não colisão, pode colidir.

Dado a situação acima descrita, essa colisão resulta num acréscimo de estados na *Quadtree*, o que consequentemente acresce o número de transições possíveis no espaço de estados e significa que as necessidades computacionais crescem exponencialmente com o número de variáveis de estado (Sutton, 2012). Esta situação é descrita por *Bellman* como "*the curse of dimensionality*".

Na Figura 6.8, a política é calculada segundo o algoritmo de iteração de valor presente na secção 5.6 para a discretização em *Quadtree*. Nessa fórmula, existem duas variáveis que são relevantes para a convergência do cálculo: o factor de desconto (γ) e o erro máximo da utilidade entre dois estados (ε). O factor de desconto permite que se possa determinar o horizonte de propagação do valor de utilidade de cada estado, esse valor encontra-se entre o limiar 0 e 1.

Segundo *Russell* e *Norvig* (2010), a actualização de *Bellman* é uma contracção[12] por um factor de γ no espaço dos valores de utilidade, o que significa que a iteração de valor é reduzida por um factor de γ a cada iteração e que converge para uma solução única sempre que $\gamma < 1$. Outra constatação de *Russell* e *Norvig* é que cada utilidade está limitada no valor pela fórmula $2R_{max}/(1-\gamma)$, onde R_{max} corresponde à recompensa máxima. Assim, verificou-se que é possível calcular o número de iterações necessárias para atingir um determinado erro (ε). A equação seguinte determina esse valor:

$$N = \left\lceil \frac{\log\left(\frac{2R_{max}}{\varepsilon(1-\gamma)}\right)}{\log\left(\frac{1}{\gamma}\right)} \right\rceil \qquad (4)$$

[12] Função que tem um ponto fixo e cujo resultado, para qualquer argumento, está próximo do ponto fixo por um factor constante em relação aos argumentos originais.

Com a fórmula anterior, é possível determinar em quantas iterações se pretende que seja convergido o cálculo da utilidade para um determinado erro. Deste modo, resta saber quais os melhores valores para serem utilizados na parametrização, para as variáveis γ e ε.

Efectuou-se o seguinte levantamento do número de iterações para as variáveis γ e ε para valores no intervalo compreendido entre $0{,}9 \leq \gamma \leq 0{,}99$ e $0{,}01 \leq \varepsilon \leq 0{,}1$.

Tabela 6.2 - Número de iterações para um determinado ε

ε\γ	0,99	0,98	0,97	0,96	0,95	0,94	0,93	0,92	0,91	0,9
0,01	985.4	455.9	289.0	208.6	161.7	131.0	109.7	93.8	81.7	72.1
0,02	916.4	421.6	266.3	191.7	148.2	119.9	100.1	85.5	74.4	65.6
0,03	876.0	401.5	253.0	181.7	140.3	113.3	94.5	80.7	70.0	61.7
0,04	847.5	387.3	243.6	174.7	134.7	108.7	90.6	77.2	67.0	59.0
0,05	825.3	376.2	236.2	169.2	130.3	105.0	74.5	64.6	56.9	56.9
0,06	807.1	367.2	230.2	164.8	126.8	102.1	85.0	72.3	62.7	55.1
0,07	791.8	359.6	225.2	161.0	123.8	99.6	82.8	70.5	61.0	53.7
0,08	778.5	353.0	220.8	157.7	121.2	97.5	81.0	68.9	59.7	52.4
0,09	766.8	347.1	216.9	154.8	118.7	95.6	79.4	67.5	58.4	51.3
0,1	756.3	341.9	213.5	152.2	116.8	93.9	77.9	77.9	57.3	50.3

Dada a simplicidade deste ambiente definiu-se que a parametrização utilizada para os valores de γ e ε deve ser um número reduzido de iterações de modo a auxiliar o comportamento do agente ainda em tempo útil, assim definiu-se que $\gamma = 0{,}9$ e $\varepsilon = 0{,}1$. Deste modo o número de iterações necessárias para convergir é igual a 50. Se por cada passo se fizerem duas iterações ao espaço de estados do modelo, então significa que o agente converge em 25 passos.

Neste caso experimental, verifica-se que não existem grandes problemas em termos de convergência do agente para o objectivo estabelecido. No entanto, voltou-se a testar o agente num ambiente mais complexo de modo a testar as diferenças em relação ao ambiente mais simples.

6.2 Caso experimental 2

Para este caso experimental utilizou-se outro tipo de ambiente mais complexo do que o anterior, com o intuito de analisar o comportamento do agente e como este lida com os aspectos inerentes a estes tipos de ambiente. O ambiente utilizado encontra-se ilustrado na figura seguinte.

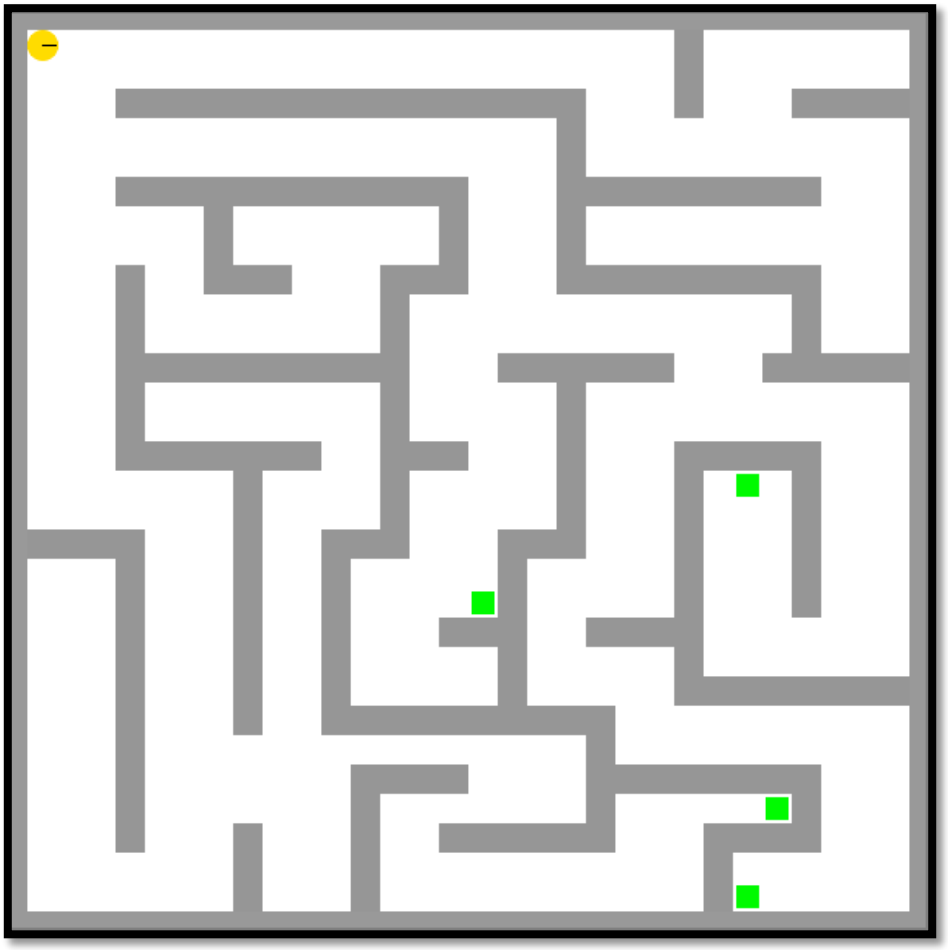


Figura 6.9 - Ambiente de teste para o caso experimental 2

A parametrização utilizada leva em conta a dualidade exploração *versus* aproveitamento pois para um ambiente complexo é importante que o agente tenha uma boa representação interna, derivada da exploração do ambiente, para que a camada deliberativa forneça uma resposta que seja, de facto, útil ao agente caso este esteja restrito à exploração de uma única zona. Assim, definiu-se inicialmente que, por cada passo, o agente iria apenas efectuar uma iteração para o cálculo da utilidade. Já o valor que corresponde ao erro máximo da utilidade (ε) e o factor de desconto (γ) foram combinados para que a política convergisse após o agente explorar o suficiente, de modo que a resposta possa auxiliar o comportamento resultante do agente. Assim, definiu-se que $\gamma = 0{,}95$ e $\varepsilon = 0{,}1$. Estes valores resultam numa convergência da política em 116 iterações (ver Tabela 6.2), valor este que corresponde ao mesmo número de passos no ambiente. Este processo permite lidar com as restrições temporais para que a camada deliberativa forneça uma resposta em tempo real.

Durante o teste do modelo proposto, verificou-se que à medida que o agente conhece o ambiente, o tempo de execução de cada passo do agente aumenta. Essa situação pode ser explicada dado o crescente número de estados à medida que o número de colisões aumenta. Isto ocorre porque o número de pontos relevantes aumenta, o que leva que a discretização não linear em *Quadtree* se torne mais detalhada para os pontos em questão (ver secção 5.5), aumentando o esforço computacional para o cálculo da distribuição probabilística dos estados sucessores no modelo PDM, uma vez que o cálculo da mesma está a ser efectuado, a cada iteração, para todos os estados do modelo.

Verificou-se, também, que caso o agente forneça, com alguma antecipação, uma acção antes de explorar o ambiente suficientemente, pode resultar em situações como a ilustrada na figura seguinte.

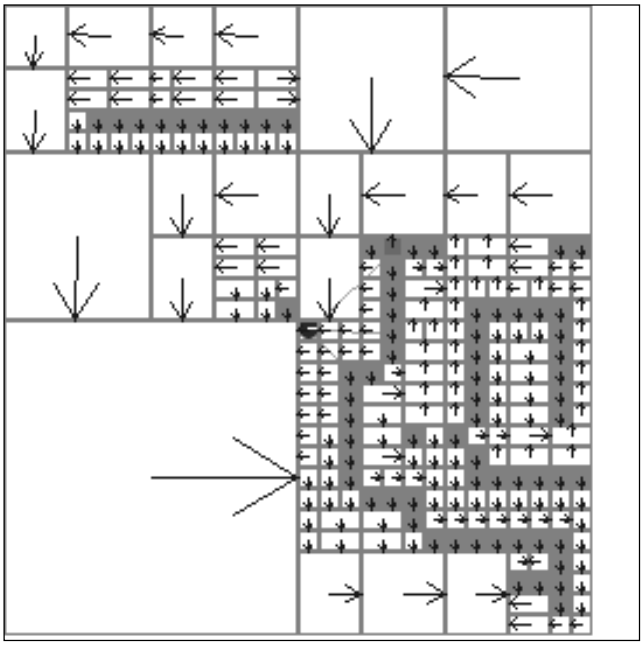


Figura 6.10 - Situação ocorrida com a falta de exploração do ambiente

A situação ilustrada na Figura 6.10, corresponde à representação de uma situação ocorrida após o cálculo da política comportamental, quando o agente ainda não tem detalhe o suficiente para gerar uma política consistente para uma parte do modelo. Isto é, o agente tem representado internamente um modelo cuja exploração para uma área não foi concretizada e por tal, a acção fornecida para essa área é de tal modo abstracta que leva o agente a cair em óptimos locais. Essa situação pode ser observada, no canto inferior esquerdo, onde a política

para o estado abstracto é ir para direita e a política no estado actual do agente é ir para a esquerda.

Em termos de não determinismo, a situação mantém-se em relação ao caso experimental anterior, isto é, o agente consegue lidar com o não-determinismo do ambiente mas verifica-se a existência de falsas classificações de obstáculos na representação interna.

Durante os testes verificou-se a existência de situações onde o comportamento de mais alto nível não é o comportamento mais apropriado. Mostrando-se menos viável do que a resposta das camadas abaixo. Dado que, a selecção da acção é feito por supressão, as camadas de mais alto nível devem actuar de forma mais precisa, e para tal é relevante conjugar a exploração e o aproveitamento da informação adquirida.

7 Conclusão

Os agentes inteligentes híbridos são um tipo de agente formado por duas camadas principais: a camada reactiva e a deliberativa. Para além das duas camadas, é possível ter mais camadas consoante as funcionalidades que se pretendem atribuir ao agente.

No âmbito desta dissertação propôs-se estudar uma forma de integrar a reacção e deliberação de agentes inteligentes, adicionando-se mais uma camada que se mostrou de grande relevância para o auxílio nas limitações da camada reactiva e para o apoio da camada deliberativa. Nesse sentido, surgiu a camada adaptativa que aprende informações directamente do ambiente e que as armazena em modelos que abstraem a complexidade deste.

Por norma, um agente inteligente híbrido necessita de responder a certas restrições de forma a poder ser utilizado em tempo real. Para isso é necessário que consiga lidar com recursos limitados e restrições temporais onde tais características a fazem diferenciar das restantes arquitecturas existentes ao longo da história da inteligência artificial.

A arquitectura proposta cumpre a maioria das características que foram colocadas como objectivo, uma vez que, se estudou através da análise de outras arquitecturas como integrar a reacção e deliberação num agente inteligente. Também possui as principais características de um agente inteligente híbrido, sendo organizado em camadas onde uma delas é a camada reactiva que permite responder rapidamente a alterações do ambiente e outra a deliberativa, que utiliza modelos internos para elaborar planos de mais alto nível para auxiliar o comportamento do agente. Esta camada tem uma camada de aprendizagem que discretiza o ambiente numa grelha linear bidimensional e armazena informação correspondente à função estado-acção, que é utilizada em termos de aprendizagem por reforço para se adaptar à nova informação aprendida.

No sentido de lidar com a complexidade espacial, a arquitectura proposta abstrai o ambiente em dois níveis que estão presentes na camada adaptativa e deliberativa. A primeira discretiza o ambiente numa grelha linear, e a segunda numa representação em *Quadtree*. A crescente aprendizagem faz com que o agente comece a armazenar muita informação do ambiente e por tal, possui um mecanismo de esquecimento que elimina informação considerada antiga e que já não possui relevância para a actuação do agente. No entanto, o agente armazena de forma

persistente a informação necessária para deliberar, uma vez que será utilizada para o processo de discretização de alto nível.

O tempo de resposta do agente é gerido pelas três camadas de forma diferente: a camada reactiva opera sob a influência de campos de potencial, pelo que tem sempre uma resposta a fornecer face as alterações do ambiente que originem campos de potencial. A camada adaptativa, fornece uma resposta aos estímulos após já conhecer algo do ambiente (obstáculos) de forma a não repetir erros. Já a camada deliberativa gera uma política comportamental através de Processos de Decisão de *Markov* e apenas responde quando já tem uma resposta para fornecer. Este último mecanismo, consome demasiados recursos computacionais para realizar o processamento em tempo-real e, por tal, a forma de ultrapassar essa restrição é atribuindo uma limitação no cálculo do PDM por cada passo do agente. Assim, o agente consegue responder às várias características do ambiente dentro das restrições impostas por um sistema que opere em tempo real.

Existe uma situação que limita a utilização deste sistema, que corresponde ao dinamismo. Este é lidado pela camada reactiva e adaptativa correctamente mas tem alguma limitação quando operado pela camada deliberativa uma vez que o custo de actualização do modelo deliberativo face as alterações do ambiente pode aumentar a carga de processamento deliberativo. Esta é uma área relevante para o trabalho futuro.

Para além da situação acima descrita, várias outras situações foram identificadas e que poderiam trazer ao agente benefícios em termos de tempo de resposta ou facilitar o processamento. Outras abordagens poderiam apenas mostrar um outro caminho que poderia ser seguido ao invés do caminho escolhido para esta dissertação. Essas alternativas são de seguida abordadas.

7.1 Trabalho futuro

7.1.1 *Quadtree* iterativa

No sentido de aumentar a rapidez no processo de criação da *Quadtree*, pensou-se numa forma de tornar o processo de criação mais eficiente. Actualmente uma nova *Quadtree* é criada por cada colisão que o agente efectua, o que para ambiente simples é um processo suportável em termos de consumo de recursos computacionais mas para ambientes cuja dimensão é elevada

esse processo poderá não ser o mais adequado. Assim, pensou-se na possibilidade de criar a *Quadtree* iterativamente, isto é, iniciar a *Quadtree* tendo apenas a informação relevante correspondente aos alvos e à medida que o agente colide com os obstáculos, fossem adicionados à *Quadtree* novos nós correspondente às novas áreas criadas surgidas pela expansão da área da *Quadtree*, ou pela divisão das áreas já existentes, dado a nova informação recebida.

Segundo *Samet* (1984), a *Quadtree* tem algumas limitações, nomeadamente, na organização da informação em comparação com uma árvore binária. No entanto, seria interessante estudar formas de ultrapassar algumas dessas limitações a fim de optimizar a procura e a inserção de nós, para que a *Quadtree* fosse sendo actualizada e expandida à medida que o agente fosse conhecendo o ambiente.

7.1.2 Aprendizagem em ambiente contínuo

Durante o desenvolvimento do trabalho pensou-se na utilização de funções de aproximação para efectuar uma aprendizagem do ambiente contínuo. Seria interessante efectuar uma aprendizagem semelhante à proposta por *Saito* (1994) através da utilização da rede neuronal CMAC[13] (Albus, 1975) para a generalização da função Q. Assim os parâmetros de entrada da função de aproximação seriam o estado e a acção, e a saída seria o valor Q expectável. A generalização da função Q poderia poupar memória e tempo na aproximação da função Q para estados não visitados.

Outros tipos de funções de aproximação podem ser utilizados tal como referidos, no trabalho desenvolvido por *Saito* (1994).

7.1.3 Optimizações de processamento deliberativo

Ao longo do desenvolvimento do modelo proposto foi identificada uma crescente lentidão no processamento do agente à medida que a complexidade da representação interna do ambiente aumentava. Após uma colisão, o agente cria uma representação em *Quadtree* e utiliza-a para a criação do modelo PDM. Por cada área é calculada a distribuição probabilística dos estados sucessores e sempre que se efectua um passo do agente um novo cálculo é efectuado. A ideia seria optimizar esse processo através da utilização de tabelas de procura que armazenariam os

[13] Cerebellar Model Articulation Controller

valores das probabilidades de transição já pré-calculadas permitindo que o processo de cálculo ocorresse apenas uma vez.

Outra situação que seria interessante, é a limitação do cálculo do PDM pelo número de estados. Actualmente por cada passo do agente, é calculada a utilidade para o espaço de estados num determinado número de vezes. Como trabalho futuro, propõe-se fazer o cálculo do PDM limitado a um determinado número de estados presentes no espaço de estados. Se o espaço de estados não tivesse o número máximo de estados a ser calculado então o cálculo seria para todo o espaço de estados. Caso contrário, o cálculo seria efectuado para alguns estados numa iteração e os restantes nas iterações seguintes.

Estas optimizações resultariam em melhorias significativas no processamento da camada deliberativa.

7.1.4 Outro mecanismo de raciocínio

Igualmente interessante seria utilizar outros mecanismos de raciocínio para a camada deliberativa, como por exemplo, a procura em espaço de estados. A procura em espaço de estados, utilizaria na mesma a discretização de alto nível para abstrair a complexidade do ambiente e gerar um plano a ser executado pelo agente, caso eventualmente o agente estivesse no ambiente estocástico, a resposta do agente seria uma das acções da camada hierarquicamente abaixo enquanto a camada deliberativa voltaria a reconsiderar as alterações do ambiente para geração de novo plano.

7.2 Considerações Finais

A implementação de agentes inteligentes capazes de formular planos e lidar com ambientes reais onde o tempo de resposta e a escassez de recursos são dois factores decisivos, foram alvos de estudo durante a década de 90 com o surgimento dos agentes inteligentes híbridos. Esses agentes são compostos por duas camadas que são responsáveis pela reactividade e pró actividade do agente. No entanto, a adaptabilidade esteve sempre muito associado à pró-actividade do agente, não sendo por isso vista como um nível intermédio entre as duas camadas, uma vez que, permite auxiliar tanto a camada reactiva como a deliberativa.

Nesse sentido, pretendeu-se mostrar nesta dissertação uma forma de integrar a reacção e a deliberação em agentes inteligentes e associar a aprendizagem como uma camada intermédia

para auxiliar as duas camadas reactiva e deliberativa, principalmente a camada reactiva por ser mais susceptível ao problema dos óptimos locais. Assim, procedeu-se à apresentação de uma proposta de um agente inteligente híbrido que possui um nível de competência destinado a lidar com o processo de aprendizagem do agente.

8 Bibliografia

ALBUS, J. *et al.* - **4D/RCS: A Reference Model Architecture For Unmanned Vehicle Systems**. Gaithersburg - Maryland : NIST - National Institute of Standards and Technology, 2002

ALBUS, J. S. - A New Approach to Manipulator Control : The Cerebellar Model Articulation Controller (CMAC). **Journal of Dynamic Systems, Measurement, and Control**. 1975. 220–227.

ARKIN, R. C. – **Behavior-Based Robotics**. Massachusetts : MIT Press, 1998.

BELLMAN, R. - The Theory of Dynamic Programming. July (1954) 27.

BIANCHI, R. A. C.; RIBEIRO, C. H. C.; COSTA, A. H. R. - Heuristically Accelerated Q – Learning : A New Approach to Speed Up Reinforcement Learning. In **XVII Brazilian Symposium on Artificial Intelligence - SBIA'04**. p. 245–254.

BROOKS, R. A. - **A robust layered control system for a mobile robot**. Massachusetts : Massachusetts Institute of Technology, 1986. 26 f.

BROOKS, R. A. - **Intelligence without representation**. Massachusetts : Massachusetts Institute of Technology, 1991. 12 f.

FERGUSON, I. A. - **TouringMachines: an architecture for dynamic, rational, mobile agents**. Cambridge : University of Cambridge, 1992. 219 f.

HUANG, H. - **RCS: The Real-time Control Systems Architecture** [Em linha]. atual. 24 Mar. 2011 [Consult. 14 Ago. 2014]. Disponível em WWW: <URL:http://www.nist.gov/el/isd/rcs.cfm>

LIND, J. - Patterns in Agent-Oriented Software Engineering. In GIUNCHIGLIA, F.; ODELL, J.; WEIS, G. (Eds.) - **Proceedings of the 3rd international conference on Agent-oriented software engineering III (AOSE'02)**. Bologna : Springer, 2002

MORGADO, L. F. G. - **Integração de Emoção e Raciocínio em Agentes Inteligentes**. Lisboa : Faculdade de Ciências da Universidade de Lisboa, 2005. 265 f.

MULLER, J. P. - The Agent Architecture - INTERRAP. In **The Design of Intelligent Agents - A Layered Approach**. Berlin : Springer, 1996. ISBN 978-3-540-62003-7. p. 45–

123.

MULLER, J. P.; PISCHEL, M. - **The Agent Architecture InteRRaP: Concept and application**. Saarbrücken : German Research Center for Artificial Intelligence, 1993. 109 f.

MURPHY, R. R. - **Introduction to AI robotics**. 1st. ed. Cambrigde, Massachusetts : A Bradford Book, 2000. 487 p. ISBN 0262133830.

PELLEGRINI, J.; WAINER, J. - Processos de Decisão de Markov : um tutorial. **RITA**. VIX:2 (2007) 133–179.

PUTERMAN, M. L. - **Markov Decision Processes:Discrete Stochastic Dynamic Programming**. New Jersey : John Wiley and Sons, 2005. 666 p. ISBN 0-471-72782-2.

RUSSELL, S.; NORVIG, P. - **Artificial Intelligence - A Modern Approach**. 3rd. ed. Upper Saddle River, New Jersey : Prentice Hall, 2010. 1152 p. ISBN 9780136042594.

SAITO, F.; FUKUDA, T. - **Learning architecture for real robotic systems-extension of connectionist Q-learning for continuous robot control domain**. Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on , pp.27,32 vol.1, 8-13 May 1994

doi: 10.1109/ROBOT.1994.351015

SAMET, H. - The Quadtree and Related Hierarchical Data Structures. **ACM Computing Surveys**. 16:2 (1984) 73.

SLACK, M. G. – **Situationally driven local navigation for mobile robots**. Virginia : Virginia Polytechnic Institute, 1990.

SUTTON, R. S.; BARTO, A. G. - Chapter 1 Introduction. In **Reinforcement Learning: An introduction**. 1st. ed. Massachusetts : A Bradford Book, 1998. p. 3–24.

SUTTON, R. S.; BARTO, A. G. - **Reinforcement Learning : An Introduction**. 2nd ed. London : The MIT Press, 2012. 334 p.

WATKINS, C. J. C. H. - **Learning from Delayed Rewards**. Cambridge : University of Cambridge, 1989. 241 f.

WOOLDRIDGE, M. - **An introduction to Multiagent Systems**. 1st. ed. Chichester, England : John Wiley and Sons, 2002. 340 p. ISBN 0 47149691X.

9 Anexos

9.1 Ambiente de Desenvolvimento

O processo de desenvolvimento desta dissertação foi totalmente auxiliada pela PSA[14] criada inicialmente na linguagem de programação *JAVA* pelo Professor Doutor *Luís Morgado* (2005, p.215) mas adaptada para a linguagem *Python* e disponibilizada desde 2012 pelo autor.

A PSA caracteriza-se por ser uma plataforma que permite:

- Adicionar uma implementação de agente (reactivo, deliberativo, aprendizagem por reforço ou híbrido);
- Definir várias configurações de ambiente;
- Parametrizar características associadas ao ambiente (dinamismo, não-determinismo, nível de detalhe, etc.);
- Visualizar, em tempo real, a evolução do agente.

Tais características permitiram que a implementação do protótipo proposto nesta dissertação fosse efectuada em módulos separados, para que no final pudessem ser combinados de forma a criar um agente inteligente híbrido.

A interface actual da PSA pode ser visualizada na figura seguinte.

[14] Plataforma de Simulação de Agentes

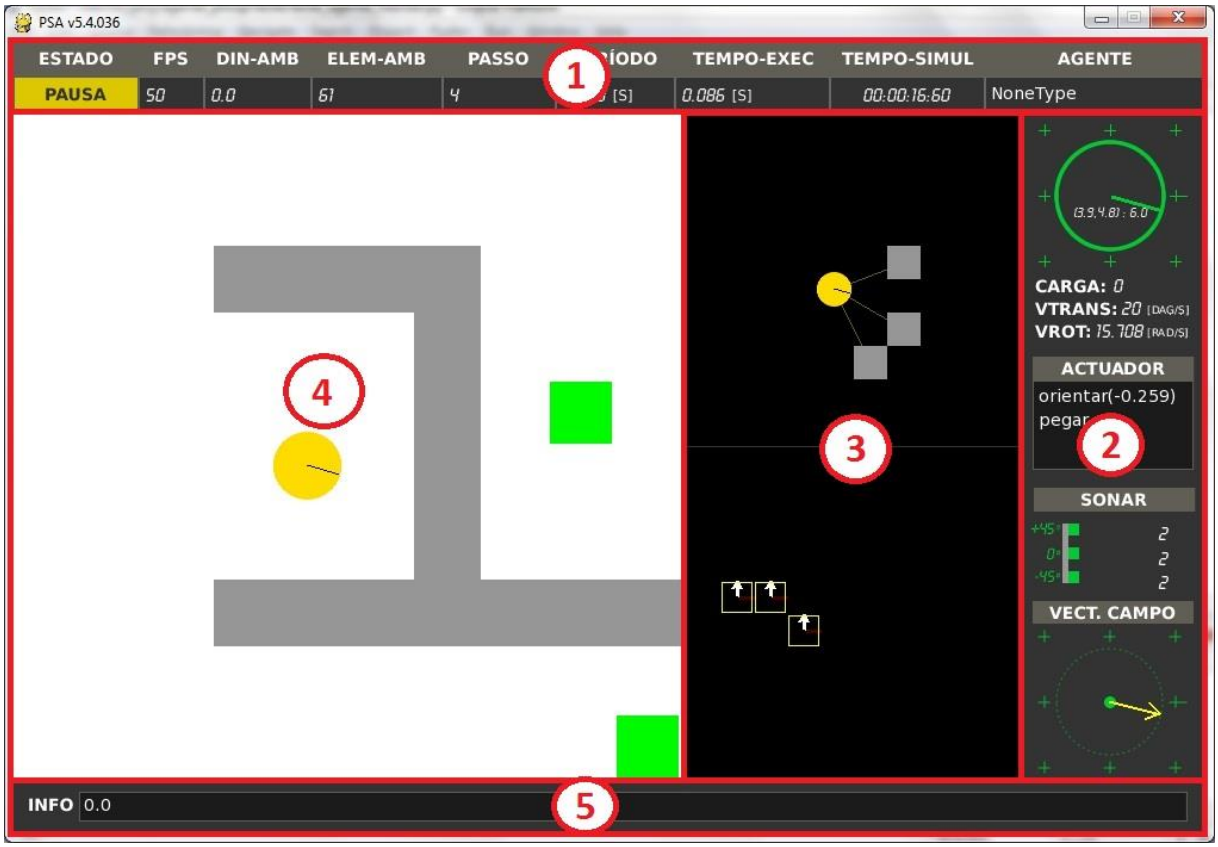


Figura 9.1 - Interface de visualização da PSA

A interface da PSA pode ser separada em cinco componentes principais, estas são:

1. Barra de informação do simulador;
2. Barra de informação do agente;
3. Visualizadores de modelo e percepção;
4. Visualizador do ambiente;
5. Barra de informação genérica.

Para a implementação deste protótipo, esta plataforma de simulação de agentes permitiu visualizar várias informações que ajudaram a interpretar os comportamentos resultantes. O componente 2 permitiu visualizar o vector resultante da soma dos alvos na camada reactiva no visualizador *Vect. Campo*. Já o componente 3 permitiu visualizar o modelo de aprendizagem e a política proveniente do Processos de Decisão de *Markov*, resultantes das camadas adaptativa e deliberativa, respectivamente. O componente 4 foi útil para visualizar e analisar o comportamento do agente. Por último, o componente 5 serviu para visualizar valores em eventuais situações de *debug*.