



## Capítulo 9

# 9. Coordenação em SMA no Futebol Robótico

O Futebol Robótico em geral e a liga de simulação em particular, constituem domínios particularmente adequados à aplicação da investigação realizada em metodologias de coordenação em Sistemas Multi-Agente, nomeadamente no que diz respeito à coordenação para agentes cooperativos.

A escolha do futebol robótico como domínio principal de teste para a maioria das metodologias desenvolvidas nesta tese, prende-se com as características do ambiente proporcionado pelo simulador *soccerserver*. O ambiente é inacessível, dinâmico, não determinístico, contínuo e impõe elevadas limitações às capacidades de comunicação dos agentes. Para além disso, a definição clara de duas equipas e uma noção inicial exacta da composição das equipas e da tarefa cooperativa a realizar por cada uma, tornam este ambiente um dos domínios por excelência para a aplicação das metodologias de coordenação de agentes cooperativos desenvolvidas, nomeadamente para a aplicação com sucesso da coordenação estratégica.

O facto de existir um agente privilegiado – o treinador – com uma visão global do ambiente, mas que não pode interferir regularmente no jogo, torna este domínio ideal para a aplicação das metodologias de coordenação por controlo parcialmente hierárquico desenvolvidas.

A disponibilidade de sensores configuráveis, nomeadamente de uma câmara virtual accionada por um pescoço flexível simulado, permite a utilização de metodologias de percepção inteligente.

A baixa largura de banda disponibilizada no domínio e as restrições impostas à comunicação, tornam o futebol robótico simulado particularmente atractivo para a aplicação de metodologias de coordenação por comunicação e de metodologias de coordenação por modelização mútua. Como não é possível comunicar uma parcela significativa da informação disponível, torna-se necessário criar modelos dos colegas de equipa e raciocinar utilizando a predição das suas acções.

Grande parte da investigação descrita neste capítulo foi desenvolvida no âmbito do projecto FC Portugal. Este é um projecto de cooperação entre a Universidade do Porto<sup>64</sup> e a Universidade de Aveiro<sup>65</sup> com o objectivo de desenvolver uma equipa de futebol robótico simulado capaz de participar nas competições RoboCup. Como resultado foi desenvolvido um sistema Multi-Agente totalmente funcional – FC Portugal – que participou com muito bons resultados em diversas competições internacionais. Este capítulo descreve o SMA FC Portugal<sup>66</sup> e a investigação que proporcionou a sua construção.

Os tópicos principais de investigação do projecto FC Portugal incluem:

- Metodologias de coordenação e cooperação em SMA;
- Metodologias de comunicação e percepção inteligentes em SMA;
- Arquitecturas de agentes inteligentes;
- Conceito de estratégia para uma competição;
- Representação de conhecimento
- Modelização de agentes e equipas de agentes;
- Definição de um treinador para competições de equipas de agentes;
- Optimização em SMA;
- Sincronização de processos em arquitecturas distribuídas;
- Desenvolvimento de ferramentas de depuração (“*debug*”) e análise de agentes.

A investigação descrita dá particular ênfase aos tópicos cuja investigação foi maioritariamente realizada pelo autor na Universidade do Porto<sup>67</sup>, nomeadamente ao desenvolvimento de metodologias eficientes de coordenação de equipas de agentes.

---

<sup>64</sup> LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto – NIAD&R - Núcleo de Inteligência Artificial Distribuída e Robótica da Faculdade de Engenharia da Universidade do Porto.

<sup>65</sup> IEETA – Instituto de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro.

<sup>66</sup> A descrição é centrada nos avanços de investigação realizados no âmbito do projecto, até Setembro de 2001. No entanto, alguns desenvolvimentos realizados durante o ano de 2002 são também resumidamente descritos.

<sup>67</sup> No entanto, contribuições de ambas as Universidades existem em quase todas as partes do projecto, sendo muito difícil separar as contribuições de cada Universidade.

## 9.1 Tipos de Coordenação Aplicáveis ao Domínio

O domínio do futebol robótico é, como foi discutido no capítulo anterior, um domínio extremamente complexo, dinâmico, parcialmente cooperativo e parcialmente adverso. As metodologias de coordenação a aplicar, de forma a construir boas equipas de futebol robótico simulado têm de reflectir as características do domínio.

Os tipos de coordenação aplicados ao domínio do futebol robótico no âmbito desta tese são as seguintes:

- **Coordenação por Comunicação.** Utilização de um protocolo de comunicação – ADVCOM – que permite coordenar a equipa através da partilha de informação e eventos úteis para a coordenação;
- **Coordenação por Percepção Inteligente.** Definição de um algoritmo de visão estratégica – *SLM – Strategic Looking Mechanism*;
- **Coordenação por Modelização Mútua.** Criação de modelos dos colegas de equipa e adversários e sua utilização no processo de coordenação;
- **Coordenação Estratégica.** Definição de uma organização da equipa que permite definir uma estratégia para um dado jogo composta por táticas com condições específicas de activação. Neste contexto, foram objecto de estudo:
  - Distinção entre situações activas e estratégicas no contexto do futebol robótico;
  - Metodologias para definir situações facilmente reconhecidas por todos os agentes;
  - Definição de papéis (tipos de jogadores) especificando o comportamento individual e colectivo dos jogadores a diversos níveis;
  - Definição de um mecanismo de posicionamento estratégico baseado em situações – *SBSP – Situation Based Strategic Positioning*;
  - Metodologias para efectuar a troca dinâmica de papéis e posicionamentos baseadas em utilidades – *DPRE – Dynamic Positioning and Role Exchange*;
- **Coordenação Parcialmente Hierárquica.** Utilização do agente treinador e de metodologias de integração dinâmica de instruções nos agentes jogadores. Neste âmbito foi também desenvolvida uma linguagem – *COACH UNILANG* – que permite a um agente treinador comunicar a estratégia para um determinado jogo à sua equipa e alterar essa estratégia no decurso do jogo;

Embora estes mecanismos de coordenação constituam a parte mais significativa da investigação realizada no âmbito do projecto FC Portugal. É de notar que muitas outras áreas foram abordadas neste projecto, de forma a constituir uma equipa de agentes

totalmente funcional capaz de participar nas competições internacionais RoboCup. Grande parte dos desenvolvimentos nesta área foram influenciados pelo futebol real e definidos tendo como base o contacto com especialistas em futebol real (treinadores, jogadores e jornalistas).

## 9.2 Influência do Futebol no Futebol Simulado

O futebol real tem uma grande influência no futebol robótico simulado e como tal a construção de uma boa equipa de futebol robótico simulado implica a inclusão de conhecimento específico do domínio. As metodologias de coordenação e decisão individual dos jogadores foram em grande parte definidas ou ajustadas por influência de conhecimento adquirido através do contacto com profissionais de futebol real: jornalistas, treinadores e jogadores. Este contacto levou à definição de 30 medidas para criar uma equipa de futebol robótico simulado [Reis, 2000] que constituíram a base para a definição das metodologias de coordenação e decisão individual utilizadas no projecto. A tabela 19 apresenta as 30 medidas definidas no início do projecto.

<i>Linhas de Orientação para Criar uma Equipa de Futebol Simulado</i>	
1) Colocar-se sempre numa posição do campo útil para a equipa.	16) Permitir que a bola trabalhe para ti e não o oposto.
2) Saber sempre com precisão a posição e velocidade da bola	17) Manter uma formação da equipa compacta.
3) Quando a bola estiver bem localizada e não for provável a alteração da sua trajectória, olhar para o campo localizando os colegas de equipa e adversários relevantes para a situação.	18) Usar o espaço livre para atacar e limitá-lo ao defender.
4) Não comunicar a não ser que saiba algo de importância para os colegas.	19) Quando a posse de bola é do adversário, todos os jogadores devem pensar “defensivamente”.
5) Comunicar sempre e de imediato eventos e informação importantes se achar que é um dos jogadores da equipa que é capaz de os descrever melhor.	20) Os avançados (e também os outros jogadores) devem trocar de posições sempre que tal for útil ou para confundir a defesa adversária.
6) Decidir acções prevendo o seu desenlace (imaginar possíveis cenários que advenham da sua execução).	21) Os extremos ofensivos da equipa devem permanecer abertos para atrair os defesas para longe do centro da área.
7) Em cada momento reduzir as opções possíveis dos oponentes e aumentar as opções possíveis dos colegas de equipa.	22) Desafiar (limitando-lhe o espaço de acção do oponente que tem a posse da bola rapidamente) para que não possa progredir de forma rápida no campo.
8) Para realizar diferentes acções colectivas no campo (atacar, defender, etc.) um comportamento de equipa diferente é necessário.	23) Limitar as opções do adversário, limitando-lhe o espaço e cobrindo as suas possíveis linhas de passe.
9) Não desperdiçar energia a realizar tarefas que não sejam importantes para a equipa.	24) Quanto mais próximo está a bola da baliza da equipa, mais apertada deve ser a defesa.
10) Tentar colocar-se no interior do cérebro dos colegas e adversários e tentar imaginar o que eles estão a pensar. Utilizar essa informação para decidir as acções.	25) Como defesa não fazer passes arriscados nem tentar intercepções arriscadas. Deixar o atacante tomar a iniciativa.
11) Se puder marcar, chutar! Senão, passar a alguém que possa marcar. Senão, passar a alguém que possa passar a alguém que possa marcar.	26) O que se faz sem bola é tão importante como o que se faz com bola. Definir formas de movimentação sem bola adequadas.

Senão...	
12) Não segurar a bola a não ser que não exista outra acção para fazer.	27) Utilizar jogadores com diferentes capacidades para diferentes posições no campo.
13) Não passar a bola para trás.	28) Mudar o estilo de jogo de acordo com o oponente e com o curso do jogo.
14) Sempre que possível passar a bola para jogadores que a possam receber atrás da defesa do adversário.	29) Ser flexível e adaptativo. Não utilizar regras rígidas para nada.
15) Cruzar a bola dos flancos para o centro da área sempre que possível.	30) Treinar contra adversários com capacidades e estratégias distintas e analisar em profundidade os jogos.

*Tabela 19: Trinta Linhas de Orientação para criar uma equipa de futebol simulado*

As duas primeiras medidas são essenciais e permitem por si só desenvolver uma equipa competitiva de futebol simulado. Aliás, baseado nestas duas medidas foi desenvolvido o código FCP Agent [Reis e Lau, 2001d] que, unicamente com 250 linhas de código (que assentam sobre o código de baixo nível do CMUnited99 [Stone et al., 1999]), permite construir uma equipa competitiva para a liga de simulação. Estas 250 linhas de código implementam um esquema de posicionamento muito simples (baseado numa simplificação do posicionamento estratégico utilizado pelo FC Portugal) que implementa a medida 1. Implementam também um esquema de comunicação muito simples (em que é comunicada sempre a posição e velocidade da bola por jogadores que a saibam) que implementa a medida 2. No entanto, para desenvolver uma equipa capaz de jogar futebol de forma semelhante à de uma equipa de futebol real, as outras 28 linhas de orientação são também necessárias.

Analisando as medidas propostas é evidente que algumas são extremamente complexas de alcançar utilizando uma equipa de agentes autónomos. Mesmo com uma equipa de futebol real é extremamente difícil treinar os seus jogadores para que cumpram com eficácia todas estas medidas. Diversas metodologias de inteligência artificial, agentes inteligentes e principalmente de Sistemas Multi-Agente foram desenvolvidas de forma a permitir construir uma equipa de agentes que siga de forma eficaz as 30 medidas sugeridas.

## 9.3 Arquitectura dos Agentes

### 9.3.1 A Arquitectura *Strategic Agent*

Para construir agentes adequados à execução de uma determinada tarefa é necessário definir uma arquitectura apropriada para esses agentes. No âmbito do futebol robótico simulado e tendo em conta a discussão apresentada em secções anteriores, relativamente às características do domínio, da equipa de agentes e da tarefa cooperativa a realizar, é necessário construir agentes capazes de balancear a reactividade com a capacidade de deliberação individual e social. Por outro lado, a inclusão de capacidades de coordenação avançadas e conhecimento específico do futebol nestes agentes, leva à necessidade de

inclusão de módulos capazes de lidar com estas metodologias de coordenação e conhecimento específico. A arquitectura dos agentes do FC Portugal designada por *Strategic Agent* – Agente Estratégico, encontra-se representada na figura 59.

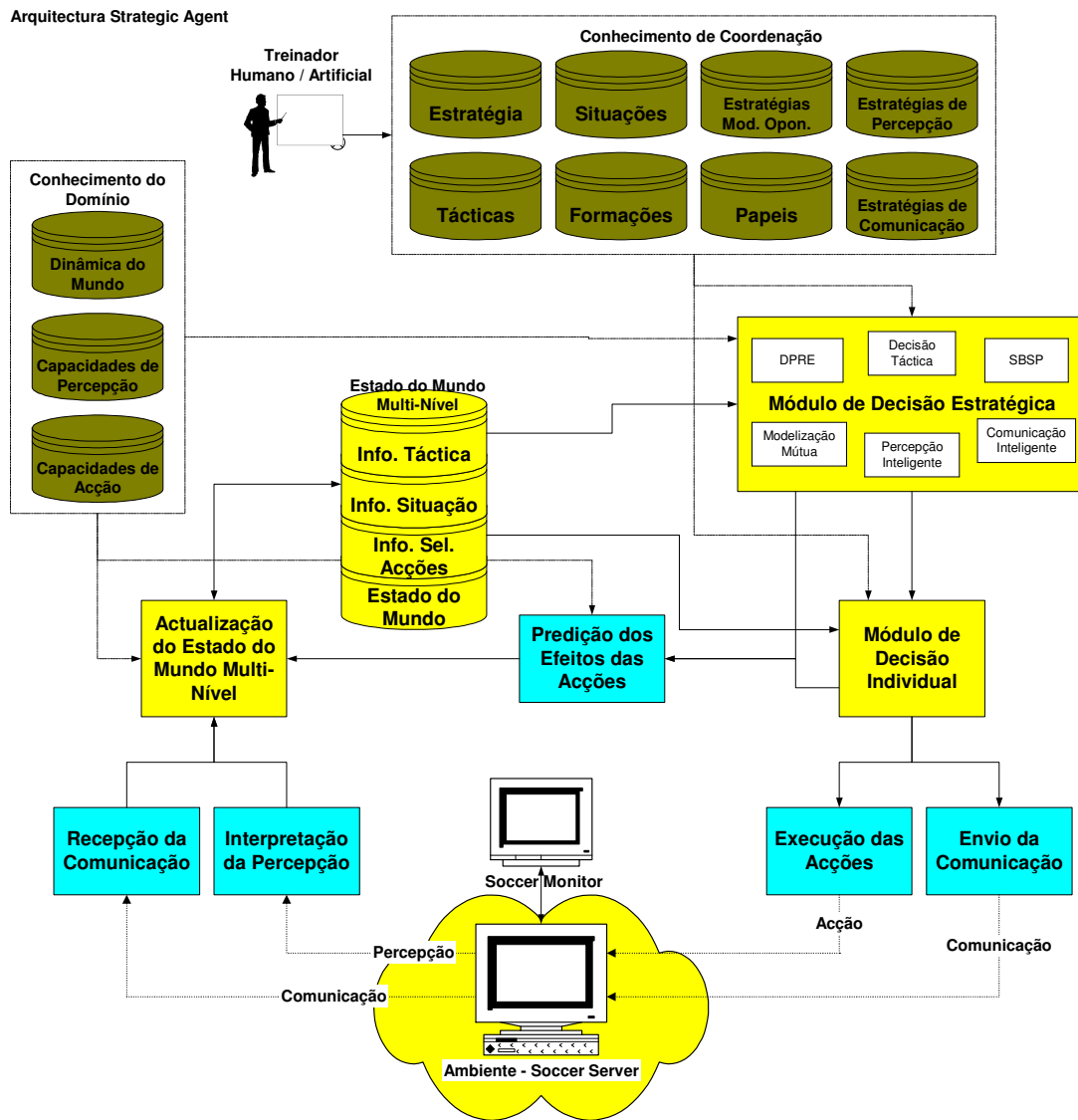


Figura 59: Arquitectura Strategic Agent dos Agentes FC Portugal

A arquitectura *Strategic Agent* centra-se na separação do conhecimento necessário em duas estruturas independentes: o conhecimento de coordenação e o conhecimento do domínio. No conhecimento do domínio é incluída a dinâmica do mundo, mais precisamente a parte desta dinâmica conhecida à partida pelo agente e as capacidades de percepção e acção do agente. No conhecimento de coordenação é incluída a estratégia a seguir, as tácticas disponíveis, as situações, formações e papéis que irão permitir utilizar a coordenação estratégica. São ainda incluídas estratégias de modelação de oponentes, de

comunicação e de percepção inteligente que irão permitir utilizar coordenação por modelização mútua, comunicação e percepção inteligente.

O estado do mundo de um agente estratégico é definido como uma estrutura multi-nível contendo desde informação de baixo-nível (incluindo as posições, velocidades e orientações dos objectos presentes no mundo) até informação de nível estratégico (incluindo informação temporal estratégica que irá permitir a selecção de táticas a utilizar). Este estado do mundo é actualizado utilizando a informação proveniente da percepção do agente, das comunicações recebidas, da dinâmica do mundo e da predição dos efeitos das suas acções e das acções dos outros agentes.

O módulo de decisão estratégica é responsável por seleccionar, em cada instante, a tática a utilizar e definir qual a situação de jogo, e de acordo com estas informações, definir qual a formação a utilizar e o consequente papel do agente e posicionamento estratégico. Este módulo é também responsável pelas decisões de comunicação e utilização dos sensores configuráveis do agente (percepção inteligente). Finalmente, é ainda este módulo que realiza a predição das acções dos colegas de equipa e dos oponentes, de acordo com a informação disponível no estado do mundo e com a informação disponível no conhecimento de coordenação relativo à modelização de colegas e oponentes.

Quando um agente se encontra em situação activa, a decisão do módulo estratégico é complementada com a decisão individual. Este módulo de decisão individual parameteriza a decisão estratégica utilizando conhecimento específico do domínio e informação contida no estado do mundo de baixo nível do agente. É também responsável pelas decisões específicas do domínio que, no caso do domínio em análise, estão relacionadas, por exemplo, com a decisão de a quem passar a bola, como driblar com a bola, para onde a chutar, como interceptar a bola, entre outras.

As decisões individuais são depois transformadas em comandos a enviar para os actuadores, que no domínio em análise correspondem aos comandos a enviar para o simulador. Esta transformação envolve, entre outras, questões de sincronização com o simulador, encriptação e compressão de informação e gestão física e monitorização dos meios de comunicação disponíveis.

### **9.3.2 Estado do Mundo Multi-Resolução**

Para que um agente se comporte de forma inteligente, é muito importante que este possua um modelo do estado do mundo que descreva o estado corrente do seu ambiente. O agente pode então utilizar este modelo para raciocinar sobre a melhor acção a executar em cada instante.



### 9.3.2.1 Características da Simulação

No futebol robótico simulado, os agentes recebem, no início do jogo, um conjunto de informações do servidor relacionado com a dinâmica do ambiente e com as capacidades de percepção e acção de cada um dos tipos de agente envolvidos no jogo. Esta informação corresponde aos parâmetros utilizados pelo *soccerserver* para executar a simulação (ver capítulo oito), incluindo: as dimensões do campo e das balizas, localização dos marcadores do campo (que irão permitir aos agentes realizar a sua auto-localização), características físicas e dinâmica da bola, características do terreno e vento no recinto, características da competição (número de partes, tempo máximo de jogo, ocorrência ou não de prolongamento e *penalties* em caso de empate) e características físicas dos jogadores (dimensões, momento máximo do pescoço, área de chute, ângulo de visão, gestão da energia, etc.).

Os agentes recebem também informação relativa aos parâmetros dos jogadores heterogéneos, incluindo: intervalos de variação dos parâmetros e definições específicas das características de cada um dos tipos de jogador heterogéneo. Todas as informações descritas são armazenadas em estruturas próprias (objectos, vectores e matrizes de diversos tipos) contidas no conhecimento específico do domínio e não são alteradas no decurso do jogo. Nestas estruturas são também armazenados os dados referentes ao número do jogador, equipa em que joga (*left* ou *right*) e nome das equipas.

### 9.3.2.2 Estado do Mundo

Durante o jogo, os jogadores recebem informação unicamente através de mensagens, mensagens estas que contêm percepção visual, sensorial e auditiva (proveniente do árbitro, treinador ou restantes jogadores). Esta informação, em conjunto com o conhecimento específico do domínio, permite a criação e actualização da estrutura contendo o estado do mundo multi-nível.

O modelo criado para os agentes FC Portugal implementados no âmbito desta tese, inclui uma representação probabilística do estado do mundo, baseada na especificação dos parâmetros de simulação e no historial de percepções do agente. Relativamente a cada jogador das outras equipas, os agentes mantêm no seu estado do mundo as posições ( $x$ ,  $y$ ), velocidades ( $v_x$ ,  $v_y$ ), orientações dos corpos e pescoços e valores relativos à confiança nesses valores  $[0..1]$  e precisão  $[0..1]$  esperada dos mesmos (figura 60).

A confiança está directamente relacionada com o instante de tempo em que o jogador foi visualizado pela última vez (o valor do tempo incluído na mensagem com informação visual, que continha esse jogador). Se a estimativa da posição do objecto foi realizada com base em percepções recentes, a confiança será elevada. O estado do mundo dos agentes é actualizado sempre que nova informação visual é disponibilizada. A posição e velocidade dos objectos e jogadores que não se encontram no campo visual do agente é também actualizada, tendo como base o seu movimento previsto ou esperado. Para estes, a

confiança na sua posição e velocidade é diminuída em cada ciclo em que não se encontrem visíveis.

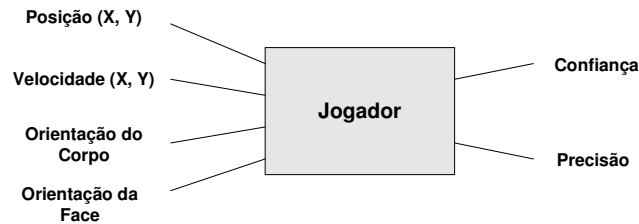


Figura 60: Informação no Nível Inferior do Estado do Mundo Relativa aos Outros Jogadores

A precisão dos valores está relacionada com a qualidade da observação dos mesmos tendo em conta o modelo do ambiente no que diz respeito aos erros de percepção. Observações de objectos que se encontram próximos do agente serão mais precisas do que observações de objectos distantes. Estimativas da posição ou orientação de objectos são também consideradas menos precisas do que observações directas desses objectos. Informação comunicada possui a precisão atribuída originalmente pelo agente que a comunicou.

As informações relativas às posições dos diversos objectos no campo são armazenadas num esquema de coordenadas absolutas  $(x, y)$ , como é indicado na figura 61. O centro do sistema de coordenadas corresponde ao centro do campo, com eixos  $x$  e  $y$ , ao longo do comprimento do campo (i.e. através do centro das balizas) e ao longo da linha central do campo. O sistema de coordenadas é definido de forma a que o sentido positivo do eixo dos  $xx$  seja no sentido atacante da equipa, independentemente de a mesma jogar da esquerda para a direita ou da direita para a esquerda. Os ângulos estão definidos no intervalo  $[-180, 180]$  de forma a que o ângulo  $0^\circ$  se encontra na direcção positiva do eixo dos  $x$ .

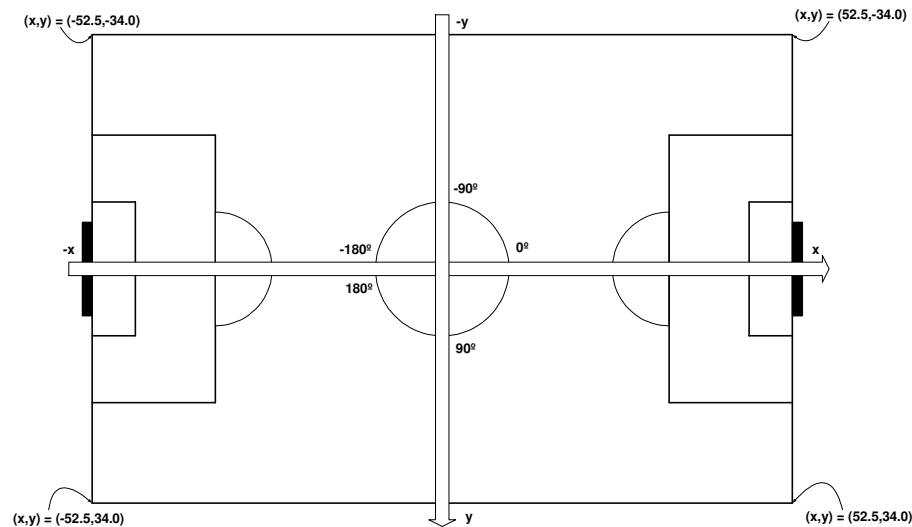


Figura 61: Coordenadas Absolutas e Ângulos do Campo de Jogo

Um aspecto importante a ter em consideração é que a informação que o agente recebe por via da sua percepção visual, é sempre relativa à perspectiva do agente no campo. Como resultado, o agente não pode determinar directamente, por observação, a sua posição absoluta no campo. A informação relativa tem de ser convertida para coordenadas absolutas no campo, mas esta conversão só será válida se a informação relativa ao próprio agente (sua posição, velocidade, orientação do corpo e pescoço) for válida e precisa.

Os agentes usam também a comunicação com o intuito de trocar informação sobre os seus estados do mundo, aumentando a precisão dos estados do mundo dos colegas de equipa. Embora em 2000 e 2001, fosse possível efectuar a troca de grandes quantidades de informação (512 bytes por mensagem), as novas regras introduzidas em 2002 reduziram a dimensão das mensagens para 10 bytes, não permitindo desta forma uma troca de informação completa e precisa acerca dos estados do mundo dos diferentes agentes. O protocolo de comunicação utilizado para este fim será analisado na secção 9.7.1.

A conversão da informação visual, relativa ao agente, para coordenadas absolutas é realizada com o auxílio dos diversos marcadores (bandeiras, linhas e balizas) colocados no campo e cuja posição é fixa e conhecida *a priori* (ver figura 56). Estes marcadores podem ser utilizados para permitir ao agente efectuar a sua auto-localização. Combinando as posições relativas dos marcadores visíveis, que o agente recebe como informação visual, com as suas posições absolutas conhecidas *a priori*, é possível ao agente determinar a sua própria localização ( $x,y$ ) e orientação. O método utilizado para realizar esta tarefa foi adaptado de [Stone et al., 1999]. Consequentemente, a posição e orientação do agente pode ser utilizada de forma a determinar as posições e orientações absolutas de todos os outros agentes.

### 9.3.2.3 Estrutura do Estado do Mundo Multi-Resolução

Uma das características mais relevantes da arquitectura apresentada na secção anterior consiste na inclusão de um estado do mundo multi-nível (figura 62).

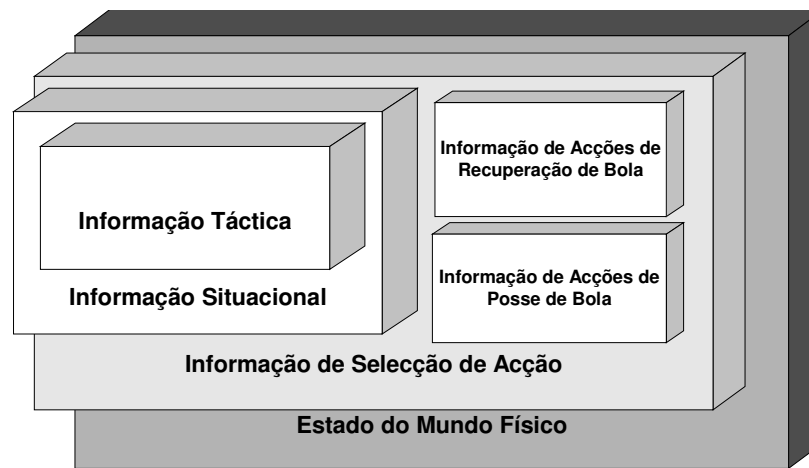


Figura 62: Estado do Mundo Multi-Nível dos Agentes FC Portugal

Como é visível na figura 62, o estado do mundo é separado em quatro níveis de abstracção [Reis et. al., 2001]:

- **Informação Tática:** Informação de alto-nível necessária para seleccionar e configurar a melhor tática em cada momento. Para além do tempo e resultado de jogo, inclui o comportamento global da equipa adversária (qualidade, capacidades colectivas e individuais, agressividade, etc.) e estatísticas de alto-nível do jogo (remates, assistências, posse de bola por região do campo, recuperações e perdas de bola por região do campo);
- **Informação Situacional:** Informação relevante para a selecção da situação de jogo e correspondente formação e ainda para a configuração do mecanismo de posicionamento estratégico baseado em situações. Esta informação está essencialmente relacionada com o estado actual do jogo e tem como objectivo ser suficiente para permitir realizar uma análise de alto-nível do momento actual;
- **Informação de Selecção de Acção:** Informação relevante para seleccionar uma acção apropriada, tal como distâncias à bola e possibilidades (tempos) de interceptação de bola de todos os jogadores, opções de movimentação (dos jogadores) e situações de “fora-de-jogo”. Inclui ainda a informação relativa a todas as acções com bola possíveis a um dado jogador, tais como os parâmetros relativos a cada passe, chuto ou drible possíveis.
- **Estado do Mundo Físico:** Informação de baixo-nível, incluindo o estado do agente e as posições, velocidades e orientações de todos os objectos presentes no ambiente.

A informação do estado do mundo multi-nível é actualizada, em cada ciclo, partindo da informação visual, sensorial e auditiva disponível e da predição dos efeitos das acções dos outros agentes. Inicialmente, é actualizado o estado do mundo físico e, baseado neste, é actualizada a informação de selecção de acção, informação situacional e informação tática.

Devido à extensão da informação contida no estado do mundo multi-nível (cerca de 350 itens de informação distintos) a sua descrição completa não é possível neste trabalho. A descrição detalhada do conteúdo de cada um dos níveis do estado do mundo multi-nível do FC Portugal pode ser encontrada em [Reis e Lau, 2001d].

#### 9.3.2.4 Actualização do Estado do Mundo

A actualização do estado do mundo é realizada utilizando a informação proveniente dos vários sensores do agente. Estes incluem a audição (mensagens dos colegas de equipa incluindo informação relativa às posições, velocidades, etc., dos diversos agentes e da bola), a visão e a percepção *sense\_body*. É também utilizada informação proveniente da

predição dos efeitos das acções do agente sobre o ambiente e dos efeitos das acções esperadas dos outros agentes sobre o ambiente.

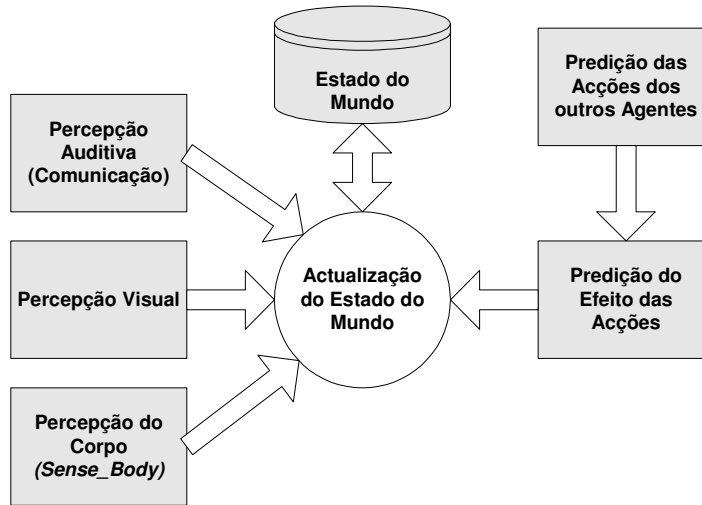


Figura 63: Atualização do Estado do Mundo

A atualização do estado do mundo multi-nível dos agentes é realizada da forma indicada no algoritmo 4. Consiste na atualização sucessiva das diversas camadas do estado do mundo multi-nível baseada na informação atualizada das camadas anteriores. Inclui ainda a criação e atualização de um histórico do estado do mundo.

```

ALGORITMO RefreshWorldState(Perceptions, MultiLevelWS, WSHistory)
    RETORNA MultiLevelWS, WSHistory
PARÂMETROS
    Perceptions - Percepção visual, auditiva e física do jogador
    MultiLevelWS - Estado do Mundo Multi-Nível do agente
    WSHistory - Historial do estado do mundo
{
    // Decompõe o MultiLevelWorldState nas suas componentes
    (WorldState, ActionWorldState, SituationWorldState, TacticalWorldState) = MultiLevelWS
    // Atualiza o estado do mundo de baixo-nível contido em MultiLevelWorldState
    WorldState = RefreshLowLevelWorldState(WorldState, Perceptions, WSHistory)
    // Atualiza o estado do mundo de selecção de acções (com e sem bola)
    ActionWorldState = RefreshActionWorldState(ActionWorldState, WorldState, WSHistory)
    // Atualiza o estado do mundo situacional
    SituationWorldState = RefreshSituationWorldState(SituationWorldState,
        ActionWorldState, WorldState, WSHistory)
    // Atualiza o estado do mundo táctico
    TacticalWorldState = RefreshTacticalWorldState(TacticalWorldState,
        SituationWorldState, ActionWorldState, WorldState, WSHistory)
    // Atualiza o histórico do estado do mundo contendo a informação do ciclo anterior
    WorldStateHistory = UpdateWorldStateHistory(MultiLevelWS, WSHistory)
    RETORNA MultiLevelWS, WSHistory
}

```

Algoritmo 4: Atualização do Estado do Mundo Multi-Nível

O estado do mundo de baixo nível é atualizado (algoritmo 5) baseia-se na utilização da informação sensorial e da predição dos efeitos das acções executadas pelo agente para

actualizar a camada inferior do estado do mundo (contendo as posições, velocidades e orientações de todos os objectos do campo). Inclui também uma filtragem do estado do mundo de forma a evitar variações bruscas inesperadas dos valores do mesmo.

```

ALGORITMO RefreshLowLevelWorldState(WorldState, Perceptions, WSHistory)
  RETORNA WorldState
PARÂMETROS
  Perceptions - Percepção visual, auditiva e física do jogador
  WorldState - Estado do Mundo do Agente
{
//Actualiza o estado do mundo considerando as percepções disponíveis e a predição do
efeito das acções do agente
  WorldState = RefreshLowLevelWorldState_SenseBody(WorldState, Perceptions)
  WorldState = RefreshLowLevelWorldState_Perceptions(WorldState, Perceptions)
  WorldState = RefreshLowLevelWorldState_Communications(WorldState, Perceptions)
  WorldState = RefreshLowLevelWorldState_Prediction(WorldState, Perceptions)
//Filtra o estado do mundo utilizando a informação dos ciclos anteriores
  WorldState = FilterWorldState (WorldState, WSHistory)
  RETORNA WorldState
}

```

*Algoritmo 5: Actualização do Estado do Mundo de Baixo-Nível*

A actualização do estado do mundo para decisão das acções a executar é realizada tendo em consideração se o agente controla a bola ou não. O algoritmo 6 descreve o método utilizado para fazer esta actualização. No anexo 5 são apresentados os algoritmos utilizados para actualizar a informação do estado do mundo quando o agente tem a posse de bola e quando não tem.

Se o agente tem o controlo de bola, a informação específica para cada acção possível de executar (remate, passe, drible, etc.) é actualizada. Se o agente não controla a bola, as informações de alto-nível relativa ao jogador e à bola, são actualizadas. Por exemplo, para a bola, esta informação de alto-nível inclui, entre outros itens: se a bola mudou de posição repentinamente, se a bola foi chutada, se a bola se desloca rapidamente, se a bola está livre no campo, se a bola foi centrada para a área, se foi chutada para a baliza ou se a bola está controlada por algum jogador.

A actualização do estado do mundo de acção prossegue com o cálculo de distâncias, tempos e pontos de intercepção de todos os jogadores à bola. Em seguida, é fundida informação de alto-nível comunicada pelos restantes elementos da equipa (por exemplo, relativa a tempos de intercepção, posse de bola, etc.). O algoritmo finaliza, através da análise de informação regional como sejam as linhas de fora de jogo e congestionamento<sup>68</sup> de regiões específicas do campo.

---

<sup>68</sup> Congestionamento corresponde à ocupação da região do campo por jogadores adversários. O número de jogadores e a sua proximidade à região considerada incrementam a sua congestão.

```

ALGORITMO RefreshActionWorldState(ActionWorldState, WorldState, WSHistory)
    RETORNA ActionWorldState
PARÂMETROS
    WorldState - Estado do Mundo do Agente
    ActionWorldState - Estado do Mundo para Selecção de Acções do Agente
{
// Actualiza parâmetros do estado do mundo necessários quer o agente tenha ou não a posse
de bola
    ActionWorldState = RefreshCommonActionWorldState (ActionWorldState, WorldState)
// Se o agente puder controlar a bola então actualiza o estado de mundo relacionado com a
posse de bola senão actualiza o estado do mundo de recuperação da bola
    SE PosseBola ENTÃO
        ActionWorldState = RefrescaBallPossessionWS (ActionWorldState, WorldState)
    SENÃO ActionWorldState = RefrescaBallRecoveryWS (ActionWorldState, WorldState)
// Aplica operações de filtragem ao estado do mundo, tendo em conta a informação histórica
do mesmo de forma a reduzir a influência dos erros neste
    ActionWorldState = FilterActionWS (ActionWorldState, WorldState, WSHistory)
    RETORNA ActionWorldState
}

```

Algoritmo 6: Actualização do Estado do Mundo de Acção

A descrição completa deste algoritmo está para além do âmbito deste trabalho pois o algoritmo completo de actualização do estado do mundo da equipa FC Portugal, implementado em linguagem C++ possui cerca de 3500 linhas de código. Uma descrição mais detalhada do algoritmo de actualização do estado do mundo multi-nível pode ser encontrada em [Reis e Lau, 2001d].

### 9.3.2.5 Predição dos Efeitos das Acções

Os métodos de predição destinam-se a prever o estado futuro do ambiente baseado em percepções passadas. Dividem-se em sete grupos principais:

- **Métodos para prever o estado do agente após executar uma determinada acção.** Por exemplo a execução de um *dash*, implicará o agente alterar a sua velocidade e um *turn* alterar a sua orientação absoluta.
- **Métodos para prever o estado futuro da bola.** No caso da bola não ser chutada a previsão é simples bastando diminuir sucessivamente a sua velocidade e alterando a sua posição em conformidade. A execução de um *kick* terá como efeito alterar a velocidade da bola.
- **Métodos para prever o estado futuro dos jogadores.** Supondo conhecidos os comandos a executar por um dado jogador, prever o seu deslocamento no campo.
- **Métodos para prever o tempo que um agente demora a atingir uma dada posição.** Supondo que o agente irá executar a sequência óptima de comandos, prever o tempo (ciclos) que o agente demorará a atingir uma determinada posição.

- **Métodos para prever o tempo que um agente demora a interceptar a bola.** Supondo que um agente irá executar a sequência óptima de comandos, prever em quantos ciclos consegue interceptar uma bola em movimento e em que ponto do campo será efectuada essa intercepção.
- **Métodos para prever as acções executadas por um dado agente nos próximos ciclos.** Estes métodos permitem prever quais as acções que cada agente irá executar no campo e alterar a informação desse agente utilizando os métodos descritos anteriormente que prevêm o efeito dessas acções.
- **Métodos para prever o posicionamento estratégico dos agentes de uma dada equipa.** Tendo em conta as posições ocupadas pelos agentes em situações anteriores do jogo, prever a posição dos jogadores adversários em cada situação.

Estes métodos constituem a base da criação do estado do mundo de alto-nível, baseado no estado do mundo físico. A sua descrição mais detalhada pode ser encontrada em [Reis e Lau, 2001d].

## 9.4 Comportamentos de Baixo-Nível (*Low-Level Skills*)

A disponibilização de metodologias de coordenação poderosas, não significa por si só a criação de uma equipa poderosa de futebol robótico simulado. As percepções e acções básicas disponíveis aos agentes são de muito baixo-nível, pelo que é necessário criar não só formas de transformar a percepção num estado do mundo de alto-nível (como foi analisado na secção anterior), como também formas de transformar decisões de alto nível em comandos básicos a enviar para o servidor. Este processo designa-se vulgarmente por criação de *low-level skills* (comportamentos de baixo-nível).

No âmbito do projecto FC Portugal, foram inicialmente utilizados os *low-level skills* disponibilizados no código fonte da equipa CMUnited, disponibilizado na Internet [Stone et al., 1999]. Aliás, o FC Portugal utilizou esses mesmos *low-level skills*, sem qualquer alteração significativa, no primeiro campeonato da Europa em que participou (Amesterdão 2000). Entretanto, após ter-se verificado que estes *low-level skills* já não estavam ao nível das melhores equipas mundiais, *low-level skills* completamente diferentes foram desenvolvidos no âmbito do FC Portugal para o campeonato do mundo disputado em Melbourne em 2000. Infelizmente, com a alteração das regras da competição e do simulador, o desenvolvimento de *low-level skills* tem de ser um processo em constante execução. Devido ao ênfase da investigação do FC Portugal não ser neste tópico, actualmente a equipa não possui *low-level skills* competitivos com as melhores equipas mundiais.



Nesta secção são descritas, de forma resumida, as principais especificidades dos *low-level skills* criados no âmbito do FC Portugal. Uma descrição mais detalhada de todos os *low-level-skills* pode ser encontrada em [Reis e Lau, 2001d].

### 9.4.1 Movimentação Evitando Adversários

Uma das habilidades básicas consiste no movimento de um agente de um ponto inicial até um ponto destino, evitando colidir com os adversários que se encontrem no caminho. No anexo 6 é apresentado o algoritmo de movimentação, evitando colidir com adversários, utilizado no decurso normal do jogo, pelos agentes implementados no âmbito desta dissertação.

O algoritmo baseia-se em calcular a distância e ângulo para o ponto de destino. Se o ponto de destino tiver sido atingido então o agente roda para a direcção final desejada e finaliza. Se for colidir com um jogador, modifica o destino momentaneamente de forma a contorna-lo. Dependendo da distância ao destino e da situação de jogo e posição no campo calcula o erro máximo no ângulo de orientação do jogador para o ponto de destino. O erro admissível será tanto maior quanto maior for a proximidade do jogador ao ponto de destino. Se o ângulo para o ponto de destino é maior do que o erro máximo admissível então roda para o ponto de destino senão acelera para o ponto tendo cuidado com a energia.

O algoritmo descrito encontra-se bastante simplificado relativamente ao utilizado na prática pelo FC Portugal (que contém diversas heurísticas específicas do domínio), apresentando-se unicamente a estratégia de alto-nível utilizada na movimentação. Detalhes sobre a implementação prática incluindo as funções para detectar colisões com adversários, tornejar adversários, calcular a direcção final desejada para o jogador, gerir a energia do jogador e ajustar de forma automática, dependendo da situação de jogo e tipo de movimento, o ângulo de erro, podem ser encontradas em [Lau e Reis, 2001d].

Para o guarda-redes foi desenvolvida uma habilidade básica de movimentação mais complexa utilizando não só movimentos para a frente mas também movimentos para trás e mantendo sempre (que possível) o contacto visual com a bola [Lau e Reis, 2001d].

### 9.4.2 Intercepção da Bola

A intercepção da bola é uma das habilidades críticas no futebol robótico simulado. No âmbito do projecto FC Portugal foram desenvolvidos algoritmos de intercepção segura de bola de modo a possibilitarem a elaboração de metodologias de decisão individual de mais alto-nível [Lau e Reis, 2002]. No anexo 7 é apresentada a metodologia seguida para realizar a intercepção de bola pelos agentes da equipa FC Portugal [Lau e Reis, 2001]. Este algoritmo verifica se é impossível a intercepção ou se esta foi já concluída, casos em que termina. Se a bola estiver parada então o jogador movimenta-se para o ponto onde esta

se encontra. Senão, testa diversas formas de interceptação utilizando comandos *dash* (preferencialmente) e *turn* e retorna qual o próximo comando a realizar. Detalhes sobre a implementação prática dos algoritmos de interceptação do FC Portugal podem ser encontrados em [Lau e Reis, 2001].

### 9.4.3 Drible Inteligente

Em 2001, a equipa chinesa *Tsinghuaeolus* venceu o campeonato do mundo de futebol robótico tendo como principal vantagem competitiva um drible inteligente e rápido. Isto demonstra a utilidade de desenvolver algoritmos de drible eficientes e seguros. O drible do FC Portugal embora não seja tão poderoso como o da equipa *Tsinghuaeolus*, permite que o agente, se desloque rapidamente no campo, controlando a bola com segurança e evitando adversários. O ângulo de drible relativamente ao opositor mais próximo é calculado de forma inteligente procurando sempre manter a bola fora do alcance deste. O algoritmo 7 descreve o drible inteligente implementado.

```

ALGORITMO SmartDribble(DesiredDir, ActionWorldState)
    RETORNA DribbleState - Estado actual do drible
PARÂMETROS
    DesiredDir - Direcção desejada de drible no campo
    ActionWorldState - Estado do mundo de acção
{
    DribbleAngle = CalculateDribbleSmartAngle(ActionWorldState)
    DribbleDir = CalculateDribbleSmartDir(DesiredDir, ActionWorldState)
    DribbleState = DribbleExecution(DribbleDir, DribbleAngle, ActionWorldState)
    RETORNA DribbleState
}

```

*Algoritmo 7: Drible Inteligente evitando Adversários*

Para ser possível fintar os adversários enquanto o agente se desloca numa determinada direcção, o algoritmo de drible baseia-se em calcular a melhor posição da bola relativamente ao corpo do agente (colocando a bola o mais longe possível dos adversários de forma a que estes não possam chutar a bola) e a direcção de movimentação do agente tendo em conta o estado do mundo de acção (contornando os adversários mais próximos). Tendo em conta as direcções calculadas, o melhor comando para enviar ao simulador é seleccionado.

O cálculo do ângulo de drible é realizado construindo um vector com os melhores ângulos de drible para driblar cada um dos adversários mais próximos e atribuindo pesos a esses adversários tendo em atenção a sua proximidade e ângulo relativamente à bola.

A direcção de drible é calculada tendo em conta a direcção desejada de drible e a informação contida no estado do mundo de acção. Se não existirem oponentes nessa direcção, o agente movimenta-se com a bola na direcção desejada. No entanto, habitualmente não é este o caso. Assim, a direcção desejada é alterada de forma a contornar os adversários. O cálculo baseia-se em construir um vector com os pesos

atribuídos a cada mudança de direcção possível. Para cada adversário, que se encontre próximo do agente e aproximadamente na direcção de drible desejada, um peso é adicionado à posição do vector correspondente à melhor direcção para o evitar. No final, a melhor direcção é seleccionada efectuando uma filtragem ao vector construído.

No anexo 8 são descritos os algoritmos de cálculo do ângulo e direcção do drible. O algoritmo de execução do drible é demasiado extenso para descrever totalmente, mesmo a alto-nível. Este algoritmo recebe como parâmetros o ângulo e direcção de drible e baseia-se na decisão de um comando *dash*, *kick* ou *turn* dependendo da posição e velocidades relativas da bola, dos oponentes mais próximos, da diferença entre o ângulo de drible desejado e actual e da diferença entre a direcção de drible desejada e actual.

#### 9.4.4 Chuto Optimizado (“*Optimization Kick*”)

A probabilidade de sucesso das acções dos jogadores depende da sua capacidade para manipular a bola correctamente, nomeadamente da sua capacidade para chutar a bola com precisão e com elevada velocidade numa dada direcção.

O modelo de chuto do *SoccerServer*, permite aos jogadores executarem sucessivos comandos *kick*, acelerando e alterando a direcção de movimento da bola, quando esta se encontra dentro de um círculo com um dado raio centrado no jogador. Esta possibilidade pode ser explorada de forma a obter, utilizando uma sequência de *kicks* correctamente calculada, uma maior velocidade de chuto da bola do que utilizando um único comando *kick*. No entanto, encontrar esta sequência de *kicks* para acelerar sucessivamente a bola, não é um problema de resolução trivial uma vez que, a eficiência de um dado chuto depende da distância e direcção relativa do jogador à bola e enormes erros estão associados à percepção, acção e movimento no *SoccerServer*. A figura 64 ilustra o problema do chuto utilizando *kicks* múltiplos, evitando adversários.

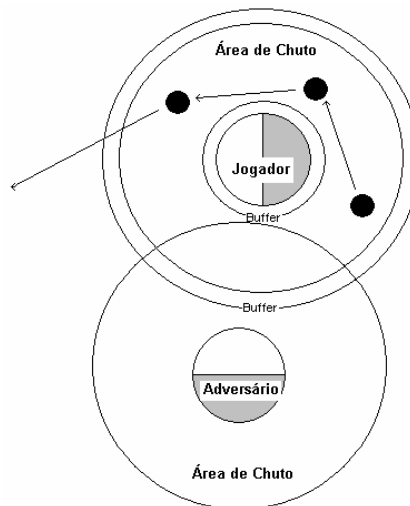


Figura 64: Chuto utilizando Vários Kicks, evitando adversários

Diversas abordagens foram utilizadas pelos investigadores da área na resolução deste problema, incluindo soluções analíticas [Stone et al., 2000a] e a utilização de aprendizagem por reforço [Riedmiller et al., 2000].

O *Optimization Kick* consiste numa metodologia baseada em técnicas de simulação e optimização, para resolver o problema do chuto e definir as acções a executar de forma a conseguir chutar a bola com uma determinada potência numa dada direcção. Tendo como base a dinâmica do *SoccerServer*, foi definida uma classe que permite simular todo o processo de chuto, prevendo a posição e velocidade da bola ao longo de uma sequência de chutos, permitindo assim utilizar algoritmos de pesquisa no sentido de encontrar uma boa solução para o problema.

As condições iniciais do problema de optimização do chuto definem as posições e velocidades iniciais do jogador e da bola e dos adversários mais próximos:

- *RelBallX, RelBallY*: Posição da bola relativamente ao jogador;
- *BallVelX, BallVelY*: Velocidade da bola relativa ao jogador;
- *PlayerVelX, PlayerVelY*: Velocidade absoluta do jogador;
- *OpponentX, OpponentY*: Posição relativa do oponente mais próximo;
- *OpponentVelX, OpponentVelY*: Velocidade relativa do oponente mais próximo;
- *OpponentRelDir*: Direcção relativa do oponente mais próximo.

O problema do chuto é definido como um problema de optimização em que a função a optimizar é:

$$f(|KickDir - KickDesDir|, |KickSpeed - KickDesSpeed|, NKicks, SafetyKickable, SafetyOpp)$$

Em que:

- *KickDir*: direcção de chuto obtida;
- *KickDesDir*: direcção de chuto desejada;
- *KickSpeed*: velocidade de chuto obtida;
- *KickDesSpeed*: velocidade desejada para o chuto;
- *NKicks*: é o comprimento (número de acções) da sequência de chutos;
- *SafetyKickable*: Probabilidade de a sequência de *Kicks* ser efectuada sem que a bola saia da área de controle do jogador nem colida com o mesmo;
- *SafetyOpp*: Probabilidade de durante a sequência de *Kicks* a bola não entrar na área de controle de um oponente.

A solução para o problema pode assumir quatro formas distintas:

- $(Pow1, Dir1)$ : Solução obtida com um único *kick*;
- $(Pow1, Dir1) (Pow2, Dir2)$ : Solução obtida com uma sequência de dois *kicks*;
- $(Pow1, Dir1) (Pow2, Dir2) (Pow3, Dir3)$ : Solução obtida com uma sequência de três *kicks*;
- $(Pow1, Dir1) (Pow2, Dir2) (Pow3, Dir3) (Pow4, Dir4)$ : Solução obtida com uma sequência de quatro *kicks*.

Embora seja possível construir soluções com um número arbitrário de passos, os resultados experimentais obtidos comprovaram que, embora não seja possível obter a solução ótima em termos de precisão na direção e velocidade, em todos os casos, é sempre possível obter uma solução sub-ótima de muito boa qualidade, utilizando quatro *kicks* na sequência<sup>69</sup>. Desta forma e tendo em conta que na qualidade da solução interessa também minimizar o número de chutos, não são consideradas soluções com mais de quatro *kicks* no *Optimization Kick*.

Os parâmetros considerados no algoritmo são:

- *BufferKickable*, *BufferCollision*, *BufferOpponent*: Buffers para evitar que a bola saia da área de chuto do jogador, colida com o corpo do jogador ou entre na área de chuto do oponente;
- *DirRandom*, *PowerRandom*: Factores de aleatoriedade na fase inicial para a direção e para a potência;
- *PowerMin2*: Potência mínima utilizada a partir do segundo chuto;
- *AngleTol2*: Tolerância máxima no ângulo do chuto relativamente ao ângulo desejado, utilizado a partir do segundo chuto;
- *DirStep*, *PowerStep*: Passos utilizados na segunda fase do algoritmo para a direção e potência;
- *MaxIterations1*, *MaxIterations2*: Número de iterações máximo na primeira e segunda fase;

---

<sup>69</sup> Esta conclusão era válida em 2000. Em 2001 a potência de chuto disponível aos jogadores foi aumentada (de 0.016 para 0.027) tornando possível a obtenção de soluções sub-ótimas de muito boa qualidade utilizando unicamente três *kicks* e na maioria dos casos, utilizando unicamente dois *kicks*. Esta alteração às regras limitou também consideravelmente o interesse competitivo do *Optimization Kick*.

```

ALGORITMO OptimizationKick(KickDesDir, KickDesSpeed, KickDesDist, BallRelPos, BallVel,
    PlayerVel, OppInfo)
    RETORNA KickSeq - Sequência de Kicks (Potência, Direcção) que permite chutar a bola
    com a direcção e velocidade desejadas
PARÂMETROS
    KickDesDir - Direcção desejada Velocidade Desejada e distância desejada de chuto
    BallRelPos - Posição relativa da bola (x,y) em relação ao corpo do jogador
    BallVel, PlayerVel - Velocidade (x,y) da bola e jogador
    OppInfo - Informação dos oponentes mais próximos
{
//Pesquisa semi-aleatória seguida de uma pesquisa tabu da solução
    InitKickSeq = RandomSearch(KickDesDir, KickDesSpeed, KickDesDist, BallRelPos,
        BallVel, PlayerVel, OppInfo)
    KickSeq = TabuSearch(InitKickSeq, KickDesDir, KickDesSpeed, KickDesDist, BallRelPos,
        BallVel, PlayerVel, OppInfo)
    RETORNA KickSeq
}

```

*Algoritmo 8: Optimization Kick*

O algoritmo do *Optimization Kick* (algoritmo 8) encontra-se dividido em duas fases:

- **Primeira Fase: Pesquisa Semi-Aleatória da Solução.** Com o objectivo de encontrar uma boa sequência (ou diversas boas sequências) de chutos, *kicks* semi-aleatórios são executados até a bola sair da área de chuto do jogador ou colidir com o mesmo. Nesta altura, o último chuto é substituído por um chuto calculado analiticamente de forma a minimizar o erro da velocidade de chuto na direcção pretendida. Diversas heurísticas são utilizadas para conduzir esta fase para uma boa solução: só o primeiro chuto é completamente aleatório, sendo os seguintes chutos potentes ( $PowerMin2 > 50$ ); a partir do segundo chuto, o ângulo da direcção do chuto com a direcção final é inferior a ( $Angle\_Tol2 < 45$ ). Cada sequência de chuto calculada é avaliada (utilizando a função de avaliação referida) e a melhor é passada para a segunda fase do algoritmo.
- **Segunda Fase: Melhoria Iterativa da Solução.** A solução obtida na primeira fase é melhorada sucessivamente de forma a encontrar um máximo local da função de optimização. O algoritmo seleccionado para esta fase foi a Pesquisa Tabu [Glover e Laguna, 1993]. Em cada iteração, soluções vizinhas da solução anterior são obtidas através da alteração de dois dos parâmetros da sequência de chuto (direcções ou potências) sendo a melhor (não contida na lista tabu) seleccionada como nova solução. Vizinhanças na mesma direcção do último movimento efectuado são exploradas preferencialmente na iteração seguinte do algoritmo. O processo termina no final de um conjunto de iterações pré-determinado.

O algoritmo 9 descreve a primeira fase do processo. Baseia-se na execução de um conjunto de iterações calculando sequências de chuto aleatórias, substituindo o último chuto por um chuto calculado analiticamente. Os chutos calculados são avaliados, sendo o melhor seleccionado, para ser objecto de melhoria iterativa na segunda fase do processo.

```

ALGORTIMO RandomSearch(KickDesDir, KickDesSpeed, KickDesDist, BallRelPos, BallVel,
    PlayerVel, OppInfo)
    RETORNA BestKickSeq - Solução inicial: sequência de Kicks (Potencia, Direcção)
{
// Inicializa o melhor valor para um valor muito baixo
    BestValue = InitializeBestValue()
// Executa um conjunto de iterações calculando sequências de chute aleatórias,
substituindo o último chute por um chute calculado analiticamente. Avalia os chutes e
selecciona o melhor
    PARA Iteration = 1 ATÉ MaxIterations1 FAÇA
    {
        KickSeq = CreateSemiRandomKickSeq(RelBallPos, BallVel, PlayerVel)
        KickSeq = ReplaceLastKickAnalytically(KickSeq, KickDesDir, KickDesSpeed, KickDesDist)
        KickSeqValue = KickSeqEvaluationFunction(KickSeq, KickDesDir, KickDesSpeed, OppInfo)
        SE KickSeqValue > BestValue ENTÃO {
            BestValue = KickSeqValue
            BestKickSeq = KickSeq
        }
    }
    RETORNA BestKickSeq
}

```

### Algoritmo 9: Optimization Kick – Pesquisa Semi-Aleatória

A pesquisa tabu (algoritmo 10) processa-se calculando em cada iteração uma vizinhança da melhor solução utilizando um algoritmo de vizinhança direccional preferencial. A melhor solução da vizinhança é seleccionada como a nova solução e a direcção preferencial, a lista tabu e o valor da solução actual são actualizados. O algoritmo termina ao fim de um número de iterações especificado ou se não for possível obter uma melhor solução.

```

ALGORTIMO TabuSearch(InitKickSeq, KickDesDir, KickDesSpeed, KickDesDist, BallRelPos,
    BallVel, PlayerVel, OppInfo)
    RETORNA KickSeq - Solução final: sequência de Kicks (Potência, Direcção)
PARÂMETROS
    InitKickSeq - Solução inicial a melhorar: sequência de Kicks (Potência, Direcção)
{
// KickSeq é inicializado para a melhor sequência da fase anterior. A lista tabu é
inicializada para uma lista vazia. A última direcção de movimento é inicializada para um
valor nulo
    KickSeq = InitKickSeq
    TabuList = InitializeTabuList()
    LastMoveDir = InitializeLastMoveDir()
    BestValue = KickSeqEvaluationFunction(InitKickSeq, KickDesDir, KickDesSpeed, OppInfo)
// Em cada iteração é calculada uma vizinhança da solução e o melhor elemento dessa
vizinhança. Caso esse elemento seja melhor do que a melhor solução actual adiciona a
solução à lista tabu e a nova solução passa a ser a solução calculada
    PARA Iteration = 1 ATÉ MaxIterations2 FAÇA
    {
// vizinhança que altera dois parâmetros da sequência de chute (potência ou direcção). São
consideradas preferencialmente soluções na direcção do último movimento na solução
determinada.
        Neighborhood = CalculateNeighborhood(KickSeq, LastDir)
//A nova sequência de chute é a melhor contida na vizinhança que não esteja na lista tabu.
        NewKickSeq = CalculateBestKick(TabuList, Neighborhood)
        NewKickSeqValue = KickSeqEvaluationFunction(NewKickSeq, KickDesDir, KickDesSpeed,
            OppInfo)
//Se a nova sequência de chute não for melhor do que melhor solução, termina

```

```

SE KickSeqValue < BestValue ENTÃO RETORNA KickSeq
// Senão actualiza a lista tabu, a direcção do último movimento, o melhor valor e a
solução actual
  LastDir = CalculateDir(KickSeq, NewKickSeq)
  TabuList = UpdateTabuList(TabuList, NewKickSeq)
  BestValue = NewKickSeqValue
  KickSeq = NewKickSeq
}
RETORNA KickSeq
}

```

*Algoritmo 10: Optimization Kick – Pesquisa Tabu*

A função de avaliação do *Optimization Kick* (algoritmo 11) é uma soma pesada dos erros na direcção e velocidade, tempo de execução do chuto (em número de comandos) e segurança evitando colisões e oponentes<sup>70</sup>.

```

ALGORITMO KickSeqEvaluationFunction(KickSeq, KickDesDir, KickDesSpeed, OppInfo)
RETORNA Value - Avaliação de uma sequência de kicks
{
//Determina os parâmetros correspondentes ao resultado da aplicação da sequência de kicks,
nomeadamente a direcção e velocidade finais, o comprimento (em número de acções da
sequência, a segurança de manter a bola na área chutável do jogador sem colidir com o
corpo do mesmo e a segurança de manter a bola fora do alcance dos adversários no decurso
da execução do kick completo.
  KickDir = GetFinalDir(KickSeq)
  KickSpeed = GetFinalSpeed(KickSeq)
  NKicks = GetLength(KickSeq)
  SafetyKickable = GetKickableSafety(KickSeq)
  SafetyOpp = GetOpponentSafety(KickSeq, OppInfo)
//Calcula os erros na distância e velocidade finais.
  ErrorDir = abs(KickDir-KickDesDir)
  ErrorSpeed = abs(KickSpeed-KickDesSpeed)
// Calcula a soma pesada de todos os factores de avaliação.
  Value = WDir * ErrorDir + WSpeed * ErrorSpeed + WKicks * (NKicks-1) +
  WSafKick * SafetyKickable + WSafOpp * SafetyOpp
RETORNA Value
}

```

*Algoritmo 11: Função de Avaliação do Optimization Kick*

A figura 65 demonstra os resultados obtidos pelo algoritmo em 2000 experiências, considerando unicamente a fase de pesquisa semi-aleatória e uma função de avaliação baseada na velocidade (com velocidade objectivo de 2.7metros/ciclo) e direcção de chuto final. É visível que o valor da mediana e média da avaliação obtida nas experiências cresce monotonicamente, atingindo um valor próximo dos 2.40 (para um máximo de 2.70) perto

<sup>70</sup> Os valores dos parâmetros utilizados em competições oficiais em 2000 foram: WDir = 0.5; WSpeed = 10; WKicks = 3; WSafKick = 20; WSafOpp = 5. Em 2001 os valores para WKicks e WSafOpp foram aumentados respectivamente para 10 e 15 pois o aumento na potência de chuto permite obter mais facilmente a velocidade de chuto desejada e os adversários eram de maior qualidade, pelo que se tornava necessário ter mais cuidado com eles.



das 1500 iterações<sup>71</sup>. O valor mínimo obtido nesta fase do algoritmo é reduzido situando-se abaixo de 2.00.

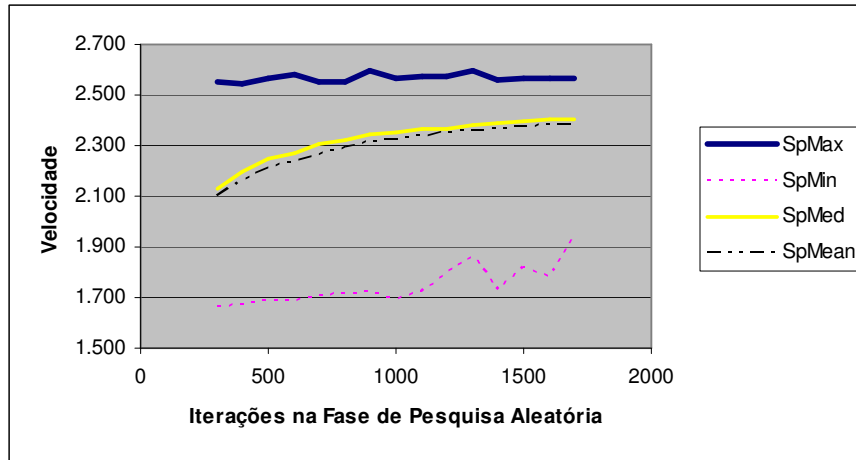


Figura 65: Resultados Obtidos na Primeira Fase do Optimization Kick em 2000 experiências (com  $\text{kick\_power\_rate} = 0.016$ )

A figura 66 contém os resultados obtidos através da segunda fase do algoritmo considerando unicamente 500 iterações na fase semi-aleatória e 150 iterações na fase de melhoria iterativa. É visível que o *Optimization Kick*, mesmo com um número muito reduzido de iterações, permite obter uma velocidade final com mediana de 2.52 metros/ciclo (muito próxima da velocidade máxima – 2.7). Este número de iterações corresponde a um tempo de execução inferior a 1 ms para todo o processo (utilizando um processador Pentium 700Mhz com 256Mb de memória) o que permite a utilização do método *on-line* sem problemas de desempenho.

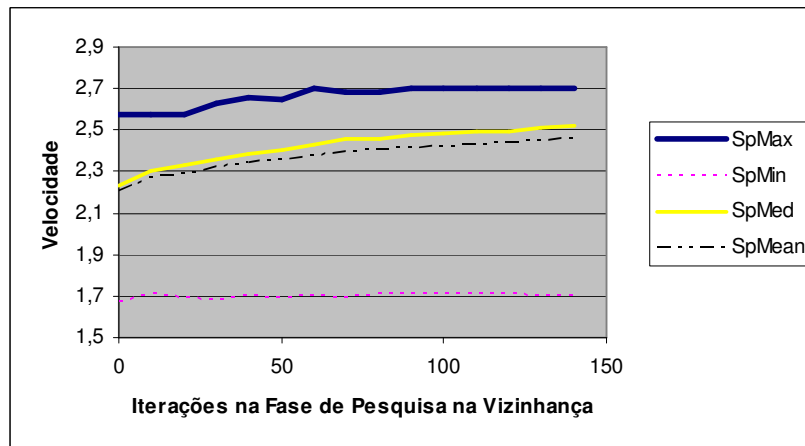


Figura 66: Resultados Obtidos na Segunda Fase do Optimization Kick em 2000 experiências (com  $\text{kick\_power\_rate} = 0.016$ )

<sup>71</sup> Com os parâmetros do servidor em 2000, i.e.,  $\text{kick\_power\_rate} = 0.016$

Diversas experiências utilizando algoritmos genéticos, arrefecimento simulado e outros métodos de optimização e sua comparação experimental com soluções analíticas do problema, foram realizadas no âmbito deste estudo. Uma descrição mais detalhada do algoritmo e das experiências referidas pode ser encontrada em [Lau e Reis Rel FCP, 2002].

## 9.5 Módulo de Decisão de Alto Nível

O funcionamento do módulo de decisão de alto-nível do FC Portugal encontra-se representado na figura 67.

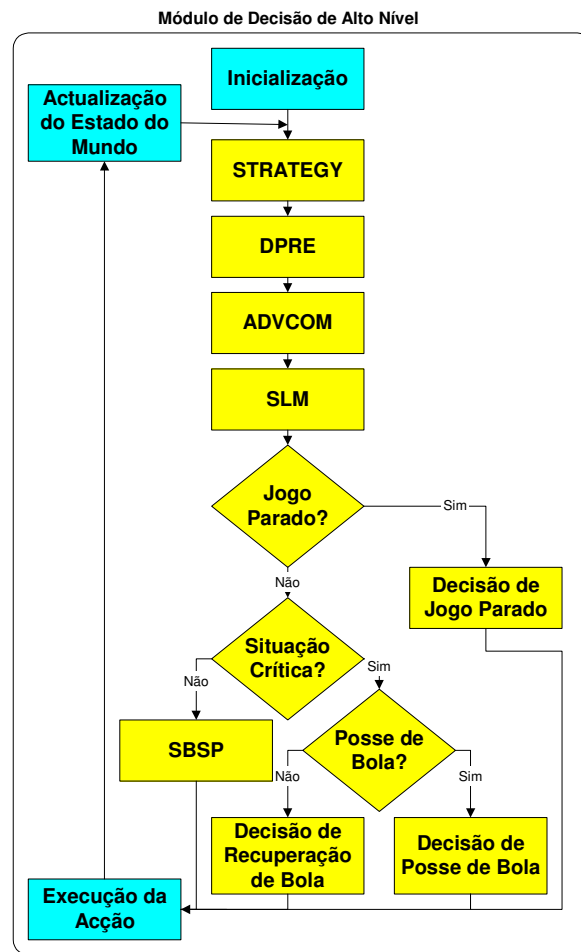


Figura 67: Fluxograma do Módulo de Decisão de Alto-Nível

O processo começa com a inicialização dos agentes, em que são executadas diversas tarefas relacionadas com a leitura dos ficheiros de configuração e inicialização do estado do mundo multi-nível dos agentes.

A fase designada STRATEGY consiste numa análise táctica do jogo. Tendo em conta a informação táctica disponível (tempo, resultado, posse de bola por regiões do campo,

acções individuais dos jogadores e seus resultados, entre outras) e a eventual disponibilidade de instruções do treinador, cada agente vai decidir qual a tática mais apropriada a utilizar.

As instruções do treinador relativas ao comportamento individual, são fundidas com as regras de comportamento individual que o jogador possuía previamente. As instruções relativas ao comportamento colectivo (táticas, parcelas de táticas e formações) são consideradas como ordens e sobrepõe-se às definições previamente existentes. Nesta fase, tendo em conta o estado do mundo de alto-nível situacional e o seu conhecimento, o jogador vai também decidir qual a situação em que o jogo se encontra e decidir qual a formação a adoptar, em consonância.

A fase seguinte consiste na decisão do interesse colectivo na execução de uma troca dinâmica de posição e comportamento com um colega de equipa. Nesta fase, é analisado iterativamente, o interesse que existe em que pares de jogadores realizem uma troca de posição. O algoritmo termina quando se atinge a estabilidade, i.e. quando numa determinada iteração não exista interesse em realizar qualquer troca de posições.

A fase seguinte designa-se por ADVCOM – Comunicação Avançada e encarrega-se das decisões relacionadas com a comunicação, ou seja, quando e o quê comunicar. De acordo com a situação, com o seu conhecimento individual do mundo e com uma estimativa do conhecimento do mundo dos colegas, os agentes decidem, em cada ciclo, se comunicam ou não e qual o conteúdo de uma eventual comunicação.

Tendo em conta a situação actual e a informação contida no estado do mundo, os agentes decidem em seguida o modo de visão a utilizar (abertura e qualidade da visão) e a direcção para onde vão direccionar o seu pescoço flexível no ciclo seguinte.

Se o jogo estiver parado, por exemplo na execução de livres ou lançamentos, os agentes utilizam um módulo de decisão com jogo parado que os procura colocar nas posições correctas para o reinício do jogo. Neste módulo, é necessário considerar a eventual existência de regiões do campo inacessíveis ao agente (por exemplo no caso de um livre a favor da equipa adversária, o agente não se pode aproximar mais do que 9m da bola). De acordo com o tipo de tática que se encontra em execução, o agente pode também optar por uma reposição muito rápida de bola (sem descansar e sem se preocupar com a finalização da colocação dos colegas) ou muito lenta (descansando ao máximo e esperando até ao limite máximo de tempo permitido para movimentar a bola).

A decisão com jogo a decorrer divide-se em duas possibilidades: situação crítica (activa) e situação estratégica (passiva). Em situação crítica, ou seja, quando o agente acha que possui o controle da bola ou pensa poder recuperá-lo rapidamente, os agentes utilizam os módulos de decisão com ou sem bola. Em situação estratégica, os agentes posicionam-se de acordo com o algoritmo *SBSP – Posicionamento Estratégico Baseado em Situações*.

A fase final consiste na definição das acções de baixo-nível a enviar para o servidor, utilizando os comportamentos descritos anteriormente. O módulo reinicia-se com a actualização do estado do mundo multi-nível utilizando a percepção visual recebida (no caso de ter sido recebida nova informação visual no ciclo), a comunicação recebida ao longo do ciclo anterior, proveniente de colegas de equipa e a predição de acções.

## 9.6 Coordenação Estratégica no Futebol Robótico

A coordenação estratégica foi introduzida por Reis et al. [Reis et al., 2001] e consiste na definição de uma estratégia composta por um conjunto de tácticas que implicam formas de actuação distintas de uma equipa de agentes. Em cada instante, uma única táctica encontra-se activa. Cada táctica utiliza um conjunto de formações que implicam não só distribuições espaciais distintas dos agentes mas também a atribuição de papéis distintos, definindo o comportamento de cada agente. Em cada instante, e dada a situação de jogo (análise de alto-nível do mesmo) é seleccionada uma dada formação da táctica activa nesse instante.

### 9.6.1 Definição Formal da Estratégia da Equipa

Na figura 68, encontra-se representada a definição de uma *Estratégia da Equipa*. Na figura é visível que a *Estratégia da Equipa* é composta por um conjunto de *Tácticas* que possuem um conjunto de *Parâmetros Tácticos* associados. Cada *Táctica* define também qual a *Formação* a utilizar em cada *Situação* de jogo. Cada *Formação* contém um conjunto de *Posicionamentos* que atribui a cada *Agente* uma movimentação dinâmica no campo e um *Papel* (tipo de jogador). O *Papel* define o comportamento de um jogador a três níveis: *Estratégico*, *Posse de Bola* e *Recuperação de Bola*.

Formalizando o conceito de *Estratégia de Equipa* dos agentes, apresentado na figura 68, esta será:

$$\text{Estratégia\_Equipa} = (\text{Tácticas}, \text{Agentes}, \text{Papéis}, \text{Formações}, \text{Situações}, \text{Est\_Com}, \text{Est\_Vis}, \text{Est\_Mod}, \text{Planos})$$

Em que *Tácticas*, consistem num conjunto de *ntácticas* aplicáveis à equipa

$$\text{Tácticas} = \{\text{Táctica}_1, \text{Táctica}_2, \dots, \text{Táctica}_{ntácticas}\}$$

*Agentes* representam os *nagentes* que constituem a equipa que irá executar a tarefa cooperativa:

$$\text{Agentes} = \{\text{Agente}_1, \text{Agente}_2, \dots, \text{Agente}_{nagentes}\}$$

*Papéis* são um conjunto de *npapéis* possíveis para os agentes definindo comportamentos individuais distintos.

$$\text{Papéis} = \{\text{Papel}_1, \text{Papel}_2, \dots, \text{Papel}_{n_{\text{papéis}}}\}$$

Formações são um conjunto de  $n_{\text{formações}}$ , que permitem a aplicação de esquemas de posicionamento distintos aos agentes:

$$\text{Formações} = \{\text{Formação}_1, \text{Formação}_2, \dots, \text{Formação}_{n_{\text{formações}}}\}$$

Situações são um conjunto de  $n_{\text{situações}}$  que permite a aplicação de diferentes esquemas de posicionamento aos *agentes*:

$$\text{Situações} = \{\text{Situação}_1, \text{Situação}_2, \dots, \text{Situação}_{n_{\text{situações}}}\}$$

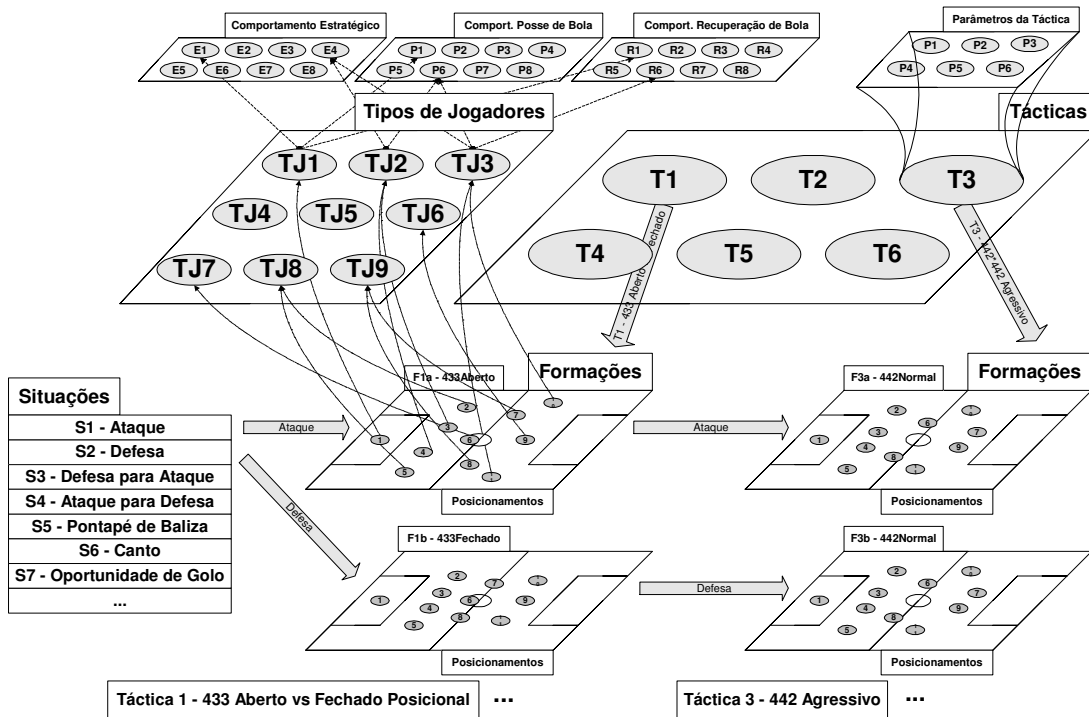


Figura 68: Diagrama da Estratégia do FC Portugal

A estratégia inclui também *Estratégias de Comunicação*, *Estratégias de Visão* e *Estratégias de Modelação de Oponentes* (e colegas de equipa) e *Planos Flexíveis* pré-concebidos:

$$\text{Est\_Com} = \{\text{ECom}_1, \text{ECom}_2, \dots, \text{ECom}_{n_{\text{ecom}}}\}$$

$$\text{Est\_Vis} = \{\text{EVis}_1, \text{EVis}_2, \dots, \text{EVis}_{n_{\text{evis}}}\}$$

$$\text{Est\_Mod} = \{\text{EMod}_1, \text{EMod}_2, \dots, \text{EMod}_{n_{\text{emod}}}\}$$

$$\text{Planos} = \{\text{Plano}_1, \text{Plano}_2, \dots, \text{Plano}_{n_{\text{planos}}}\}$$

As *Táticas* são utilizadas para indicarem à equipa um conjunto de *Parâmetros Táticos* globais, quais as *Formações* e *Planos Flexíveis* que devem ser utilizados em cada *Situação* de jogo (de entre as *Formações* e *Planos* pré-definidos).

$$\text{Tática}_i = (\text{Param\_Tática}_i, \text{Form\_Sit}_i, \text{Plan\_Sit}_i, \text{Reg\_Act\_Tática}_i) \forall i = 1..n_{\text{táticas}}$$

Cada *Plano Flexível* contém *Regras de Ativação*, *Regras de Terminação* e a Evolução dos *Posicionamentos* e *Papéis* dos *Agentes* participantes ao longo do tempo. Contém ainda as *Ações Preferenciais* a executar por cada jogador ao longo da sua execução<sup>72</sup>.

As *Táticas* são seleccionadas de acordo com um conjunto de *Regras de Ativação Táticas* (*Reg\_Act\_Tática*) que utiliza a *Informação Tática* disponível na camada superior do estado do mundo multi-nível. Por exemplo, quando a equipa está a perder próximo do final de um jogo, mas se estiver a ter domínio do jogo (em termos de posse de bola e oportunidades para marcar golos) uma *Tática* mais atacante, arriscada e agressiva poderá ser seleccionada. Se for claro das estatísticas de jogo e informação de modelização do oponente (disponíveis na informação tática do estado do mundo multi-nível), que a equipa adversária está a dominar totalmente o jogo, então uma *Tática* mais defensiva, utilizando formações mais recuadas, que tente minimizar a derrota, pode ser seleccionada.

As *Formações* fornecem aos jogadores informação relativa ao seu *Posicionamento Estratégico* dinâmico e informação relativa ao seu *Papel* individual (características relacionadas com o comportamento individual).

Uma *Formação* é definida por *n* agentes *Posicionamentos* cada qual definindo o *Posicionamento Estratégico* de cada *Agente* e respectivos *Papel* e *Importância* para a equipa.

$$\text{Formação}_i = \{\text{Posicionamento}_{i,1}, \text{Posicionamento}_{i,2}, \dots, \text{Posicionamento}_{i,n_{\text{Agentes}}}\} \\ \forall i = 1..n_{\text{formações}}$$

Para cada *Tática*, em cada *Situação* é utilizada uma das *Formações* definidas:

$$\text{Sit\_Form}_i = \{\text{FormAtrib}_{i,p}, \text{FormAtrib}_{i,2}, \text{FormAtrib}_{i,3}, \dots, \text{FormAtrib}_{i,n_{\text{situações}}}\} \\ \forall i = 1..n_{\text{táticas}}$$

As *Formações* utilizadas em cada *Situação* para cada *Tática* devem estar definidas na *Estratégia da Equipa*:

$$\text{FormAtrib}_{i,s} \in \{1..n_{\text{formações}}\} \quad \forall i = 1..n_{\text{táticas}} \quad \forall s = 1..n_{\text{situações}}$$

O *Posicionamento* de um dado *Agente* numa *Formação* é definido pela sua *Posição Estratégica de Referência*, pelo seu *Papel* e pela sua *Importância*:

$$\text{Posicionamento}_{i,p} = (\text{PosiçãoReferência}_{i,p}, \text{PapelAtribuido}_{i,p}, \text{ImportânciaPosicionamento}_{i,p}) \\ \forall i = 1..n_{\text{formações}} \quad \forall p = 1..n_{\text{players}_i}$$

Para cada *Posicionamento* numa dada *Formação*, é definida uma *Posição Estratégica de Referência*:

---

<sup>72</sup> Neste trabalho não é abordada a definição dos planos flexíveis e sua utilização prática no domínio do futebol robótico simulado pois esse trabalho de investigação foi realizado em data posterior ao início da escrita desta dissertação.

$$PosiçãoReferência_{i,p} = (PosiçãoReferênciaX_{i,p}, PosiçãoReferênciaY_{i,p})$$

$$PosiçãoReferênciaX_{i,p} \in \{-field\_length..field\_length\}$$

$$PosiçãoReferênciaY_{i,p} \in \{-field\_width..field\_width\}$$

A *Posição Estratégica de Referência* é ajustada tendo em conta a informação correspondente à movimentação estratégica da *Formação* e à movimentação estratégica do *Papel* assumido pelo *Agente*.

Cada *Posicionamento* tem associada uma dada *Importância*. Por exemplo, no domínio do futebol robótico o *Posicionamento* do guarda-redes, defesa central e avançado centro são muito importantes, nunca devendo estar desocupados. A *Importância* de cada *Posicionamento* pode ser definida numa escala qualitativa:

$$ImportânciaPosicionamento_{i,p} \in \{VeryLow, Low, Medium, High, VeryHigh\}$$

Cada *Papel* define as características de um determinado *Agente*. Todos os *Papéis* utilizados em cada uma das *Formações* devem estar definidos na *Estratégia da Equipa*:

$$PapelAtribuido_{i,p} \in \{1..npapeis\} \quad \forall i = 1..nformações \quad \forall p = 1..nagentes_i$$

Os *Papéis* definidos na *Estratégia da Equipa* atribuem a cada *Agente* características a dois níveis distintos: *Estratégicas* e *Activas*. No domínio do futebol robótico simulado as *Características Activas* incluem dois tipos de características: *Posse de Bola* e *Recuperação de Bola*. Cada *Papel* inclui ainda um conjunto de *Regras de Situação Crítica* indicando as condições para o agente abandonar o *Comportamento Estratégico* e utilizar um *Comportamento Activo*.

$$Papel_i = (CaracterísticasActivas_i, CaracterísticasEstrategicas_i, RegrasSituaçãoCrítica_i) \\ \forall i = 1..npapeis$$

As características de *Posse de Bola* e *Recuperação de Bola*, contrariamente aos restantes conceitos apresentados na definição da *Estratégia da Equipa*, são dependentes do domínio e devem ser definidas de forma adequada a este.

$$CaracterísticasActivas_i = (CaracterísticasPosseBola_i, CaracterísticasRecuperaçãoBola_i, \\ RegrasPosseBola_i) \\ \forall i = 1..npapeis$$

Os *Papéis* podem ser utilizados para definir *Agentes* com diferentes comportamentos tais como um *defesa\_agressivo*, *defesa\_posicional*, *extremo\_driblador*, *medio\_distribuidor*, etc.

Uma descrição mais detalhada da definição de características activas e estratégicas, independentes do domínio, para os papéis definidos, pode ser encontrada em [Reis et al, 2001]

## 9.6.2 Definição de Situações

A aplicação da coordenação estratégica implica que os *Agentes* possuam a definição de um conjunto de *Situações* facilmente identificáveis e cuja identificação temporal não sofra variações consideráveis de *Agente* para *Agente*. Cada *Situação* é definida em termos de condições lógicas para os parâmetros respectivos no estado do mundo de alto-nível. No âmbito da aplicação da coordenação estratégica ao futebol robótico simulado, definiram-se vinte *Situações*. No entanto, a utilização da coordenação estratégica não implica a necessidade de definição de *Formações* para todas estas *Situações*. Cada *Situação* possui uma série de *Situações Substitutas* que a substituem no caso de não se encontrar uma *Formação* definida para ela. As *Situações* definidas e respectivas *Situações Substitutas* consideradas encontram-se representadas na tabela 20.

<i>Situação</i>	<i>Sit. Substituta 1</i>	<i>Sit. Substituta 2</i>	<i>Sit. Substituta 3</i>
<i>Defesa</i>			
<i>Ataque</i>	<i>Defesa</i>		
<i>Passagem Defesa-Ataque</i>	<i>Defesa</i>		
<i>Passagem Ataque-Defesa</i>	<i>Ataque</i>	<i>Defesa</i>	
<i>Ataque Perigoso (Equipa)</i>	<i>Ataque/Defesa</i>		
<i>Livre (Equipa)</i>	<i>Ataque/Defesa</i>		
<i>Reposição de Bola (Equipa)</i>	<i>Livre(Equipa)</i>	<i>Ataque/Defesa</i>	
<i>Pontapé de Baliza (Equipa)</i>	<i>Reposição de Bola (Equipa)</i>	<i>Livre(Equipa)</i>	<i>Ataque/Defesa</i>
<i>Pontapé de Início (Equipa)</i>	<i>Livre(Equipa)</i>	<i>Ataque/Defesa</i>	
<i>Lançamento (Equipa)</i>	<i>Livre(Equipa)</i>	<i>Ataque/Defesa</i>	
<i>Canto (Equipa)</i>	<i>Lançamento (Equipa)</i>	<i>Livre(Equipa)</i>	<i>Ataque/Defesa</i>
<i>Penalty (Equipa)</i>	<i>Livre (Equipa)</i>	<i>Ataque/Defesa</i>	

Tabela 20: *Situações utilizadas e Situações substitutas*

Embora o conceito seja generalizável a outros domínios, a forma de definição das *Situações* é específica de cada domínio. Por exemplo, no domínio do futebol robótico, uma *Situação* de ataque poderia ser definida (utilizando informação contida no estado do mundo situacional) como:

*AttackSituation* =

$(PlayMode = Play\_On \ E \ BallPossession(Our) \ E \ TimeBallPossession > TimeThres \ E$   
 $TheirMiddleField(BallPos) \ E \ Não(TheirDangerAttackPos(BallPos)))$

Uma situação de ataque existe quando o modo de jogo indica que o jogo não está interrompido, que a posse de bola ser da nossa equipa (o jogador mais rápido a interceptar a bola é de equipa e o último chute na bola ter sido executado por um jogador da equipa) durante um limiar de tempo pré-especificado, que a bola está localizada no meio campo do adversário mas não numa região perigosa.



A definição apresentada é muito simplificada relativamente às definições utilizadas em competições oficiais pela equipa FC Portugal para as várias *Situações* de jogo apresentadas. Em virtude de estas definições serem totalmente dependentes do domínio, não são descritas em detalhe neste trabalho. Uma análise mais detalhada das *Situações* utilizadas na equipa FC Portugal e sua definição, encontra-se em [Reis e Lau, 2001c].

### 9.6.3 SBSP – Posicionamento Estratégico Baseado em Situações

O *Posicionamento Estratégico Baseado em Situações* baseia-se na definição de *Formações* dinâmicas que estabelecem em cada instante uma *Posição Estratégica* para cada *Agente* que compõe a equipa. Em cada instante a *Formação* correspondente à *Situação* actual de jogo é activada, e tendo em conta parâmetros específicos do domínio a sua dinâmica é definida pelo algoritmo 12, que dada a tática e situação actuais, determina a posição estratégica para um agente que ocupa um dado posicionamento na formação.

O algoritmo de *Posicionamento Estratégico Baseado em Situações* calcula, tendo em conta a *Táctica* e *Situação* actuais, a *Posição Estratégica* ( $x, y$ ) no campo para um dado *Agente* que ocupa um dado *Posicionamento* na *Formação*. Começa por determinar a *Formação* e respectivos parâmetros utilizados na *Táctica* e *Situação* activa. Em seguida determina a *Posição de Referência* ( $x, y$ ) para o *Posicionamento* que o *Agente* ocupa actualmente na *Formação* (tendo em conta as trocas dinâmicas de posicionamentos efectuadas e os parâmetros da *Formação* actual).

No caso de existir informação dinâmica específica para o *Papel Estratégico* atribuído ao *Agente*, esta informação é utilizada. Caso contrário, a informação dinâmica da *Formação* é utilizada por defeito. Esta informação dinâmica inclui a *Atração pela Bola*, *Atrações e Repulsões por Regiões* específicas do campo ou objectos, quando a bola se encontra em regiões pré-especificadas, e *Limites de Movimentação* para cada jogador.

A *Posição Estratégica Base* do *Agente* é alterada considerando a *Atração pela Bola* definida para o seu *Papel Estratégico* ou *Formação*. A *Posição da Bola* considerada para este efeito corresponde à posição prevista da próxima intercepção da bola por um dado *Agente*.

A *Posição Estratégica Base* calculada é ajustada posteriormente considerando as *Atrações e Repulsões por Regiões* ou objectos pré-especificados do campo e *Limites de Movimentação* para o *Papel Estratégico* ou *Formação*. Isto permite uma fácil adaptação do algoritmo de posicionamento estratégico a outros domínios.

Tendo em conta informação específica do domínio, a *Posição Estratégica* é ajustada para uma posição válida. No caso do futebol robótico simulado, as posições inválidas incluem, entre outras: posições de fora de jogo, posições no interior da área adversária no instante em que ocorrem os pontapés de baliza, posições no meio campo adversário antes do início

do jogo, posições a menos de nove metros de distância do local da marcação de livres. Informação específica do domínio pode também ser utilizada, numa fase final, para ajustar a *Posição Estratégica* para uma posição mais útil.

```

ALGORITMO SituationBasedStrategicPositioning(Tactic, Situation, Player)
RETORNA Position - Vector contendo a Posição Estratégica do Jogador Player na presente
Tática e Formação

PARÂMETROS
    Tactic - Tática activa da estratégia da equipa
    Situation - Situação activa
    Player - Número do jogador cujo posicionamento estratégico vai ser calculado
{
// Dada a Tática e Situação actuais determina qual a Formação a utilizar e os parâmetros
relativos ao seu comprimento (eixo dos xx) e largura (eixo dos yy)
    Formation = SituationFormation(Tactic, Situation)
    FormHeight = SituationFormationHeight(Tactic, Situation)
    FormWidth = SituationFormationWidth(Tactic, Situation)
    Positioning = PlayerGetPositioning(Player)
// Determina a Posição de referência para o posicionamento que o agente ocupa actualmente
(é necessário ter em conta que os agentes podem trocar dinamicamente de posicionamentos)
    Position = HomeSBSPPosition(Formation, FormWidth, FormHeight, Positioning)
// Determina qual o papel atribuído ao jogador que ocupa o posicionamento em causa
e qual o papel estratégico correspondente a esse papel
    Role = FormationGetRole(Formation, Positioning)
    RoleStrategic = RoleGetStrategicRole(Role)
// Calcula posição da bola a utilizar no algoritmo de posicionamento estratégico. A
posição utilizada é a próxima posição prevista de intercepção da bola por um qualquer
jogador, ajustada para o interior do campo
    BallAdjPos = AdjustedBallPosition(BallPosition)
// Utilizando a posição básica e a informação relativa à dinâmica da formação e ao papel
estratégico do jogador, calcula a posição estratégica do jogador
    Position = BasicSBSPPosition(Position, Formation, RoleStrategic, Positioning,
BallAdjPos)
// Utilizando a posição estratégica, e informação definida na formação e papel estratégico
do jogador relativa a atracções e repulsões de regiões do campo, calcula a nova posição
estratégica do jogador
    Position = RegionalAdjustSBSPPosition(Position, Formation, RoleStrategic, Positioning,
BallAdjPos)
//Ajusta a posição SBSP para uma posição legal (i.e. admissível tendo em conta informação
específica do domínio). Por exemplo, a posição não deve estar em "fora de jogo", antes do
jogo começar a posição tem de estar no meio campo defensivo, na marcação de livres tem de
estar a 9m de distância, na marcação de pontapés de baliza tem de estar fora da área
    Position = LegalAdjustSBSPPosition(Position, BallAdjPos)
//Ajusta a posição SBSP para uma posição mais útil, tendo em conta informação específica
do domínio, por exemplo garantindo distâncias mínimas entre jogadores, ficando atrás da
linha da bola em determinadas situações, etc.
    Position = DomainAdjustSBSPPosition(Position, BallAdjPos)
RETORNA Position
}

```

### Algoritmo 12: Posicionamento Estratégico Baseado em Situações

O algoritmo 13 calcula a posição base SBSP do jogador, de acordo com a formação que se encontra em utilização e respectiva largura e altura.

```

ALGORITMO HomeSBSPPosition(Formation, Width, Height, Positioning)
RETORNA Position - Vector com a posição estratégica base de um posicionamento da formação
PARÂMETROS
  Formation - Formação activa da estratégia da equipa
  Width, Height - Largura (eixo dos yy) e comprimento (eixo dos xx) da formação
  Positioning - Posicionamento na formação cuja posição estratégica vai ser calculada
{
// Na informação relativa à formação actual retira a posição base do jogador que ocupa o
posicionamento Positioning na formação.
  Position = FormationGetHomePosition(Formation, Positioning)
//Ajusta a posição base do jogador tendo em conta a largura e comprimento da formação
(factorios multiplicativos)
  Position.x = Position.x * Height
  Position.y = Position.y * Width
RETORNA Position
}

```

*Algoritmo 13: Cálculo da Posição Estratégica Base*

O algoritmo de posicionamento estratégico básico (algoritmo 14), ajusta a posição base SBSPP de acordo com a dinâmica da formação (ou do jogador individualmente) tendo em conta a posição da bola.

```

ALGORITMO BasicSBSPPosition(Position, Formation, RoleStrategic, Positioning, BallAdjPos)
RETORNA Position - Vector com a posição estratégica dinâmica ajustada para a posição
da bola
PARÂMETROS
  Position - Posição estratégica base do jogador
  Formation - Formação activa da estratégia da equipa
  RoleStrategic - Papel estratégico do jogador que ocupa Positioning
  Positioning - Posicionamento na formação cuja posição estratégica vai ser calculada
  BallAdjPos - Posição ajustada da bola relativa à qual Position vai ser calculada
{
// Se o Papel do jogador tiver definida a atracção pela posição da bola então utiliza essa
informação, senão utiliza a informação genérica da formação
  SE (RoleGetBallAttraction(RoleStrategic) != NULL) ENTÃO
    Attraction = RoleGetBallAttraction(RoleStrategic)
  SENÃO Attraction = FormationGetBallAttraction(Formation, Positioning)
// Ajusta a posição de acordo com a Posição ajustada da Bola de forma a deslocar a
formação coordenadamente para a zona onde a bola se encontra (rá)
  Position.x = Position.x + BallAdjPos.x * Attraction.x
  Position.y = Position.y + BallAdjPos.y * Attraction.y
RETORNA Position
}

```

*Algoritmo 14: Posicionamento Estratégico Dinâmico*

O algoritmo 15 aplica atracções e repulsões por pontos e limita a posição estratégica à região admissível considerada no papel ou formação. Embora o algoritmo seja genérico, as regiões consideradas como pontos de atracção podem por exemplo ser as marcas de *penalty* (no caso de a bola estar próxima da área) ou pontos de chute pré-definidos. Possíveis regiões consideradas como pontos de repulsão podem ser as linhas de limite do campo (em qualquer caso).

```

ALGORITMO RegionalAdjustSBSPPosition(Position, Formation, RoleStrategic, Positioning,
    BallAdjPos)
RETORNA Position - Vector com a posição estratégica dinâmica ajustada considerando
    atracções e repulsões regionais
PARÂMETROS
    Position - Posição estratégica base do jogador
    Formation - Formação activa da estratégia da equipa
    RoleStrategic - Papel estratégico do jogador que ocupa Positioning
    Positioning - Posicionamento na formação cuja posição estratégica vai ser calculada
    BallAdjPos - Posição ajustada da bola relativa à qual Position vai ser calculada
{
// Para cada região considerada como possível conducente a atracções/repulsões por pontos
    PARA Region = 1 ATÉ RegionMax FAÇA
// Se a posição ajustada da bola estiver na região considerada então
    Se InRegion(BallAdjPos, Region) ENTÃO
    {
//Se existir informação de atracção/repulsão no papel estratégico, é utilizada essa
informação, senão é utilizada a informação geral da formação. Valores negativos de
atracção correspondem a repulsões. É também determinada a região de atracção (que pode ser
um ponto, linha, círculo ou rectângulo.
        SE (RoleGetRegionAttraction(RoleStrategic) != NULL) ENTÃO
        {
            RegionAttraction = RoleGetRegionAttraction(Region, RoleStrategic)
            RegionAttractionLoc = RoleGetRegionAttractionLoc(Region, RoleStrategic)
        }
        SENÃO
        {
            RegionAttraction = FormationGetRegionAttraction(Region, Formation,
                Positioning)
            RegionAttractionLoc = FormationGetRegionAttractionLoc(Region, Formation,
                Positioning)
        }
//Ajusta a posição estratégica do jogador considerando o valor da atracção e a região para
onde a posição vai ser atraída/repelida
        Position = ApplyRegionAttraction(Position, RegionAttraction,
            RegionAttractionLoc)
    }
//Garante que a posição estratégica não sai dos limites considerados para o papel
estratégico do jogador ou formação
    SE (RoleGetAdmissibleRegion(RoleStrategic) != NULL) ENTÃO
        RegionAdmissible = RoleGetAdmissibleRegion(RoleStrategic)
    SENÃO RegionAdmissible = FormationGetAdmissibleRegion(Formation, Positioning)
    Position = LimitRegionToRegionAdmissible(Position, RegionAdmissible)
    RETORNA Position
}

```

*Algoritmo 15: Posicionamento Estratégico Baseado em Situações*

Na figura 69, é apresentado um exemplo demonstrando posicionamentos calculados pelo algoritmo SBSP para diversas situações de jogo de ataque e defesa.

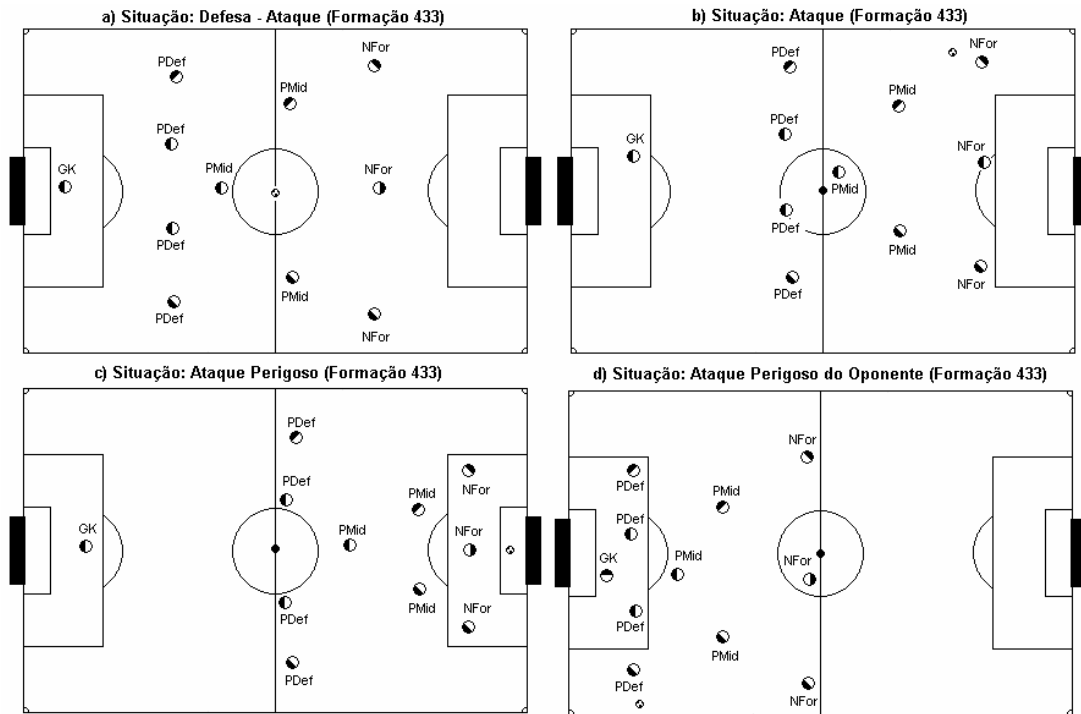


Figura 69: Exemplos de Aplicação do Algoritmo SBSP em Diversas Situações

Na figura 70 é visível a diferença do posicionamento utilizando o SBSP e o algoritmo utilizado pelos campeões de futebol robótico simulado em 1998 e 1999: SPAR [Stone et al., 2000a].

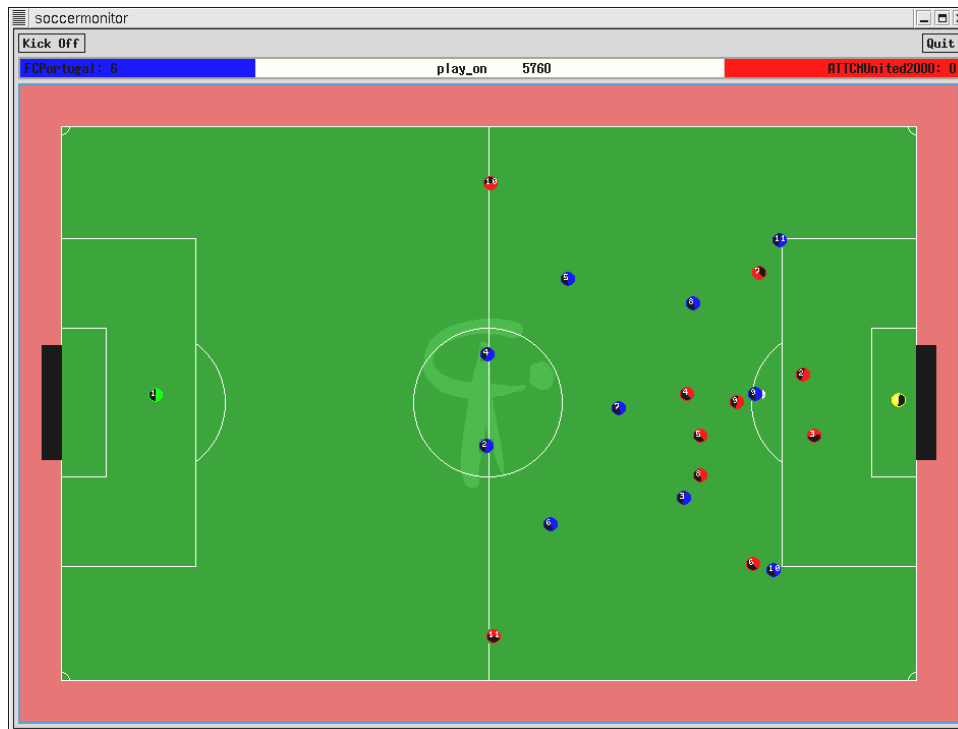


Figura 70: Comparação entre o SBSP e o algoritmo de posicionamento SPAR

O SPAR é um posicionamento activo baseado em atracções por adversários e repulsões por colegas de equipa. Na imagem são visíveis as vantagens de um posicionamento estratégico relativamente a um posicionamento activo. Enquanto os jogadores da equipa FC Portugal se encontram bem distribuídos no campo, numa formação 433, os jogadores da equipa adversária, utilizando o algoritmo SPAR não conseguem compor uma formação. Aliás, é também visível que o jogador da equipa FC Portugal que tem a posse de bola (avanzado centro) possui não só espaço para avançar no terreno e rematar à baliza como também linhas de passe livres para os outros dois avançados da equipa.

Na secção 9.13.2, a influência do SBSP no comportamento global da equipa, nomeadamente nos seus resultados, será objecto de uma análise mais detalhada.

### 9.6.4 DPRE – Troca Dinâmica de Papéis e Posicionamentos

A coordenação estratégica baseia-se na definição de um conjunto de *Táticas* compostas por *Formações* que atribuem a cada jogador, em cada instante, um *Posicionamento* (posição estratégica dinâmica) e um *Comportamento* distinto. Isto permite efectuar também dinamicamente a troca da alocação de cada posicionamento aos elementos que compõem a equipa, trocando também o respectivo comportamento.

A *Troca Dinâmica de Papéis e Posicionamentos (DPRE – Dynamic Positioning and Role Exchange)* utiliza o cálculo de uma função de utilidade para avaliar, em cada instante, a utilidade de efectuar a troca de posicionamentos entre agentes da equipa.

Como parte do seu estado do mundo, cada agente possui uma estimativa da posição de todos os outros agentes no campo:

$$\begin{aligned} \text{Posi\c{c}o\c{e}sAgentes} &= \{ \text{PosAgente}_1, \text{PosAgente}_2, \dots, \text{PosAgent}_{n_{\text{agentes}}} \} \\ \text{PosAgente}_p &= ( \text{PosAgente}X_p, \text{PosAgente}Y_p ) \\ &\forall p = 1..n_{\text{agentes}} \end{aligned}$$

Em cada instante do jogo, cada agente possui um dado posicionamento e um papel (características de comportamento) na formação utilizada para a situação actual. Cada agente conhece não só o seu próprio posicionamento mas também a informação de quais os posicionamentos estimados para os restantes membros da equipa:

$$\begin{aligned} \text{PosicionamentosAlocados} &= \{ \text{PosicionAloc}_1, \text{PosicionAloc}_2, \dots, \text{PosicionAloc}_{n_{\text{agentes}}} \} \\ \text{PosicionAloc}_p &\in \{1..n_{\text{agentes}}\} \end{aligned}$$

Se a equipa for composta por agente homogéneos, cada agente pode desempenhar o papel de qualquer outro agente e ocupar qualquer posicionamento na formação. Tendo em conta que a equipa é composta por agentes cooperativos, a troca dinâmica de posicionamentos e papéis deverá ocorrer sempre que tal possa aumentar a utilidade global da equipa. Esta utilidade é calculada utilizando a seguinte fórmula:

$$\begin{aligned}
 \text{UtilidadeDPRE}(\text{form}, p_a, p_b) = & \\
 & \text{UtilidadeChange}(\text{form}, p_a, p_b) - \text{UtilidadeKeep}(\text{form}, p_a, p_b) \\
 & \forall p_a = 1..n_{\text{agentes}}_{\text{form}}, \forall p_b = 1..n_{\text{agentes}}_{\text{form}}, p_a < p_b
 \end{aligned}$$

ou seja, a utilidade DPRE é definida como a diferença entre as utilidades de trocar e manter os posicionamentos. Estas utilidades são calculadas tendo em consideração as distâncias das posições actuais dos agentes às suas posições estratégicas, adequações dos agentes a cada papel e importância dos posicionamentos (algoritmo 16).

Se a utilidade para a troca de posicionamentos e papéis para um dado par  $(p_a, p_b)$  for positiva, então a troca dinâmica de posicionamentos e papéis tem lugar e cada agente assume o posicionamento e papel do outro agente.

$$\begin{cases}
 \text{PositionAloc}_{\text{form}, p_a} = \text{PositionAloc}_{\text{form}, p_b} \\
 \text{PositionAloc}_{\text{form}, p_b} = \text{PositionAloc}_{\text{form}, p_a}
 \end{cases}$$

O algoritmo 16 demonstra a aplicação do DPRE ao domínio do futebol robótico simulado ou a outro domínio que inclua movimentação espacial de uma equipa de agentes homogéneos ou heterogéneos. Baseia-se em calcular a actualização de um vector contendo os posicionamentos de todos os jogadores da equipa, tendo em consideração possíveis trocas de posicionamentos entre os jogadores.

```

ALGORITMO DynamicPositioningExchange(WorldState, Situation, Positionings)
RETORNA Positionings(TeamSize) - Vector com os posicionamentos na formação de todos os
    jogadores da equipa

PARÂMETROS
    WorldState - Estado do Mundo disponível ao agente
    Positionings[TeamSize] - Posicionamentos atribuídos aos agentes da equipa
    Situation - Situação do jogo
{
// Para cada par de jogadores da equipa exceptuando o guarda-redes averiguar a utilidade
da sua troca de posicionamentos na formação actual
    PARA Jog1 = 2 ATÉ TeamSize-1 FAÇA
        PARA Jog2 = Jog1+1 ATÉ TeamSize FAÇA {
// Se as posições de ambos os jogadores forem válidas
            SE PositionValid(Jog1) E PositionValid(Jog2) ENTÃO {
// Calcula as distâncias de cada um dos jogadores à sua posição estratégica e à do colega
                Dist11 = Distance(Position(Jog1), SBSPPosition(Jog1))
                Dist22 = Distance(Position(Jog2), SBSPPosition(Jog2))
                Dist12 = Distance(Position(Jog1), SBSPPosition(Jog2))
                Dist21 = Distance(Position(Jog2), SBSPPosition(Jog1))
// Calcula as adequações de cada um dos jogadores ao seu posicionamento actual e ao do
colega
                Adeq11 = PosAdequacy(Jog1, Positioning[Jog1])
                Adeq22 = PosAdequacy(Jog2, Positioning[Jog2])
                Adeq12 = PosAdequacy(Jog1, Positioning[Jog2])
                Adeq21 = PosAdequacy(Jog2, Positioning[Jog1])
// Calcula a utilidade de efectuar uma troca dinâmica de posições
                Util = ExchangePositions(DPREMode, Situation,
                    Dist11, Dist22, Dist12, Dist21, Adeq11, Adeq22, Adeq12, Adeq21,
                    PosImportance(Positioning[Jog1], PosImportance(Positioning([Jog2]))
// Se a utilidade de troca for superior a um dado limiar dependente da situação de jogo,
então troca os posicionamentos
                SE Util > ThresUtil(Situation) ENTAO {

```

```

        Aux = Positionings[Jog1]
        Positionings[Jog1] = Positionings[Jog2]
        Positionings[Jog2] = Aux
    }
}
}
RETORNA Positionings
}

```

*Algoritmo 16: Troca Dinâmica de Papéis e Posicionamentos*

O cálculo da utilidade das trocas dinâmicas de posicionamentos é uma soma pesada de três factores relacionados respectivamente:

- Ganho de distância relativamente às posições estratégicas;
- Ganho de adequação dos agentes aos posicionamentos;
- Ganho de cobertura de posições importantes.

O algoritmo 17 descreve o método utilizado para realizar este cálculo de utilidade.

```

ALGORITMO ExchangePositions(DPREMode, Situation, Dist11, Dist22, Dist12, Dist21,
    Adeq11, Adeq22, Adeq12, Adeq21, Imp1, Imp2)
RETORNA
    Util - Utilidade da troca de posicionamentos de dois jogadores
PARÂMETROS
    DPREMode - Modo de troca de posicionamentos considerado
    Situation - Situação de jogo
    Dist11, Dist22, Dist12, Dist21 - Distâncias entre as posições actuais e estratégicas
    Adeq11, Adeq22, Adeq12, Adeq21 - Adequações dos agentes aos posicionamentos
    Imp1, Imp2 - Importâncias dos posicionamentos
{
// Cálculo do ganho em distância às posições estratégicas com a troca de posicionamentos
    DistKeep = Dist11 + Dist22
    DistChange = Dist12 + Dist21
    DistGain = DistKeep - DistChange
// Cálculo do ganho em adequação aos posicionamentos com a troca
    AdeqKeep = Adeq11 + Adeq22
    AdeqChange = Adeq12 + Adeq21
    AdeqGain = AdeqChange - AdeqKeep
// Cálculo do ganho em cobertura de posições importantes com a troca. O nível de cobertura
é calculado para o caso de manutenção de posições e troca de posições. Baseia-se na soma
da multiplicação da importância de cada posicionamento pelo respectivo nível de cobertura
    CoverKeep = Imp1*CoverLevel(Dist11,Adeq11) + Imp2*CoverLevel(Dist22,Adeq22)
    CoverChange = Imp1*CoverLevel(Dist21,Adeq21) + Imp2*CoverLevel(Dist12,Adeq12)
    CoverGain = CoverChange- CoverKeep
// Cálculo da utilidade tendo em conta factores WDist, WAdeq e WImp que estão definidos
para cada modo de troca dinâmica de posições e cada Situação de jogo
    Util = WDist(DPREMode, Situation) * DistGain +
        WAdeq(DPREMode, Situation) * AdeqGain +
        WImp(DPREMode, Situation) * CoverGain
RETORNA Util
}

```

*Algoritmo 17: Calculo da Utilidade de uma Troca Dinâmica de Papéis e Posicionamentos*



Embora as funções de utilidade sejam definidas para que convirjam espacialmente no sentido em que para cada par de agentes seja impossível um agente pensar que a troca de posições é útil e outro pensar o oposto, o facto de se tratar de um sistema multi-agente com agentes com percepções distintas do ambiente, torna o problema da troca de posicionamentos mais complexo. Dado os agentes possuírem apenas uma perspectiva local do ambiente, as suas utilidades para uma troca DPRE serão distintas e como tal um determinado agente pode considerar esta troca útil e o outro não. Para a resolução deste problema, é utilizada a comunicação para reforçar a troca dinâmica de posicionamentos. Acresce a este facto que quando um agente considera que a troca de posicionamentos tem uma utilidade positiva, irá tentar realizar essa troca deslocando-se para a posição estratégica do colega de equipa, o que tende a aumentar a utilidade estimada da troca para ambos os agentes.

De forma a testar o algoritmo de troca dinâmica de posicionamentos e papéis foram definidos os seguintes modos de DPRE:

- *DPRE0*: Trocas dinâmicas de posição não são efectuadas;
- *DPRE1*: Trocas dinâmicas de posição pouco frequentes baseadas unicamente em distâncias às posições estratégicas;
- *DPRE2*: Trocas dinâmicas de posição frequentes baseadas unicamente em distâncias às posições estratégicas;
- *DPRE3*: Trocas dinâmicas de posição frequentes baseadas em distâncias às posições estratégicas e adequações aos posicionamentos;
- *DPRE4*: Trocas dinâmicas de posição frequentes baseadas em distâncias às posições estratégicas, adequações aos posicionamentos e importâncias dos posicionamentos.

Uma primeira avaliação do algoritmo de troca dinâmica de posicionamentos desenvolvido foi realizada aferindo o número de descoordenações entre agentes no decurso de uma série de jogos utilizando os modos de *DPRE1* e *DPRE2* (tabela 21)<sup>73</sup>.

---

<sup>73</sup> No RoboCup 2000 e no RoboCup 2001 a troca dinâmica de posicionamentos utilizada pelo FC Portugal foi o DPRE2. Nestas competições, a equipa FC Portugal não utilizou jogadores heterogéneos (introduzidos em 2001).

Adversário	Modo DPRE 1					Modo DPRE 2				
	Num DPRE	Desc1	Desc2	Desc1 /Num	Desc1 /Num	Num DPRE	Desc1	Desc2	Desc1 /Num	Desc2 /Num
CMUnited99	19.6	11.6	2.6	0.59	0.13	32.8	14.1	2.9	0.43	0.09
ATTCMUnited2000	26.3	15.5	3.3	0.59	0.13	36.8	18.1	2.3	0.49	0.06
YowAI2001	34.2	16.6	0.9	0.48	0.03	51.5	17.6	1.5	0.34	0.03
UvaTrilearn	39.2	20.6	2.8	0.52	0.07	62.2	19.6	2.3	0.31	0.04
Karlsruhe Brains.	27.8	17.2	1.6	0.62	0.06	43.8	17.3	1.2	0.39	0.03
Tsinghuaeolus	36.6	17.7	1.6	0.48	0.04	54.6	21.7	1.4	0.40	0.03
Média	30.6	16.5	2.1	0.54	0.08	47.0	18.1	1.9	0.39	0.05

Tabela 21: Descoordenações na Troca Dinâmica de Posições (10 jogos contra cada equipa)

Para efeitos de avaliação foram considerados unicamente dois tipos de descoordenações e sempre relativos aos próprios jogadores:

- *Desc1*: Descoordenações em que dois jogadores pensam que o seu posicionamento é o mesmo durante 2 a 5 ciclos
- *Desc2*: Descoordenações em que dois jogadores pensam que o seu posicionamento é o mesmo durante 6 ou mais ciclos

Analisando os resultados (tabela 21), facilmente se conclui que as descoordenações de tipo *Desc1* ocorrem com alguma frequência, pois é normal que durante 2 a 5 ciclos os jogadores não vejam o colega de equipa e devido às diferenças nos estados do mundo pensem que o posicionamento de ambos é o mesmo. Verifica-se também que as descoordenações de tipo *Desc2* ocorrem pouco frequentemente (média de 0.08 e 0.05 vezes por cada troca de posições). Usualmente estas descoordenações ocorrem quando ambos os jogadores se encontram a tentar interceptar a bola, e como estão em movimento rápido, olhando para bola, perdem deste modo, precisão na posição dos colegas de equipa. Finalmente, verifica-se que a percentagem de descoordenações na troca de posições é menor no modo DPRE2. Este resultado deve-se a que a função de troca de posições tem um declive superior à do modo DPRE1 e como tal é mais fácil que ambos os jogadores se apercebam no mesmo instante da utilidade de uma troca de posições.

## 9.7 Outros Mecanismos de Coordenação

### 9.7.1 Coordenação por Comunicação Avançada - ADVCOM

O mecanismo de coordenação por comunicação avançada ADVCOM baseia-se na criação de um estado do mundo comunicado, i.e., um estado do mundo construído, utilizando unicamente a informação proveniente da comunicação dos colegas de equipa, sem fazer uso de informação sensorial ou de predição do próprio jogador (figura 71). Da comparação entre a informação disponível neste estado do mundo comunicado e a informação disponível no seu próprio estado do mundo, cada jogador pode em cada instante aferir do interesse que cada item de informação do seu estado do mundo possui para a equipa.

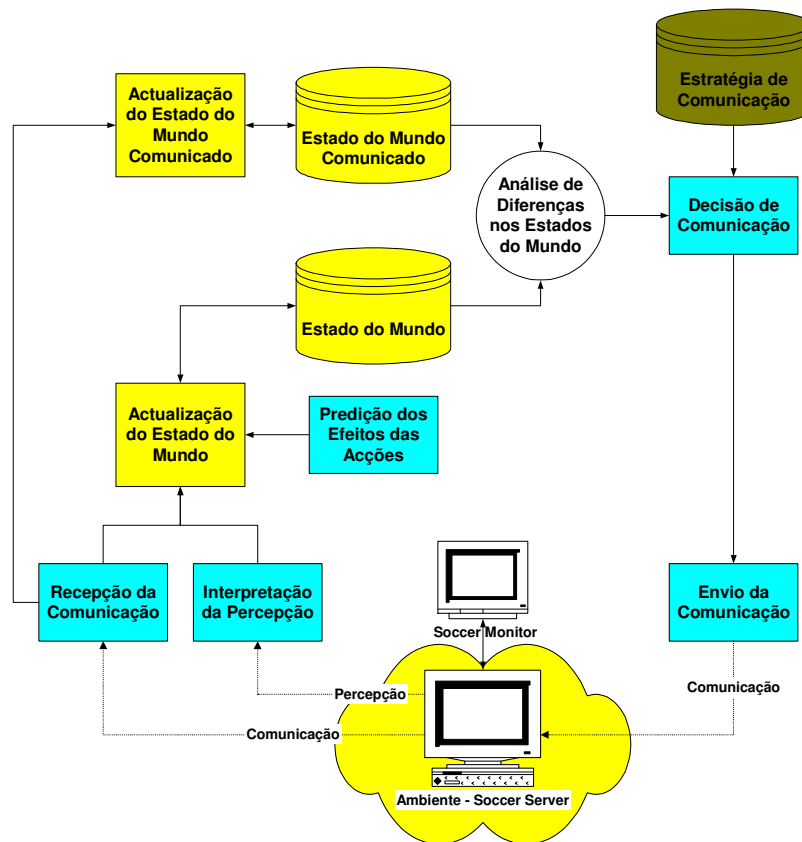


Figura 71: Criação de um Estado do Mundo Comunicado e sua Utilização na Decisão de Comunicação

O algoritmo 18 mostra a utilização da comunicação avançada no domínio do futebol robótico simulado. Tendo em conta a limitada largura de banda disponível, o algoritmo baseia-se em determinar a quantidade de informação a comunicar, e seleccionar para ser comunicada apenas a informação cuja variação entre o estado do mundo comunicado e o estado do mundo normal do agente é mais significativa.

```

ALGORITMO ADVCOM_Communication(WorldState)
    RETORNA Message - Mensagem a comunicar ou Null
PARÂMETROS
    WorldState - Estado do Mundo disponível ao agente
{
// Cria a estrutura da mensagem a comunicar
    Message = CreateMessage
// Dependendo dos parâmetros de comunicação verifica quantas posições pode comunicar
    NPlayers = GetNumberPlayersCommunication()
// Calcula o interesse em comunicar a informação da bola
    BallInterest = CalculateBallCommunicationInterest(Situation, BallPos, BallConf,
        BallNewPos, BallKicked, BallFast, BallFree)
// Dependendo do interesse da bola decide comunicar ou não a bola e baseado nessa decisão
// decide o número de posições de jogadores a comunicar e adiciona a informação da bola à
// mensagem
    SE BallInterest > ThresBallInt ENTÃO {
        NPlayers = ReduceNPlayersBall()
        Message = AddInfoBall(Message)
    }
// Selecciona os jogadores cuja posição vai comunicar
    PARA Team = 1 ATÉ 2 FAÇA
        PARA Jog = 1 ATÉ TeamSize FAÇA {
            PInterest = CalculatePlayerCommunicationInterest(Team, Jog, Situation, BallPos)
            SE NBest(PInterest, NPlayers) ENTÃO Message = AddInfoPlayer(Team, Jog, Message)
        }
// Calcula o interesse total da mensagem (bola e jogadores) e caso este esteja acima de um
// determinado limiar retorna a mensagem
    Interest = CalculateMessageTotalInterest(Message)
    SE Interest > ThresCommunicate ENTÃO RETORNA Message SENÃO RETORNA Null
}

```

*Algoritmo 18: ADVCOM – Comunicação Avançada*

O cálculo do interesse em comunicar a posição dos jogadores é realizado tendo como base a diferença entre o estado do mundo do agente e o estado do mundo criado exclusivamente por comunicação (algoritmo 19). O algoritmo que determina o interesse da comunicação é independente do domínio e considera que interessa comunicar a posição de jogadores:

- Cujas confiança na posição seja superior à confiança comunicada recentemente ou cuja posição já não seja comunicada há algum tempo. Estas condições são consideradas da mesma forma pois a confiança na posição dos jogadores no estado do mundo comunicado é decrementada em cada ciclo e como tal, a confiança na posição de jogadores que não sejam comunicados há algum tempo, será reduzida;
- Cujas distância relativamente à última posição comunicada seja significativa. Se o jogador ainda se encontra na mesma posição que foi comunicada, é pouco interessante comunicar a sua posição novamente.
- Que estejam em posições de interesse na situação actual de jogo. Não interessa considerar jogadores que estejam muito longe da bola ou que estejam atrás da bola quando a equipa ataca. Interessa bastante comunicar a posição do guarda-redes e dos defesas adversários que se encontram na área, numa situação de

ataque perigoso da equipa.

```

ALGORITMO CalculatePlayerCommunicationInterest(Team, PlayerNumber, Situation, BallPos)
  RETORNA Interest - Interesse em comunicar a posição de um dado jogador
PARÂMETROS
  Team - Equipa do jogador
  PlayerNumber - Número do jogador
  Situation - Situação de jogo
  BallPos - Posição da Bola
{
//Se a confiança está abaixo do limiar mínimo então não interessa comunicar a posição
desse jogador
  PlConf = PlayerPositionConf(Team, PlayerNumber)
  PlPos = PlayerPosition(Team, PlayerNumber)
  SE (PlConf < ConfThresholds) ENTÃO RETORNA 0.0
// Distância à posição que foi comunicada pela última vez deste jogador (posição que se
encontra no estado do mundo comunicado)
  DistComm = Distance(PlPos, CommPosition(Team, PlayerNumber))
// Calcula a diferença entre a confiança da posição no estado do mundo e no estado do
mundo comunicado
  DifConf = PlConf - CommConf(Team, PlayerNumber)
// Calcula três factores: 1) Dependente da distância da posição actual do jogador à última
posição comunicada; 2) Dependente da diferença de confiança na posição do jogador do
estado do mundo e do estado do mundo "comunicado"; 3) Dependente do interesse que um
jogador naquela posição terá para o jogo que se encontra numa dada situação com a bola
numa dada posição
  FactDist = CalculateFactorDistance(DistComm)
  FactConf = CalculateFactorConfidence(DifConf)
  FactPosInt = CalculateFactorPosInterest(PlPos, Situation, BallPos)
// Baseado nos três factores calcula o interesse em comunicar a posição do jogador. O
algoritmo utilizado para isto é essencialmente a limitação dos três factores à escala
[1,10] seguida da multiplicação dos mesmos.
  Interest = CalculateInterest(FactDist, FactConf, FactPosInt)
  RETORNA Interest
}

```

*Algoritmo 19: ADVCOM – Interesse em Comunicar a Posição de um Jogador*

O cálculo do interesse em comunicar a posição da bola (algoritmo 20) é baseado em diversas heurísticas específicas do domínio. Interessa comunicar a posição da bola se a confiança na sua posição é elevada, se a bola mudou de posição subitamente, se foi chutada por um jogador ou se se desloca rapidamente no campo.

```

ALGORITMO CalculateBallCommunicationInterest(Situation, BallPos, BallConf, BallNewPos,
BallKicked, BallFast, BallFree)
  RETORNA Interest - Interesse em comunicar a posição e velocidade da bola
PARÂMETROS
  Situation - Situação de jogo
  BallPos - Posição da Bola
  BallConf - Confiança na Posição da Bola
  BallNewPos - Posição da Bola mudou repentinamente
  BallKicked - Velocidade da bola mudou significativamente pelo que esta foi chutada
  BallFast - Bola está em deslocamento rápido no campo (velocidade elevada)
  BallFree - Bola está livre no campo, i.e., nenhum jogador a pode chutar
{
// Se a bola está livre no campo ou o jogador não tem uma elevada confiança na sua posição
então não tem interesse em comunicar a sua posição
  SE BallFree OU BallConf < ConfThresholds RETORNA 0
// Calcula o interesse que a bola tem na presente situação. O interesse será elevado se a
bola mudou de posição ou foi chutada ou desloca-se muito rápido

```

```

BallInterest = CalculateBallInterest(Situation, BallNewPos, BallKicked, BallFast)
// Distância da bola à posição que foi comunicada pela última vez da bola
DistBallComm = Distance(PlPos, BallCommPosition())
// Calcula a diferença entre a confiança da posição da bola no estado do mundo e no estado
do mundo comunicado
DifBallConf = BallConf - BallCommConf()
// Calcula o interesse que comunicar a bola terá, tendo em conta o interesse da bola, a
confiança que o jogador tem na sua posição, a distância da bola à última posição
comunicada e a diferença de confiança entre o estado do mundo e o estado do mundo
comunicado
Interest = CalculateBallCommInterest(BallInteres, BallConf, DistBallComm, DifBallConf)
RETORNA Interest
}

```

*Algoritmo 20: ADVCOM – Interesse em Comunicar a Posição da Bola*

A aplicação do algoritmo de comunicação avançada permite uma diminuição significativa do erro no estado do mundo dos jogadores. Na tabela 22, é analisado o efeito da comunicação avançada no erro médio das posições dos jogadores, utilizando as seguintes estratégias de comunicação avançada:

- **ADVCOM0:** Sem comunicação;
- **ADVCOM1:** Velocidade e Posição da Bola + Velocidade e Posição de Todos os jogadores e respectivas confianças;
- **ADVCOM2:** Velocidade e Posição da Bola + Posição de quatro ou seis jogadores (dependendo se a bola é comunicada) e respectivas confianças;
- **ADVCOM3:** Velocidade e Posição da Bola + Posição de um ou três Jogadores (dependendo se a bola é comunicada) e respectivas confianças.

Na tabela 22 são visíveis os resultados obtidos em 5 jogos contra cada uma das equipas de teste. No cálculo do erro total foi considerado um peso de 5 para o erro na posição da bola relativamente ao erro na posição dos jogadores, e não foram consideradas as posições dos guarda-redes (pois estes estão por vezes muito afastados dos restantes jogadores e como tal a sua posição não é conhecida por estes). Nesta experiência os jogadores que se encontram em comportamento estratégico olham para uma direcção aleatória (entre -90 e 90 graus) e os restantes olham para a bola.

<i>Adversário</i>	<i>ADVCOM0</i>	<i>ADVCOM1</i>	<i>ADVCOM2</i>	<i>ADVCOM3</i>
<i>CMUnited99</i>	5.4	1.5	1.9	3.8
<i>ATTCMU2000</i>	7.1	1.8	3.5	5.4
<i>YowAI2001</i>	8.3	1.7	2.8	5.6
<i>UvaTrilearn</i>	12.1	2.1	3.6	6.2
<i>Karlsruhe Brains.</i>	6.9	1.8	2.4	4.5
<i>Tsinghuaeolus</i>	9.2	1.7	3.2	4.9
<i>Média</i>	8.2	1.8	2.9	5.1

*Tabela 22: Erro Médio no Estado do Mundo com diferentes estratégias de comunicação*

Na tabela é visível que a comunicação das posições e velocidades de todos os jogadores diminui significativamente os erros obtidos no estado do mundo. É também visível que os erros com a estratégia ADVCOM0 (sem comunicação) são muito mais elevados do que os restantes. Os maiores erros verificam-se contra a equipa UVATrilearn [de Boer et al., 2001]. Este facto deve-se à forma rápida do seu jogo (utilizando jogadores heterogéneos e passes longos). Caso, não só os jogadores se movimentam mais no campo, como a bola sofre maiores variações de posição, contribuindo para um maior erro médio final.

Na secção 9.13.4 são apresentados os resultados das experiências realizadas de forma a aferir da influência competitiva do ADVCOM.

### **9.7.2 Coordenação por Modelização Mútua**

A utilização da coordenação estratégica, nomeadamente no que diz respeito ao posicionamento estratégico baseado em situações, permite a cada agente da equipa poder estimar a posição de cada colega de equipa se considerar que este se encontra em comportamento estratégico. Se não existir informação visual ou auditiva precisa e actual relativa à posição de um dado agente da equipa, é possível efectuar a previsão da posição deste agente considerando que o mesmo se desloca para a sua posição estratégica (e consequentemente prevendo a sequência de comandos e trajectória que esse agente irá seguir).

O algoritmo 21 demonstra a modelização estratégica utilizada para os elementos da própria equipa. Este conceito foi posteriormente estendido para a equipa adversária, utilizando uma predição prévia da formação dessa equipa, não estando no entanto este trabalho concluído à data de início da escrita desta tese.

O algoritmo de modelização mútua estratégica verifica, para cada jogador de uma equipa, se a confiança na posição desse jogador é inferior a um valor limiar de confiança (senão não utiliza a modelização mútua estratégica). Em caso afirmativo, verifica qual a posição actual do jogador e a sua posição estratégica e efectua a previsão do comando que o jogador irá emitir no sentido de se deslocar para a sua posição estratégica. A nova posição

para o jogador é depois armazenada no estado do mundo do jogador, substituindo a informação previamente disponível.

```

ALGORITMO StrategicMutualModelling(WorldState, Team)
  RETORNA WorldState - Estado do Mundo
PARÂMETROS
  WorldState - Estado do Mundo disponível ao agente
  Team - Equipa do jogador
{
  PARA Jog = 1 ATÉ TeamSize FAÇA {
  //Se a confiança está abaixo do limiar mínimo então não interessa comunicar a posição
  //desse jogador
    PlConf = PlayerPositionConf(Team, Jog)
    SE PlConf < ThresMutualModelling ENTAO {
      PlPos = PlayerPosition(Team, Jog)
      ModelPos = PlayerStrategicPosition(Team, Jog)
      PlNewPos = PredictPlayerPosition(PlPos, ModelPos)
      StorePlayerPos(WorldState, Team, Jog, PlNewPos)
    }
  }
  RETORNA WorldState
}

```

*Algoritmo 21: Previsão da Posição dos Agentes da Equipa utilizando Modelização Mútua*

Embora o algoritmo seja genérico, diversas heurísticas específicas do domínio foram adicionadas na equipa FC Portugal, no sentido de o tornar mais adequado ao futebol robótico simulado. Para testar a influência da modelização mútua na precisão do estado do mundo dos jogadores foram realizadas experiências com diferentes níveis de limiar de utilização da modelização mútua estratégica (correspondente á confiança na posição do jogador a partir da qual se utiliza a modelização mútua estratégica):

- **MM0:** Sem modelização mútua;
- **MM1:** Modelização mútua com *ThresMutualModelling* = 0.90;
- **MM2:** Modelização mútua com *ThresMutualModelling* = 0.94;
- **MM3:** Modelização mútua com *ThresMutualModelling* = 0.97;

Os resultados obtidos para o erro médio no posicionamento dos jogadores da própria equipa (sem considerar o guarda-redes, a bola e os adversários) em cinco jogos com as equipas de teste, utilizando o esquema de visão descrito anteriormente e sem utilizar comunicação podem ser analisados na tabela 23.



<i>Adversário</i>	<i>MM0</i>	<i>MM1</i>	<i>MM2</i>	<i>MM3</i>
<i>CMUnited99</i>	5.4	4.6	3.2	4.2
<i>ATTCMU2000</i>	5.8	4.9	3.8	5.1
<i>YowAI2001</i>	6.3	4.7	3.7	5.7
<i>UvaTrilearn</i>	7.2	5.8	3.4	5.4
<i>Karlsruhe Brains.</i>	5.9	4.8	4.3	6.1
<i>Tsinghuaeolus</i>	8.2	5.7	4.1	7.4
<i>Média</i>	6.5	5.1	3.8	5.7

*Tabela 23: Erro Médio no Estado do Mundo com diferentes estratégias de modelização mútua*

Analisando a tabela 23 verifica-se que o erro médio mais baixo é obtido utilizando um limiar de 0.94 (estratégia MM2). A utilização de uma estratégia de modelização mútua com um limiar reduzido (MM1) conduz a que a modelização só comece a ser utilizada quando a confiança na posição dos jogadores já é muito reduzida e produz um efeito de atraso no deslocamento dos jogadores para a sua posição estratégica. A utilização de um limiar elevado (MM3) conduz a que a modelização mútua seja utilizada logo após não ter sido visualizado um jogador num dado ciclo. Como tal, por vezes, mesmo quando os jogadores se encontram em comportamento activo e se afastam das suas posições estratégicas, a sua posição no estado do mundo dos restantes jogadores será sempre muito próxima da posição estratégica (com conseqüente incremento de erro relativamente à estratégia MM2). Este incremento do erro é aliás mais acentuado contra as equipas Karlsruhe Brainstormers e Tsinghuaeolus, que são equipas que driblam bastante com a bola e fazem com que os jogadores do FC Portugal abandonem frequentemente as suas posições estratégicas para tentarem tirar a bola aos adversários.

### 9.7.3 Percepção Inteligente

A percepção inteligente baseia-se numa utilização inteligente dos sensores do robô virtual para que a informação que provém destes, em cada instante, permita melhorar a precisão do estado do mundo do agente e incrementar a sua coordenação com os restantes agentes da equipa. Em cada instante os agentes decidem o melhor ângulo de posicionamento do seu pescoço flexível de acordo com a situação de jogo e com o conteúdo do seu estado do mundo de baixo-nível.

Os agentes do FC Portugal podem utilizar o pescoço para olhar de três formas:

- **Visão Activa.** Utilizar o pescoço rotativo de forma a olhar para o local de destino de uma dada acção (por exemplo um passe);
- **Visão da Bola.** De forma a reagir rapidamente a mudanças de velocidade da bola;
- **Visão Estratégica.** De forma a olhar estrategicamente no campo incrementando a

precisão do estado do mundo e maximizando a probabilidade de sucesso das suas acções cooperativas com os restantes agentes.

O algoritmo 22 apresenta o método de decisão entre estas três formas de utilização do pescoço.

```

ALGORITMO DecideLookingDir(WorldState, AgentAction)
    RETORNA LookingDir - Direcção para onde o jogador deve olhar
PARÂMETROS
    WorldState - Estado do Mundo do agente
{
// Se o agente está a executar um passe, um remate ou um passe em profundidade então
determina a direcção para onde vai enviar a bola e caso possa, olha para lá
    SE AgentAction = Shoot OU AgentAction = Pass OU AgentAction = Forward ENTÃO {
        ActionDir = CalculateActionRelativeDir(AgentAction)
        SE PossibleToLook(ActionDir) ENTÃO RETORNA ActionDir
    }
// Se a bola for essencial, i.e., a confiança do jogador na sua posição for baixa, estiver
controlada (ou próximo disso) por um outro jogador, estiver muito próxima do jogador ou
deslocar-se muito rápido, então caso possa (i.e. rodando o pescoço consiga olhar nessa
direcção) o agente olha para a bola
    SE BallEssential() ENTÃO
        BallDir = CalculateBallRelativeDir()
        SE PossibleToLook(BallDir) ENTÃO RETORNA BallDir
// Caso contrário, o agente olha de forma estratégica
    StrategicLookingDir = StrategicLookingMechanism(WorldState, AgentAction)
RETORNA StrategicLookingDir
}

```

#### Algoritmo 22: Decisão da Direcção para onde olhar

A visão estratégica dos agentes é baseada em calcular a direcção, de entre todas as direcções para onde o agente pode olhar, que previsivelmente propiciará um maior incremento na precisão do estado do mundo do agente (algoritmo 23).

```

ALGORITMO StrategicLookingDir(WorldState, Situation, AgentAction)
    RETORNA StrategicLookingDir - Direcção estratégica para onde o jogador deve olhar
PARÂMETROS
    WorldState - Estado do Mundo do agente
    Situation - Situação actual
{
    BestInterest = 0
// Para todos os ângulos possíveis de rotação do pescoço (com uma dada discretização)
calcula a lista de jogadores que estarão visíveis de o jogador olhar nessa direcção (com
uma dada tolerância em termos de distância ao limite do cone de visualização)
    PARA Angle = MinAngle ATÉ MaxAngle FAÇA {
        ListPlayers = CalculateVisiblePlayersDir(Angle, Tolerance)
        TotalInterest = 0
// Para todos os jogadores da lista de jogadores soma o interesse que existe em visualiza-
los. Caso a bola esteja visível na direcção soma um factor de interesse em visualizar a
bola
        PARA Player EM ListPlayers FAÇA {
            Interest = CalculatePlayerVisualInterest(Player, Situation, BallPos)
            TotalInterest = TotalInterest + Interest
        }
    }
    SE BallVisible(Angle) ENTÃO
        TotalInterest = TotalInterest + CalculateBallVisualInterest(Situation, BallPos)
// Caso o interesse seja superior ao máximo anterior então actualiza o máximo e a direcção
respectiva
}

```

```

    SE TotalInterest > BestInterest ENTAO {
        BestInterest = TotalInterest
        StrategicLookingDir = Angle
    }
}
RETORNA StrategicLookDir
}

```

*Algoritmo 23: Percepção Inteligente (SLM – Strategic Looking Mechanism)*

O cálculo do interesse em visualizar um dado jogador ou visualizar a bola é realizado de forma semelhante ao cálculo do interesse em comunicar a posição de um dado jogador realizado no ADVCOM (algoritmo 24). Baseia-se em verificar o aumento de confiança na posição do jogador em causa através da sua visualização e o interesse que esse jogador tem na situação actual.

```

ALGORITMO CalculatePlayerVisualInterest (Player, Situation, BallPos)
    RETORNA Interest - Interesse em visualizar um dado jogador
PARÂMETROS
    Player -Jogador considerado
    Situation - Situação de jogo
    BallPos - Posição da Bola
{
// Calcula a diferença entre a confiança com que será previsivelmente visualizado o
// jogador e a confiança da posição no estado do mundo actual
    DifConf = VisualizationConf (Player) - PlayerPositionConf (Player)
// Calcula o interesse da posição do jogador na situação actual
    FactPosInt = CalculateFactorPosInterest (Player, Situation, BallPos)
// Calcula o interesse global em visualizar o jogador (basicamente multiplica o ganho de
// confiança pelo factor de interesse do jogador na situação actual)
    Interest = CalculateInterest (DifConf, FactPosInt)
    RETORNA Interest
}

```

*Algoritmo 24: SLM – Interesse em Visualizar a Posição de um Jogador*

De forma a aferir da influência dos esquemas de visão na precisão do estado do mundo dos agentes foram realizadas experiências com os seguintes esquemas de visão:

- **SLM0:** Visão Aleatória;
- **SLM1:** Visão da Bola sempre que possível, senão Visão Aleatória;
- **SLM2:** Visão Estratégica simples;
- **SLM3:** Visão da Bola quando esta é essencial, senão Visão Estratégica;

Os resultados obtidos para o erro médio no posicionamento de todos os jogadores (sem considerar os guarda-redes, e considerando um peso de 5 para a bola) em cinco jogos com as equipas de teste sem utilizar comunicação nem modelização mútua podem ser analisados na tabela 24.

<i>Adversário</i>	<i>SLM0</i>	<i>SLM1</i>	<i>SLM2</i>	<i>SLM3</i>
<i>CMUnited99</i>	5.4	6.8	3.7	3.8
<i>ATTCMU2000</i>	7.1	8.3	4.2	4.9
<i>YowAI2001</i>	8.3	9.6	4.1	4.6
<i>UvaTrilearn</i>	12.1	13.4	5.3	5.2
<i>Karlsruhe Brains.</i>	6.9	12.2	4.5	7.4
<i>Tsinghuaeolus</i>	9.2	13.6	5.1	8.1
<i>Média</i>	8.2	10.7	4.5	5.7

*Tabela 24: Erro Médio no Estado do Mundo com diferentes estratégias de Visão Estratégica*

A utilização de visão direccionada sempre para a bola provoca um decréscimo da precisão no estado do mundo. Este facto é sobretudo visível contra as equipas Karlsruhe Brainstormers e Tsinghuaeolus que ao manterem a bola controlada muito tempo sempre no mesmo local fazem com que os agentes do FC Portugal percam precisão relativamente ao que se passa no resto do campo.

A utilização de visão estratégica simples (sem preocupações com a bola) é a que permite obter melhores resultados a nível de erro médio no estado do mundo. De facto, como na região da bola existe usualmente uma grande concentração de jogadores, utilizando este tipo de visão, os jogadores têm tendência a olhar para essa região frequentemente, visualizando a bola e como tal não perdem precisão na informação relativa à bola. No entanto, o facto de por vezes não verem a bola ser chutada (dado que estão a olhar estrategicamente) e o consequente atraso na reacção para uma possível intercepção, faz com que o esquema de visão SLM2 tenha piores resultados competitivos do que o esquema SLM1. Na secção 9.13.4 são apresentados os resultados das experiências realizadas de forma a aferir da influência competitiva do SLM.

## 9.8 Decisão de Alto-Nível

Embora as metodologias de coordenação de equipas de agentes sejam extremamente importantes, para obter bons resultados numa competição como o RoboCup, é necessário também dispor de metodologias de decisão individuais competitivas. O futebol é um jogo muito complexo e o *SoccerServer* consegue modelizar essa complexidade de forma muito realista. Em cada situação, mesmo para um humano com bons conhecimentos de futebol e com uma perspectiva global e sem erros do jogo, é muito difícil decidir o que fazer numa determinada situação: Deverá o agente passar a bola? Para quem? Como? Para que posição especificamente? Deverá o agente driblar? Para onde? Deverá o agente segurar a bola e esperar que os colegas se desmarquem? Deverá tentar rematar à baliza? Para que local na baliza? E se não tiver a bola: Deverá o agente tentar interceptar a bola? De que

forma? Deverá fazer um *tackle*? Deverá marcar um adversário? Para um agente autónomo com perspectiva local e incompleta do mundo esta decisão é ainda mais complexa.

As metodologias de decisão de alto-nível individuais do FC Portugal dividem-se em quatro grupos principais:

- Decisão com Bola (posse de bola);
- Decisão sem Bola (recuperação de bola);
- Decisão com o Jogo Interrompido;
- Decisão do Guarda-Redes.

O algoritmo 25 descreve o método de decisão individual dos agentes. Este algoritmo determina qual o comportamento a seleccionar tendo entre os quatro comportamentos referidos tendo em conta a informação contida no estado do mundo do agente.

```

ALGORITMO IndividualDecision(ActionWorldState) RETORNA PlayerDecision
PARÂMETROS
    ActionWorldState - Estado do Mundo de Selecção de Acções
{
    // Se o jogador não tiver confiança na sua própria posição ou na posição da bola vai
    // efectuar acções conducentes a melhorar a sua localização ou localizar a bola
    SE PlayerPosInvalid OU BallPosInvalid ENTÃO RETORNA Decision_Localize
    // Se o jogo estiver parado então efectua uma decisão de jogo parado
    SE GameStopped ENTÃO RETORNA Decision_GameStopped
    // Se o jogador estiver em situação crítica vai tomar uma decisão activa
    SE CriticalSituation ENTÃO {
        // Se o jogador for o guarda-redes e o seu comportamento actual for de guarda-redes (pois
        // este agente pode-se comportar da mesma forma que os restantes jogadores)
        SE GoaliePlayer E GoalieBehavior ENTÃO RETORNA Decision_Goalie
        // Se o jogador estiver com a posse de bola e no próximo ciclo não for possível obter uma
        // posse mais segura de bola então efectua uma decisão com posse de bola senão efectua uma
        // decisão de recuperação de bola
        SENAO { SE BallPossession() E NOT(BetterPossessionNext()) ENTÃO
            RETORNA Decision_BallPossession
            SENÃO RETORNA Decision_BallRecovery }
    }
    // Se não estiver numa situação crítica então decide estrategicamente, i.e. efectua o
    // posicionamento estratégico
    SENÃO RETORNA Decision_StrategicPositioning
}

```

*Algoritmo 25: Decisão Individual dos Agentes*

O módulo de decisão do guarda-redes foi construído de forma separada pois os processos de decisão deste agente são muito distintos dos processos de decisão dos restantes agentes da equipa. Realça-se o facto de que o agente guarda-redes, pode em qualquer instante assumir o comportamento de um jogador comum, utilizando os processos de decisão

disponíveis para este tipo de agentes<sup>74</sup>. Nesta secção serão descritos, com um nível de detalhe reduzido, os processos de decisão individual dos agentes da equipa FC Portugal.

### 9.8.1 Decisão com Bola

O algoritmo de decisão com bola do FC Portugal é baseado na selecção entre seis tipos de acções possíveis em cada instante:

- **Remate à Baliza (“Shoot”).** Rematar a bola em direcção à baliza na tentativa de marcar um golo;
- **Passe Directo (“Pass”).** Passar a bola com velocidade na direcção aproximada de um colega de equipa. O colega de equipa deverá interceptar a bola ainda em movimento;
- **Passe em Profundidade (“Forward”).** Passar a bola para um ponto do campo onde um colega de equipa (ou o próprio jogador) a poderá recolher (já sem muita velocidade ou mesmo parada);
- **Drible (“Dribble”).** Avançar numa determinada direcção no campo mantendo a bola controlada (ou seja, sempre na área de chuto do jogador);
- **Aliviar a Bola (“Clear”).** Chutar a bola numa direcção em que evite adversários e a retire do local onde está (que usualmente é um local perigoso);
- **Segurar a Bola (“Hold”).** Manter a bola controlada sem que o jogador se movimente no campo, procurando evitar que os adversários mais próximos consigam chutar a bola.

A decisão de qual a acção a executar em cada instante é baseada no cálculo de um conjunto de factores de avaliação para cada acção possível. Os factores de avaliação são posteriormente utilizados no cálculo de uma avaliação global que não é mais do que uma soma pesada dos diversos factores de avaliação (algoritmo 26). Os pesos utilizados nesta soma pesada são diferentes para cada tipo de jogador, permitindo assim a criação de jogadores com comportamentos distintos.

```

ALGORITMO IndividualDecision(Role, ActionWorldState) RETORNA PlayerBallDecision
PARÂMETROS
  Role - Papel atribuído ao jogador na formação actual
  ActionWorldState - Estado do Mundo de Selecção de Acções
{
  Shoots = CalculateNBestShoots(ActionWorldState)
  Passes = CalculateNBestPasses(ActionWorldState)
  Forwards = CalculateNBestForwards(ActionWorldState)

```

<sup>74</sup> Esta potencialidade é aliás utilizada no FC Portugal, nomeadamente em táticas que utilizam a formação designada por 443 (formação essa em que o guarda-redes abandona a baliza e se comporta como um defesa central).

```

    Dribbles = CalculateNBestDribbles(ActionWorldState)
    PlayerBallDecision = DecideBest(Shoots, Passes, Forwards, Dribbles)
}

ALGORITMO DecideBest(Role, Shoots, Passes, Forwards, Dribbles) RETORNA PlayerBallDecision
PARÂMETROS
    Role - Papel atribuído ao jogador na formação actual
    Shoots, Passes, Forwards, Dribbles - Melhores acções de cada tipo
{
// Determina o papel de posse de bola atribuído ao jogador
    RoleBall = GetBallPossessionRole(Role)
// Pesa as decisões utilizando os pesos que o papel atribui a cada tipo de acção
    ShootValue = BestShootValue(Shoots) * RoleShootValue(RoleBall)
    PassValue = BestPassValue(Shoots) * RolePassValue(RoleBall)
    ForwardValue = BestForwardValue(Shoots) * RoleForwardValue(RoleBall)
    DribbleValue = BestDribbleValue(Shoots) * RoleDribbleValue(RoleBall)
// Decide a melhor acção comparando os valores pesados de cada acção
    PlayerBallDecision = GetBest(ShootValue, PassValue, ForwardValue, DribbleValue)
    RETORNA PlayerBallDecision
}

```

Algoritmo 26: Decisão Com Bola (Posse de Bola)

As acções *Hold* e *Clear* não foram utilizadas pelo FC Portugal até agora em competições oficiais (nem foram utilizadas nas experiências realizadas no âmbito desta dissertação) pelo que não serão descritas nesta secção.

### 9.8.1.1 Geração e Avaliação de Passes

A geração dos passes numa determinada circunstância é realizada considerando algumas das linhas de passe possíveis em cada instante. Uma discretização parameterizável destas linhas de passe é utilizada na geração. Dependendo da região do campo, passes para trás não são gerados. Na figura 72 é apresentado um esquema dos passes gerados numa determinada situação e que vão ser sujeitos a avaliação.

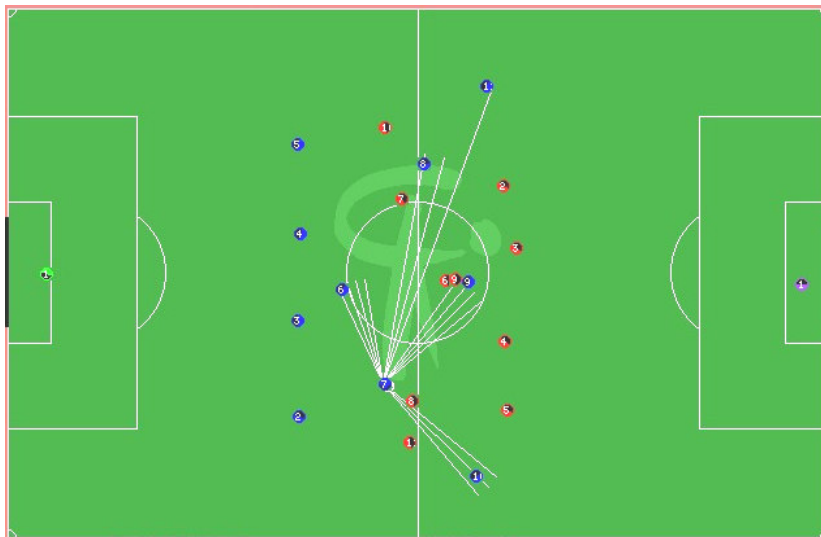


Figura 72: Geração de Passes

Analisando a figura é visível que na região considerada (meio campo defensivo) são gerados unicamente passes que não sejam muito para trás (i.e. com a direcção aproximada da própria baliza) e que tenham a direcção aproximada de jogadores da equipa.

A avaliação dos passes é realizada utilizando as seguintes medidas (consideradas numa escala qualitativa  $\{Very\_Poor, Poor, Medium, Good, Very\_Good\}$  convertível para a escala quantitativa  $\{0,1,2,3,4\}$ ):

- **Valor Posicional (*vpos*)**. Ganho posicional obtido se o passe for bem sucedido. A cada região do campo é atribuído um valor posicional, sendo o valor da região próxima da baliza adversária o mais elevado e o valor da região próxima da própria baliza o mais reduzido. O ganho posicional é obtido pela diferença entre o valor da região de destino do passe e o valor da região de origem.
- **Valor do Cone (*vcone*)**. Cálculo da probabilidade de sucesso do passe em termos das possíveis intercepções dos oponentes que se encontram no cone que une a posição de origem à posição de destino do passe.
- **Valor da Confiança na Posição (*vcp*)**. Confiança na posição do receptor. Se o jogador estiver a olhar para o receptor a confiança será elevada em passes para jogadores que se encontram nas costas do jogador, o valor da confiança será reduzido.
- **Valor da Confiança na Recepção (*vcr*)**. Confiança que o jogador receptor consiga interceptar o passe. Passes efectuados directamente para uma posição muito próxima do receptor terão uma confiança mais elevada do que passes efectuados para uma posição distante do jogador.
- **Valor do Tipo de Recepção (*vtipo*)**. Passes em que o jogador ao realizar a sua intercepção, receba a bola voltado para próximo da baliza adversária têm um valor de tipo superior a passes cuja recepção seja de costas para a baliza adversária, pois permitem avançar mais rapidamente no campo.
- **Valor de Erro (*verro*)**. Valor relacionado com o que acontece no caso do passe chegar próximo do receptor mas ser mal sucedido, i.e., o receptor não conseguir interceptar a bola. No caso de a bola sair do campo, ficar numa região perigosa ou ficar em posse de um adversário, este valor será reduzido. No caso de outros jogadores da equipa poderem emendar um erro do receptor, interceptando a bola, então o valor de erro será elevado.
- **Valor de Distância (*vdist*)**. Passes com uma distância média são preferidos a passes muito curtos ou muito longos. Este facto deve-se a que passes curtos podem ser mal sucedidos devido ao receptor não reagir imediatamente (por exemplo por não ver a bola partir) e passes longos podem ser mal sucedidos pois um pequeno erro no ângulo de saída da bola para o passe pode provocar um grande erro na posição de recepção.



- **Valor do Congestionamento Final (*vcongf*)**. Passes para regiões do campo livres de adversários, i.e., com baixo congestionamento, são preferidos a passes para regiões congestionadas.
- **Valor da Congestionamento Inicial (*vcongi*)**. Congestionamento inicial junto ao ponto de partida da bola na região por onde a bola irá passar de forma a ser enviada para o destino final. Convém evitar as zonas mais congestionadas à saída, para que o passe não seja imediatamente interceptado.
- **Valor do Guarda-Redes (*vgr*)**. Possibilidade de interceptação do guarda-redes adversário tendo em conta a sua qualidade. Esta medida é útil sobretudo em centros para a área.
- **Valor de Chuto (*vhuto*)**. Possibilidade de o passe ser uma assistência que permita ao receptor chutar a bola para a baliza. Se o receptor tiver uma linha de chuto livre para a baliza então o valor de chuto será muito elevado.

A avaliação final de cada passe é a soma pesada de todos os factores referidos, utilizando os pesos definidos no tipo de jogador (posse de bola):

$$\begin{aligned}
 VPass_i &= vpos_i * fpos_p + vcone_i * fcone_p + vcp_i * fcp_p + vcr_i * fcr_p + vtipo_i * ftipo_p + \\
 &verro_i * ferro_p + vdist_i * fdist_p + vcong_f_i * fcong_f_p + vcong_i * fcong_i_p + vgr_i * fgr_p + \\
 &vhuto_i * fhuto_p \\
 \forall i &= 1..npasses \quad p \in \{1..ntipos\}
 \end{aligned}$$

Os melhores passes (os que têm a avaliação mais elevada) são analisados no sistema de avaliação de alto-nível e comparados com os melhores *forwards*, remates e dribles. No sistema implementado na prática no FC Portugal foram também incluídos limites (máximo e mínimo) para cada parâmetro de avaliação.

### 9.8.1.2 Geração e Avaliação de Passes em Profundidade (Forwards), Remates e Dribles

Os passes em profundidade incluem todos os passes que permitam colocar a bola parada em pontos de uma matriz discretizada com dimensão configurável (por defeito de 25 por 40 metros). Na figura 73 é apresentado um esquema explicativo dos possíveis passes em profundidade num dado instante.

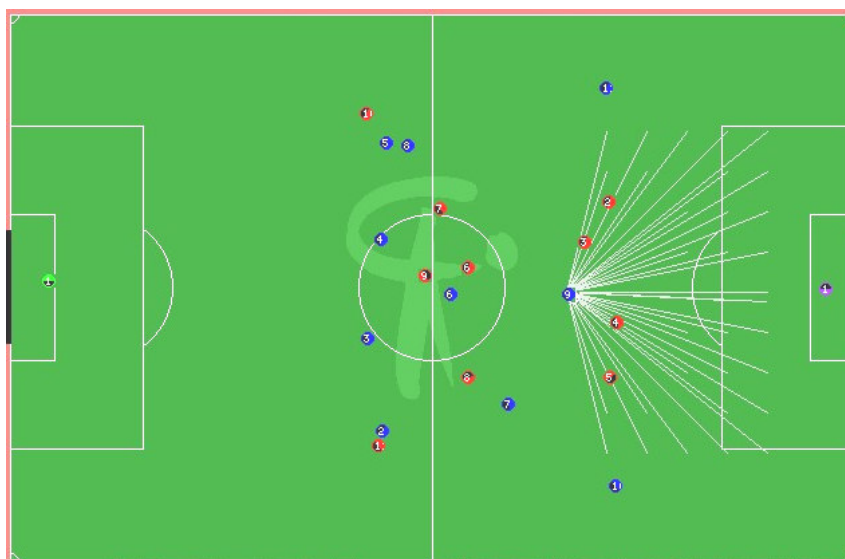


Figura 73: Geração de Passes em Profundidade (Forwards)

Para os remates, o procedimento é semelhante, sendo geradas como possíveis acções os remates para todos os pontos da baliza adversária, espaçados de 0.5 metros.

A avaliação dos *forwards*, remates e dribles é realizada de forma semelhante aos passes, utilizando no entanto parâmetros de avaliação distintos.

Detalhes sobre a avaliação destas acções do FC Portugal podem ser consultados em [Reis e Lau, 2001b] e [Reis e Lau behavior selection, 2002]

### 9.8.1.3 Fusão das Avaliações

A fusão das avaliações deverá ser realizada aplicando uma análise de alto-nível aos parâmetros de avaliação calculados para as  $n$  melhores acções de cada tipo. Na prática, o avaliador utilizado foi simplesmente uma comparação entre o valor final da avaliação da melhor acção de cada tipo pesado utilizando os factores definidos no tipo de jogador para cada acção de posse de bola. As restantes acções foram utilizadas unicamente para efeitos de correcção do código e afinação dos parâmetros de avaliação utilizando a ferramenta *Visual Debugger* cuja descrição resumida se encontra na secção 9.9.1.

### 9.8.1.4 Avaliação da Decisão com Bola

De modo a avaliar a qualidade da decisão com bola foram realizadas algumas experiências em 2000, analisando a qualidade desta decisão aferida por um especialista do domínio<sup>75</sup> [Reis e Lau, 2001b]. Os resultados obtidos na realização de 2 jogos (contra a equipa Mainz Rolling Brains) encontram-se na tabela 25.

<sup>75</sup> O especialista foi um treinador profissional de futebol da primeira liga portuguesa.

<i>Decisão</i>	<i>Nº Ocorrências</i>	<i>Qualidade da Decisão</i>			<i>Qualidade dos Parâmetros</i>		
		<i>Boa</i>	<i>Média</i>	<i>Má</i>	<i>Boa</i>	<i>Média</i>	<i>Má</i>
<i>Pass</i>	191	144 (75%)	34 (18%)	13 (7%)	128 (89%)	7 (5%)	9 (6%)
<i>Forward</i>	136	98 (72%)	23 (17%)	16 (11%)	43 (44%)	40 (41%)	15 (15%)
<i>Dribble</i>	130	62 (48%)	45 (34%)	23 (18%)	44 (71%)	15 (25%)	3 (4%)
<i>Shoot</i>	34	23 (67%)	11 (33%)	0 (0%)	21 (92%)	1 (4%)	1 (4%)
<i>Total</i>	491	327 (67%)	113 (23%)	52 (10%)	236 (72%)	63 (19%)	28 (9%)

*Tabela 25: Resultados experimentais na avaliação da decisão com bola*

A avaliação realizada incidiu sobre a qualidade de cada decisão tomada pelo agente (passe, passe em profundidade, drible ou remate) e pela qualidade dos parâmetros seleccionados na sua execução (velocidade, rapidez, posição). Embora o método de avaliação envolva alguma subjectividade, é visível que a decisão dos agentes é em geral acertada. Os resultados experimentais demonstram que em mais de dois terços dos casos, mesmo possuindo informação local e incompleta os agentes realizam decisões de boa qualidade e escolhem os parâmetros adequados na sua execução. Uma das conclusões interessantes, apontadas pelo especialista do domínio, era que os agentes driblavam demasiado em vez de realizarem passes rápidos ou passes em profundidade. Esta conclusão foi utilizada de forma a melhorar o estilo de jogo em torneios disputados após a realização deste estudo<sup>76</sup> [Reis e Lau, 2001b].

### 9.8.2 Decisão sem Bola

A decisão sem bola da equipa FC Portugal é realizada através da decisão de um dos seguintes comportamentos de alto-nível [Reis e Lau, 2001b]:

- **Intercepção.** Intercepção de uma bola em movimento no ponto do campo que permita a intercepção mais rápida possível ao jogador;
- **Intercepção Passiva.** Intercepção de uma bola em movimento esperando que esta atinja um determinado ponto onde o jogador aguarda a sua chegada;
- **Intercepção Arriscada.** Intercepção com baixa probabilidade de sucesso mas que dependendo dos erros na percepção, movimento e acção, pode eventualmente ser bem sucedida. Este comportamento é seleccionado em casos em que é importante interceptar a bola mas não existe qualquer intercepção segura possível;
- **Aproximação da Bola.** Embora não seja possível interceptar a bola, o jogador

<sup>76</sup> O estudo foi realizado em Maio de 2000.

aproxima-se desta. Este comportamento é utilizado para limitar o espaço disponível a adversários que controlam a bola;

- **Marcação de um Adversário.** O jogador marca um dado adversário, mantendo uma posição próxima deste;
- **Marcação de uma Linha de Passe.** O jogador marca a linha de passe de um adversário para outro adversário, mantendo uma posição em cima desta linha;
- **Marcação de Pontos.** O jogador marca um determinado ponto fixo pré-especificado, mantendo uma posição próxima deste ponto;
- **Desmarcação.** O jogador desmarca-se deslocando-se para um ponto livre de oponentes no campo.

A selecção do comportamento de recuperação de bola é efectuada através de um conjunto de regras que tendo em conta a informação do estado do mundo activo, permite atribuir um valor à execução de cada comportamento de recuperação de bola.

Tal como nos comportamentos de posse de bola, o tipo de jogador influencia significativamente o comportamento do jogador. Na definição do tipo de jogador são incluídos os pesos atribuídos à execução de cada comportamento de recuperação de bola. Tendo em conta os valores calculados para os vários comportamentos e os respectivos pesos definidos no tipo de jogador, são calculados os valores finais associados à execução de cada comportamento de recuperação de bola. O comportamento com o valor mais elevado é seleccionado para execução. Uma descrição mais detalhada dos algoritmos de recuperação de bola do FC Portugal pode ser encontrada em [Reis e Lau, 2001b] [Reis e Lau, 2001c].

### 9.8.3 Decisão do Guarda-Redes

O comportamento do guarda-redes é um dos mais afectados com a inexistência de uma terceira dimensão no futebol robótico simulado. Como não é possível passar a bola sobre o guarda-redes, diversas equipas utilizam estratégias de posicionamento do guarda-redes baseadas na sua colocação muito avançada e próxima dos limites da grande-área. Até ao ano de 2000, dado que a grande maioria das equipas utilizava essencialmente acções individuais dos seus jogadores como forma de chegar próximo da baliza adversária, esta estratégia era bastante eficaz. Desta forma, a estratégia do guarda-redes do FC Portugal 2000 incluía o seu posicionamento numa posição bastante avançada. No entanto, após o ano de 2000, a grande maioria das equipas optou por, tal como o FC Portugal em 2000, efectuar movimentos colectivos de ataque baseados na utilização dos flancos e centros para o interior da área. Deste facto, resultou que esta estratégia de defesa deixou de ser eficaz e o recuo do guarda-redes para uma posição mais típica do futebol real, foi realizada por quase todas as equipas, incluindo o FC Portugal 2001.

O guarda-redes, se a bola estiver próxima, utiliza sempre o modo de visão *narrow*. A utilização deste modo de visão permite-lhe obter informação visual em todos os ciclos o que lhe dá uma muito maior reactividade, essencial para o tipo de tarefa a executar por este agente.

A visão estratégica é unicamente utilizada pelo guarda-redes quando a bola não se encontra muito próxima e está claramente livre (não se encontra ao alcance de qualquer jogador). Nos outros casos, o guarda-redes prefere controlar a posição e velocidade da bola, estando constantemente a olhar para bola.

O guarda-redes do FC Portugal, embora seja idêntico aos restantes agentes da equipa<sup>77</sup>, possui um conjunto de comportamentos bastante distinto dos restantes jogadores, incluindo a acção *catch*, a execução de pontapés de baliza, intercepções desesperadas e intercepções “às cegas”. No anexo 9 é incluída uma descrição dos comportamentos do guarda-redes.

O desenvolvimento do guarda-redes do FC Portugal implicou ainda o desenvolvimento de um comportamento especial que permite que este agente se movimente mantendo sempre que possível a bola visualmente controlada.

Devido ao facto do guarda-redes ser um agente idêntico aos restantes, o seu comportamento pode variar ao longo do jogo e este agente pode, inclusivamente, optar por utilizar um comportamento idêntico aos dos restantes jogadores e jogar de forma idêntica aos restantes jogadores da equipa numa outra posição (avançado, defesa central, etc.). Esta flexibilidade táctica foi testada no nosso laboratório e tendo em conta os bons resultados obtidos (ver secção 9.12) foi explorada pela equipa FC Portugal (também com sucesso) em competições oficiais<sup>78</sup>.

## 9.9 Ferramentas Associadas

No âmbito do projecto FC Portugal foram desenvolvidas cinco ferramentas de análise de jogos e *debug* da equipa:

- *Visual Debugger*;
- *Client Off-Line*;

---

<sup>77</sup> A única diferença do guarda-redes relativamente aos restantes jogadores consiste nos parâmetros da sua configuração inicial.

<sup>78</sup> Em Seattle no campeonato do mundo RoboCup 2001, nos quartos de final, a equipa FC Portugal encontrava-se a perder 0-1 contra os holandeses UvaTrilearn já perto do final do jogo. Após uma mudança táctica que implicou o guarda-redes jogar como defesa central, a equipa conseguiu inverter o resultado e vencer o jogo com o resultado final de 4-1.

- *WSMetrics* - Analisador do Erro do Estado do Mundo;
- Analisador de jogo e cálculo de estatísticas;
- Visualizador 3D.

Devido ao facto de a investigação realizada no desenvolvimento destas ferramentas ser um pouco periférica à investigação central realizada nesta tese<sup>79</sup>, a sua descrição será realizada de modo muito resumido.

### 9.9.1 *Visual Debugger*

O *Visual Debugger* é a ferramenta principal de teste da equipa FC Portugal. Permite analisar o que os agentes da equipa vêem, sentem, ouvem, pensam e fazem ao longo de um jogo de futebol robótico simulado. Foi inicialmente construído através da integração do *SoccerMonitor* tradicional com a ferramenta “*layered extrospection*” desenvolvida pela Carnegie Mellon University [Stone et al., 1999] [Stone et al., 2000b]. A ferramenta foi posteriormente alterada eliminando as dependências de rede e adicionando uma série de novas capacidades.

O *Visual Debugger* baseia-se na gravação, por parte dos jogadores em análise, de *logfiles* especiais no decurso de um jogo, que depois podem ser analisados em conjunto com o *logfile* do jogo gravado pelo simulador. Ao ser visualizado, o filme do jogo é complementado com a informação gravada no respectivo *logfile* por cada jogador incluindo o conteúdo do seu estado do mundo, as suas percepções, decisões e acções. Esta informação pode ser visualizada num formato textual ou sob a forma de símbolos gráficos, incluindo: círculos, quadrados, rectângulos, linhas, cruces. Nas figuras 74 e 75 são visíveis duas imagens do *Visual Debugger*.

---

<sup>79</sup> Aliás, a implementação do *Offline Client* e *WSMetrics* foi realizada essencialmente pelo Nuno Lau, a implementação do *Visualizador 3D* pelo Sérgio Louro e a implementação do *Visual Debugger* foi realizada também em grande parte pelo Nuno Lau.



Figura 74: Interface do Visual Debugger – Decisão de Chuto à Baliza



Figura 75: Interface do Visual Debugger – Decisão sem Bola

O *Visual Debugger* permite ainda visualizar ampliações de regiões do campo. Permite também efectuar comparações entre o estado do mundo dos jogadores e o estado do mundo real. Informação mais detalhada sobre esta ferramenta e uma versão para *download* da mesma pode ser encontrada em [Reis e Lau, 2001d].

### 9.9.2 *Offline Client*

Em diversas situações mesmo com a informação disponibilizada pelo *Visual Debugger*, não era possível compreender as razões de uma determinada decisão de um jogador num dado momento. Para efectuar uma análise de baixo-nível detalhada deste tipo de situações foi desenvolvida a ferramenta *Offline Client*. Esta ferramenta permite efectuar a repetição exacta do comportamento individual de um jogador num dado ciclo do jogo, podendo desta forma utilizar um *debugger* normal (tal como o *gdb*) para examinar o conteúdo das variáveis do jogador, introduzir pontos de quebra no código, etc.

De forma a poder utilizar o *Offline Client*, um *logfile* especial que regista todas as interacções entre o simulador e o cliente, e todos os sinais temporais é gerado e armazenado. Para repetir a execução do jogador, foi implementada uma rotina que lê o conteúdo deste *logfile* e o trata exactamente como se fosse a comunicação via *socket UDP* com o simulador. Sendo conhecida toda a informação recebida do simulador e os instantes temporais em que se deu essa recepção, é então possível repetir a execução do agente e utilizar um *debugger* normal para detectar e corrigir erros de baixo-nível. É também possível alterar o próprio código do agente e efectuar análises do tipo “*What if...*”, analisando o que o agente faria numa determinada situação, utilizando diferentes algoritmos de decisão. Informação mais detalhada sobre esta ferramenta pode ser encontrada em [Reis e Lau, 2001d]

### 9.9.3 **Analizador do Erro no Estado do Mundo**

A disponibilidade de um estado do mundo actualizado e preciso é de extrema importância para o processo de tomada de decisão dos agentes. De forma a analisar a precisão do estado do mundo dos agentes da equipa foi desenvolvida uma ferramenta – *WSMetrics* com o objecto de calcular o erro no estado do mundo de cada agente ao longo de um jogo. O *WSMetrics* inclui rotinas que permitem a construção de dois ficheiros especiais contendo informação em cada ciclo do jogo relativa ao estado do mundo estimado pelo agente e do estado do mundo real utilizado pelo simulador. A ferramenta em si é depois capaz de ler os dois ficheiros referidos e calcular diversos tipos de erros entre o estado do mundo do agente e o estado do mundo real, incluindo erros na posição e velocidade da bola e na posição, velocidade e orientações de todos os jogadores.



Utilizando esta ferramenta é possível testar e comparar diversas abordagens de actualização do estado do mundo dos agentes. É também possível comparar o efeito das estratégias de visão e de comunicação na qualidade do estado do mundo dos agentes.

### 9.9.4 Analisador de Jogo e Cálculo de Estatísticas

O analisador de jogo calcula diversas estatísticas do jogo que podem depois ser utilizadas pelo treinador para definir a melhor tática a utilizar. Entre estas estatísticas destacam-se a localização da bola por região do campo, número de remates às balizas e número de oportunidades de golo de cada equipa.

O analisador de jogo encontra-se actualmente integrado numa ferramenta mais extensa designada por *Team Designer*. O *Team Designer* permite também efectuar a edição gráfica de estratégias de jogo para a equipa FC Portugal (incluindo táticas, formações, e tipos de jogadores). Permite ainda funcionar como treinador *off-line* ensaiando e treinando jogadas estudadas e outras situações de jogo.

### 9.9.5 Visualizador 3D

O visualizador 3D é uma ferramenta definida com o objectivo de permitir a visualização de jogos bidimensionais de futebol robótico simulado num ambiente tridimensional incluindo animação e som. A implementação foi realizada maioritariamente por Sérgio Louro [Louro et al., 2003] como parte da sua tese de Mestrado.

O sistema permite não só a visualização em tempo-real de jogos de futebol robótico simulado (através da conexão ao *soccerserver*) como também efectuar a visualização *off-line* de *logfiles*. O sistema é em si próprio um sistema multi-agente em que cada câmara é modelizada como um agente inteligente autónomo [Louro et al., 2003]. O sistema inclui ainda agentes para o realizador e para o público.

Os agentes “*camaramen*”, de acordo com as suas percepções (posições e velocidades dos jogadores e da bola), decidem a melhor perspectiva de filmagem e *zoom* para a câmara que controlam. Estes agentes enviam informação de alto nível para o agente realizador que baseado nesta informação decide qual a imagem ou imagens a mostrar aos espectadores.

O sistema Virtual 3D inclui como principais potencialidades:

- Ambiente 3D incluindo um estádio e animações realistas;
- Controle de câmara inteligente através de um Sistema Multi-Agente;
- Repetições em câmara lenta dos melhores momentos do jogo;
- Geração de sons 3D baseado em agentes emocionais que representam os espectadores de cada equipa;
- Capacidade de conexão em tempo-real com o *soccerserver* e de leitura de *logfiles*

(versão 2 e 3);

- *Display* de estatísticas de jogo fornecidas por um agente analisador de jogo;

O sistema foi desenvolvido em linguagem C++ para a plataforma Windows XP. Permite a ligação em tempo-real ao *soccerserver*. Permite ainda a ligação em tempo-real ao agente analisador de jogo que fornece estatísticas de jogo que são disponibilizadas utilizando uma interface própria definida para o efeito. A arquitectura do sistema de visualização pode ser analisada na figura 76.

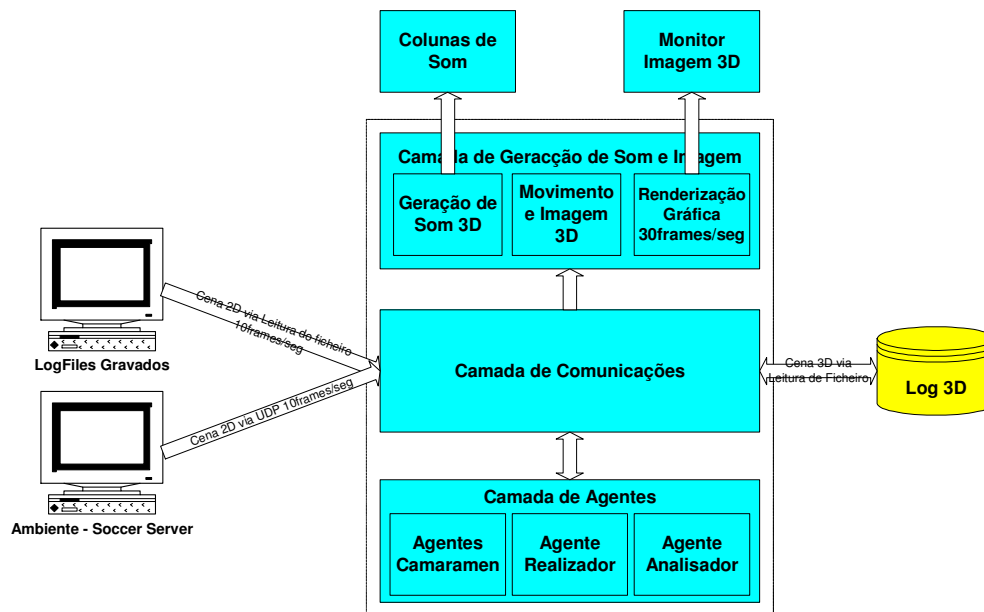


Figura 76: Arquitectura do Visualizador 3D

O sistema permite efectuar a leitura de ficheiros contendo *logfiles* dos jogos ou a ligação directa ao servidor no decurso de um jogo. Permite também efectuar a preparação, gravação e leitura de *logfiles* contendo já informação tridimensional do jogo. A camada de comunicações é responsável por efectuar a ligação entre os diversos módulos do sistema.

A camada de agentes contém três tipos de agentes: *camaramen*, realizador e analisador. Os agentes *camaramen* são responsáveis por decidir os parâmetros de filmagem das suas câmaras (ângulo de abertura, direcção e zoom) e enviar informação de alto-nível sobre a imagem visualizada ao agente realizador. O agente analisador é responsável por calcular diversas estatísticas de jogo e as enviar ao agente realizador. O agente realizador é responsável por decidir em cada instante qual a imagem e estatísticas de jogo a mostrar.

A camada de som e imagem contém as rotinas de geração de som e imagem tendo em conta a câmara utilizada na visualização de jogo e os seus parâmetros seleccionados pelo *camaramen*.



Figura 77: Interface do Visualizador 3D

Embora o visualizador 3D esteja ainda no início do seu desenvolvimento, o impacto do seu primeiro protótipo nos espectadores foi extremamente animador<sup>80</sup>. Os espectadores confirmaram que a visualização de jogos da liga de simulação utilizando o visualizador 3D, transmitia a sensação de observar jogos de futebol real na televisão. Este facto está relacionado não só com o ambiente 3D mas também com as animações realistas, efeitos de som e controlo inteligente de câmara.

## 9.10 A Linguagem *Coach Unilang*

### 9.10.1 Introdução

Embora a disponibilização de táticas flexíveis seja essencial para dispor de uma equipa de futebol robótico robusta e capaz de jogar contra equipas com estratégias distintas, a capacidade de seleccionar a melhor tática em cada instante é essencial. A funcionalidade de selecção da tática pode ser inserida nos jogadores desde que a informação utilizada nesta selecção seja idêntica em todos os jogadores. No entanto, esta informação inclui estatísticas do jogo e informação relativa ao comportamento do oponente que é muito difícil de ser calculada da mesma forma por todos os jogadores.

A linguagem COACH UNILANG permite treinar uma equipa a vários níveis distintos. A linguagem introduz ainda um agente assistente de treinador que é capaz de analisar o jogo (directamente ou lendo um *logfile*) e recolher informação do oponente e informação estatística do jogo a vários níveis. Este agente pode ser usado para ajudar o treinador, fornecendo-lhe informação de alto-nível que possa ser utilizada para decidir a tática a

<sup>80</sup> O primeiro protótipo deste sistema obteve o terceiro lugar na competição de sistemas de visualização e apresentação realizada no RoboCup 2002.

utilizar em cada instante de forma mais eficiente. O treinador assistente pode ser utilizado como um único agente (figura 78a). Pode também ser integrado com o treinador principal, funcionando ambos como um único agente (figura 78b). Esta arquitectura é a utilizada pela equipa FC Portugal actualmente. O treinador principal pode também ser incluído no interior dos agentes jogadores (figura 78c). Se o treinador principal apenas precisar de informação do oponente e estatísticas simples do jogo, provenientes do treinador adjunto, então ele pode ser incluído no interior dos agentes jogadores. Nesta arquitectura, os jogadores são os responsáveis pela mudança da tática da equipa (utilizando todos o mesmo algoritmo). Nesta última arquitectura (figura 78d), o treinador não é um agente autónomo. Esta foi a arquitectura do treinador da equipa FC Portugal em 2000, em que os jogadores utilizaram estatísticas simples do jogo e informação do comportamento do adversário para alterarem a estratégia da equipa.

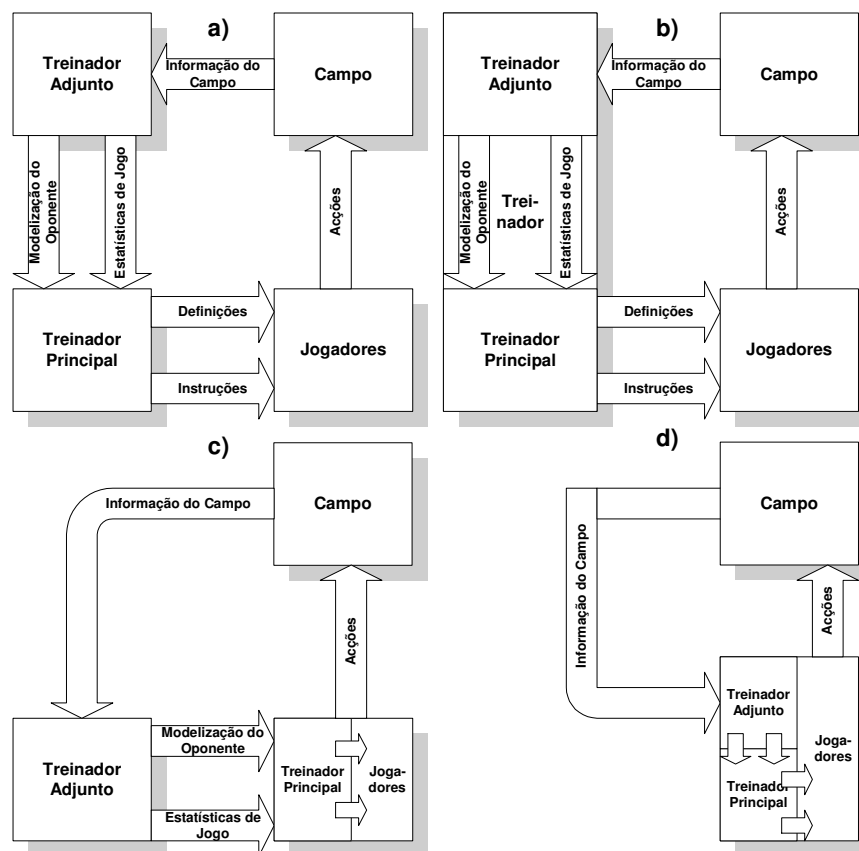


Figura 78: Arquitecturas de treinador possíveis utilizando a linguagem COACH UNILANG

A linguagem COACH UNILANG permite treinar uma equipa a diferentes níveis de abstracção:

- **Nível 1 – Instruções.** Este nível é utilizado para treinar equipas com jogadores inteligentes que estão habituados a jogar juntos. Permite treinar uma equipa a muito alto-nível, mas não é muito flexível porque utiliza conceitos genéricos

fixos de futebol;

- **Nível 2 – Estatísticas e Informação do Oponente.** Este nível é destinado unicamente a auxiliar um treinador a treinar a sua equipa. Esta é a linguagem utilizada pelo treinador adjunto do FC Portugal para comunicar com o treinador principal da equipa<sup>81</sup>. O treinador adjunto está preocupado com o cálculo das estatísticas de jogo como seja a posse de bola por região;
- **Nível 3 – Definições e Instruções.** Este nível de treino é extremamente flexível, uma vez que permite ao treinador definir os conceitos a utilizar na comunicação com os jogadores durante o jogo. Esta comunicação não se limitará a utilizar os conceitos pré-definidos na linguagem permitindo um tipo de treino de muito mais baixo-nível.

A linguagem COACH UNILANG é baseada em diversos conceitos derivados do futebol real. Estes incluem:

- **Regiões do Campo.** Correspondem a áreas do campo com diversas formas possíveis tais como rectângulos, círculos, semi-círculos, quadriláteros, etc;
- **Períodos de Tempo.** Períodos do jogo com uma dada duração e/ou instante de tempo inicial e final;
- **Táticas.** Permitem a configuração a alto-nível do comportamento global da equipa definindo as características ofensivas e defensivas desta, tais como o ritmo de jogo, estilo de ataque e defesa, pressão colocada no oponente, formações utilizadas para diferentes situações, etc;
- **Formações.** Descrevem distribuições espaciais da equipa no campo;
- **Situações.** Análises de alto-nível do estado do jogo. Situações típicas de um jogo incluem o ataque, defesa, passagem da defesa para o ataque, lançamento, pontapé de baliza, canto, etc.;
- **Tipos de Jogadores.** Definem o comportamento individual dos jogadores a diversos níveis. Permitem definir o comportamento dos jogadores com a posse de bola e sem a posse de bola.

### 9.10.2 Requisitos da Linguagem

Na definição da linguagem COACH UNILANG, foram tomados em consideração uma série de requisitos muito semelhantes aos considerados na definição da linguagem SCHEDULING UNILANG (ver secção 7.3). Estes requisitos procuram inserir na

---

<sup>81</sup> Devido ao facto de as regras da liga de simulação não permitirem a utilização de dois agentes como treinadores, na equipa FC Portugal, optamos por implementar os dois agentes como um único agente.

linguagem um compromisso entre generalidade e simplicidade. A linguagem deve ser suficientemente geral de forma a permitir representar as variantes mais comuns das estratégias empregues por treinadores reais de futebol, mas deve ter também uma sintaxe simples e capacidade para permitir treinar equipas a diferentes níveis de detalhe. Os requisitos principais na definição da linguagem COACH UNILANG foram:

- Independência de detalhes de implementação e estratégias de treinadores e jogadores;
- Simplicidade de expansão com novos conceitos e restrições;
- Linguagens e estratégias de equipas de futebol robótico existentes devem ser facilmente traduzidas para o formato COACH UNILANG;
- O formato deve ser simples de ser lido e interpretado por agentes mas também por humanos;
- O formato deve ser multi-resolução e permitir a representação de estratégias de treinadores a diversos níveis de abstracção;
- A semântica deve ser clara. Todos os conceitos e associações devem ser claros para pessoas com conhecimentos básicos de futebol e futebol robótico;
- Possibilidade de representar qualquer tipo de região do campo ou período de tempo e possibilidade de representar as tácticas, formações, situações ou tipos de jogadores mais comuns;
- Regiões, períodos de tempo, tácticas, formações, situações e tipos de jogadores mais comuns devem ser incluídos por defeito mas deverá ser simples definir outros;
- Deve ser o mais simples e compacta possível;
- Deve ser robusta permitindo validações simples de dados e ignorar erros básicos.

Com base nestes requisitos procedeu-se a uma análise detalhada das estratégias das melhores equipas de futebol robótico simulado de forma a procurar encontrar uma linguagem que as permita representar de forma genérica.

### 9.10.3 Especificação Formal

A linguagem COACH UNILANG possui quatro tipos de mensagens:

- **Definição.** Permite ao treinador definir ou alterar um determinado conceito (região, período de tempo, táctica, formação, situação ou tipo de jogador);
- **Estatística.** Permite ao treinador (ou treinador adjunto) informar os jogadores acerca de estatísticas de alto-nível do jogo (tais como número de remates, posse de bola, perdas e recuperações de bola, etc.)

- **Modelo do Oponente.** Este tipo de mensagens é utilizado pelo treinador (ou treinador adjunto) para informar os jogadores acerca do comportamento da equipa adversária ou dos seus jogadores individuais (a diversos níveis). Pode também ser utilizado para fornecer informação acerca do comportamento da sua própria equipa ou dos seus próprios jogadores<sup>82</sup>;
- **Instruções.** Mensagens com instruções são utilizadas pelo treinador para alterar a tática da equipa. Este tipo de mensagem permite alterar as diferentes partes que compõem a tática da equipa (formações utilizadas, ritmo de jogo, estilo de jogo, tipos de jogadores em cada posição, etc.)

A linguagem *COACH UNILANG* foi definida utilizando BNF – BakusNaur form [Naur, 1960]. A definição dos tipos de mensagens utilizadas é a seguinte:

```

<MESSAGE> ::= (<TIME> <ID> <MESSAGE PART> {<MESSAGE PART>})
<MESSAGE PART> ::=
    <DEFINITION_MESS> |
    <STATISTICS_MESS> |
    <OPP_MOD_MESS> |
    <INSTRUCTION_MESS>
<DEFINITION_MESS> ::= (definition <ID> <DEFINITION >)
<STATISTICS_MESS> ::= (statistics <ID> <STATISTICS>)
<OPP_MOD_MESS> ::= (opp_mod <ID> <OPP_MOD>)
<INSTRUCTION_MESS> ::= (instruction <ID> <INSTRUCTION>)

```

Foi também construída uma versão simplificada da linguagem *COACH UNILANG* denominada *COACH UNILANG LIGHT*. Esta versão é compatível na parte de baixo nível com a linguagem *Clang*, sendo deste modo útil para equipas que pretendam evoluir da linguagem de baixo nível *Clang* para uma linguagem que permita a representação de táticas, formações e tipos de jogadores a alto-nível.

### 9.10.3.1 Definição de Conceitos

As mensagens de definição de conceitos permitem definir novos conceitos de forma a serem utilizados como parte da linguagem no decurso de um jogo. São utilizadas essencialmente antes do jogo começar como uma forma de melhorar a capacidade de comunicação entre o treinador e os jogadores durante o jogo (altura em que a largura de banda disponível será muito menor).

Os conceitos possíveis de serem definidos, incluem regiões, períodos de tempo, táticas, formações, situações e tipos de jogadores (comportamento dos jogadores com e sem bola). Os conceitos definidos são identificados por um nome para que possam ser referenciados em futuras mensagens do treinador:

---

<sup>82</sup> Isto é especialmente útil no caso de os jogadores não conhecerem bem os seus colegas de equipa. Neste caso, o treinador, antes ou durante o jogo, poderá informar os seus jogadores das capacidades dos outros jogadores que compõem a equipa e das capacidades e fragilidades da equipa como um todo.

```

<DEFINITION> ::=
    (region <REGION_NAME> <REGION>) |
    (period <PERIOD_NAME> <PERIOD>) |
    (tactic <TACTIC_NAME> <TACTIC>) |
    (formation <FORMATION_NAME> <FORMATION>) |
    (formation_types <FORMATION_TYPES_NAME> <FORMATION_TYPES>) |
    (situation <SITUATION_NAME> <SITUATION>) |
    (player_type <PLAYER_TYPE_NAME> <PLAYER_TYPE> ) |
    (active_type <ACTIVE_TYPE_NAME> <ACTIVE_TYPE>) |
    (recovery_type <RECOVERY_TYPE_NAME> <RECOVERY_TYPE>)

```

Os nomes definidos são simplesmente cadeias de caracteres (*strings*). Por uma questão de simplicidade, e de forma a manter o formato compacto, não devem exceder os 16 caracteres e não devem incluir espaços.

### 9.10.3.2 Regiões do Campo

Para treinar uma equipa de futebol é necessário ser capaz de comunicar em termos de regiões do campo. O conceito de região é um dos mais comuns em futebol e os profissionais de futebol (jogadores, treinadores e jornalistas), usualmente comunicam utilizando termos tais como: “a grande-área”, “a faixa direita do ataque”, “o meio campo ofensivo”, “o miolo do terreno”, etc. Adicionalmente, grande parte das tarefas individuais e cooperativas dos jogadores durante o jogo vai exigir um raciocínio proeminentemente espacial e as acções a efectuar vão depender em larga medida da região do campo onde o jogador se encontra em cada instante. A partilha de uma definição do campo em termos de regiões por todos os elementos da equipa, é essencial para uma boa coordenação da equipa.

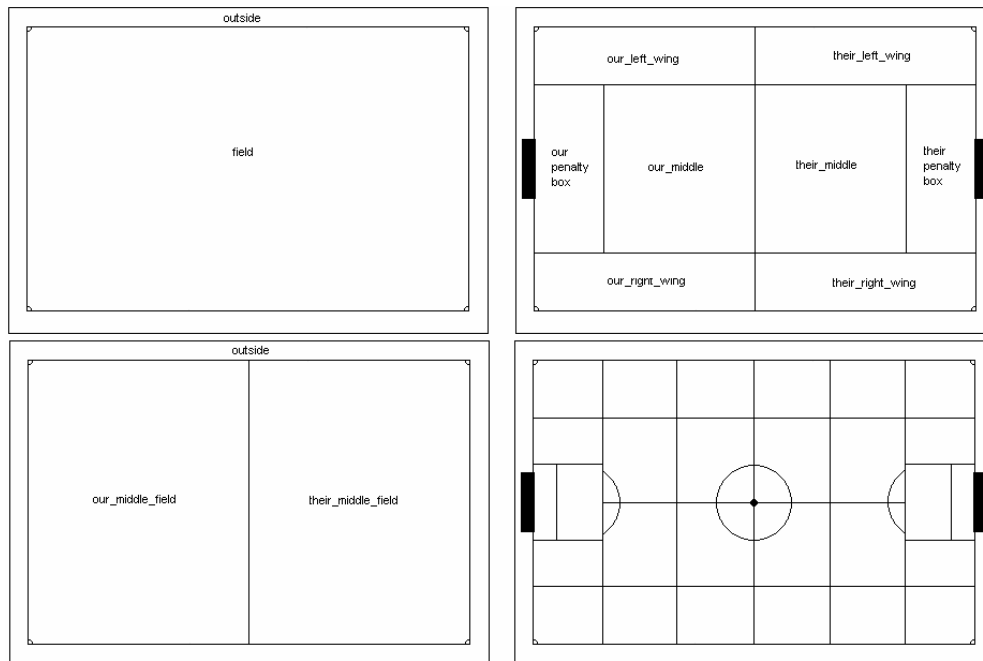


Figura 79: Estrutura Multi-Resolução composta pelas regiões predefinidas na linguagem



COACH UNILANG disponibiliza pré-definidas as regiões mais comuns de um campo de futebol. Disponibiliza também um mecanismo simples para definir novas regiões. As regiões pré-definidas incluídas na linguagem permitem uma descrição do campo em termos de uma estrutura multi-resolução (figura 79).

A definição de novas regiões pode ser efectuada definindo regiões simples (rectângulos, círculos, quadriláteros, etc.), agrupando regiões através da sua união ou intersectando regiões já definidas. As regiões pré-definidas procuram ser o mais simples possível, pelo que unicamente rectângulos foram utilizados na sua definição.

```

<REGION> ::= <SIMPLE_REGION> | (<SIMPLE_REGION> {<SIMPLE_REGION>})
<SIMPLE_REGION> ::= <PREDEFINED_REGION> | <REGION_NAME> | <REGION_DEFINITION>
<PREDEFINED_REGION> ::=
    field | outside | our_middle_field | their_middle_field |
    our_leftwing | our_middle | our_rightwing | our_penalty_box |
    our_leftwing_back | our_leftwing_middle | our_leftwing_front |
    our_rightwing_back | our_rightwing_middle
    our_rightwing_front | our_middle_back | our_middle_front |
    our_middle_back_right | our_middle_back_left | ...
<REGION_DEFINITION> ::= <POINT> | NULL
    (rect <POINT> <POINT>) |
    (tri <POINT> <POINT> <POINT>) |
    (quad <POINT> <POINT> <POINT> <POINT>) |
    (circle <POINT> [real]) |
    (arc <POINT> [real] [real] [real] [real])
<POINT> ::= (point [real] [real])

```

### 9.10.3.3 Períodos de Tempo

O tempo é muito importante nas análises que um treinador tem de efectuar. Grande parte das análises utiliza conceitos como: “na segunda parte”, “no prolongamento”, “nos últimos 5 minutos”, “desde que alterei a forma de atacar da equipa para x”, “desde que efectuei a substituição y”, etc.

COACH UNILANG providencia um mecanismo simples de definição de períodos de tempo. Existem diversos períodos de tempo que são conceitos pré-definidos na linguagem. Para além disso o treinador pode definir períodos de tempo para utilizar posteriormente nas suas mensagens ou pode ainda usar directamente intervalos de tempo nas suas mensagens.

```

<PERIOD> ::= <PREDEFINED_PERIOD> | <PERIOD_NAME> | <PERIOD_DEFINITION>
<PREDEFINED_PERIOD> ::= game | first_half | second_half | extra_time |
    last_1000 | last_500 | last_300 | last_100 | last_50 | last_20
<PERIOD_DEFINITION> ::= (<TIME> <TIME>) | <LENGTH>
<TIME> ::= [integer] | now
<LENGTH> ::= [integer]

```

Os períodos de tempo pré-definidos incluem o jogo completo, a primeira parte, a segunda parte, o prolongamento, os últimos 1000 ciclos, etc. Para definir novos períodos de tempo deve ser definido o seu início e fim ou o seu início e duração. Um identificador especial “now” pode ser utilizado para o final do período. A sua utilização, por exemplo (3000

*now*), significa que o período refere-se ao intervalo de tempo desde o ciclo 3000 até ao instante actual. A definição de um período em termos de duração, por exemplo (100) significa que esse período corresponde aos últimos 100 ciclos de jogo.

### 9.10.3.4 Condições e Situações

Uma parcela considerável dos raciocínios efectuados por um jogador de futebol é baseada numa análise da situação em que se encontra a diversos níveis. Esta situação inclui um vasto conjunto de possíveis condições e análises espaciais. No entanto, de forma a manter a linguagem *COACH UNILANG* simples e compacta, optamos por incluir unicamente um reduzido conjunto de condições adicionado a construtores lógicos simples como forma de definir possíveis situações de jogo.

A definição das situações na linguagem *COACH UNILANG* é baseada no trabalho de Gal Kaminka, Patrick Riley e Timo Stefens na definição das condições a utilizar na linguagem *Clang* [Socccserver, 2001]. As condições incluídas em *COACH UNILANG* são mais simples do que as da linguagem *Clang* pois pretende-se ser capaz de definir unicamente situações de alto-nível (tais como ataque, defesa, etc.) e não situações de baixo-nível que controlem totalmente o comportamento individual dos jogadores como é objectivo da linguagem *Clang*.

```

<SITUATION> ::= <PREDEF_SITUATION> | <SITUATION_NAME> | <SITUATION_DEFINITION>
<PREDEF_SITUATION> ::= attack | defend | defense_to_attack | attack_to_defense |
    our_opportunity | their_opportunity | our_goalie_kick |
    their_goalie_kick
<SITUATION_DEFINITION> ::= <CONDITION>
<CONDITION> ::= (true) | (false) |
    (team_in_possession <TEAM> <PERIOD>) |
    (player_position <TEAM> <UNUM_SET> <REGION>) |
    (ball_position <REGION>) |
    (ball_velocity <VALUE>) |
    (ball_possession <TEAM> <UNUM_SET>) |
    (play_mode <PLAY_MODE>) |
    (not <CONDITION>) |
    (and <CONDITION> <CONDITION>) |
    (or <CONDITION> <CONDITION>)

```

### 9.10.3.5 Tácticas

A táctica é provavelmente o conceito de futebol mais complexo. Na definição de uma táctica, diversos outros conceitos complexos são incluídos, como sejam formações, ritmo de jogo, tipos de jogadores, estilo de jogo, etc. Como foi analisado anteriormente, na nossa definição de estratégia, esta é composta por um conjunto de tácticas e em cada instante, num dado jogo, uma dada táctica está activa. Assim, uma táctica deve permitir definir totalmente o comportamento colectivo da equipa num dado instante. De acordo com o decurso do jogo e com o comportamento da equipa adversária, o treinador da equipa pode alterar a táctica em utilização globalmente ou parcialmente.

A definição de tática na linguagem COACH UNILANG baseia-se em conceitos comuns do futebol tais como formações, tipos de jogadores, mentalidade da equipa, ritmo de jogo, pressão, estilo de jogo ofensivo e defensivo, risco assumido, etc. De forma a incluir estes parâmetros na tática, a sua definição formal é a seguinte:

```

<TACTIC> ::= <PREDEFINED_TACTIC> | <TACTIC_NAME> | <TACTIC_DEFINITION>
<PREDEFINED_TACTIC> ::= normal | score_lots | counter_attack | keep_possession |
    attack_all | defend_all
<TACTIC_DEFINITION> ::= <MENTALITY> <GAME_PACE> <PRESSURE> <FIELD_USE>
    <PLAYING_STYLE> <RISK_TAKEN> <OFFSIDE_TACTIC> <POS_EXCHANGE_USE>
    <FORMATIONS_USED>
<MENTALITY> ::= very_offensive | offensive | normal | defensive | very_defensive
<GAME_PACE> ::= sleep | ease_up | normal | speed_up | give_everything
<RISK_TAKEN> ::= very_safe | safe | normal | risky | very_risky
<PRESSURE> ::= no_pressure | standoff | normal | pressure | high_pressure
<FIELD_USE> ::= wings | middle | balanced | right_wing | left_wing
<PLAYING_STYLE> ::= (<ATTACKING_STYLE> <DEFENDING_STYLE>)
<ATTACKING_STYLE> ::= short_passing | passing | normal | long_ball |
    verylong_ball
<DEFENDING_STYLE> ::= zonal_defense | normal | individual_marking
<OFFSIDE_TACTIC> ::= (<OFFSIDE_TRAP_USE> <OFFSIDE_TRAP_CAREFUL>)
<OFFSIDE_TRAP_USE> ::= <VALUE>
<OFFSIDE_TRAP_CAREFUL> ::= <VALUE>
<POS_EXCHANGE_USE> ::= <VALUE>
<FORMATIONS_USED> ::= ({ (<SITUATION> <FORMATION> <FORMATION_SHAPE>
    <FORMATION_TYPES> ) })
<FORMATION_SHAPE> ::= <FORMATION_WIDTH> <FORMATION_HEIGHT>
<FORMATION_WIDTH> ::= <VALUE>
<FORMATION_HEIGHT> ::= <VALUE>

```

Construir táticas pré-definidas é uma tarefa extremamente complexa. Não existe nenhuma definição aceite por todos do que é uma tática de futebol e que permita descrever a tática de uma dada equipa a determinado momento num jogo. Como exemplo, incluímos diversas táticas pré-definidas relativamente comuns em futebol na linguagem COACH UNILANG: tática normal (*normal*), contra-ataque (*counter\_attack*), posse de bola (*keep\_possession*), ataque (*attack\_all*), defesa (*defend\_all*) e ataque total (*score\_lots*).

A definição completa de uma tática inclui as formações a utilizar para atacar e defender, os tipos de jogadores a utilizar em cada posição do campo e diversos parâmetros que permitem ajustar o comportamento individual da equipa de forma a seguir uma dada estratégia colectiva. Uma tática é então composta por:

- **Mentalidade da equipa (muito ofensiva, ofensiva, normal, defensiva, muito defensiva).** Uma equipa com uma mentalidade ofensiva preocupa-se já com a forma de iniciar o ataque quando está a defender, i.e. sem a posse de bola. Este tipo de equipa usualmente defende com menos jogadores do que uma equipa com mentalidade defensiva. Por exemplo, alguns jogadores podem ficar já perto da linha de meio campo enquanto o adversário ataca. Opostamente, uma equipa com mentalidade defensiva, mesmo enquanto ataca mantém grandes preocupações

defensivas. Os seus jogadores procuram sempre permanecer em posições defensivas úteis que evitem contra-ataques rápidos do adversário.

- **Ritmo de jogo (muito rápido, rápido, normal, lento, muito lento).** O ritmo de jogo está relacionado com a velocidade imposta ao jogo pela equipa quer nos movimentos ofensivos quer nos movimentos defensivos. Equipas com um ritmo de jogo elevado (por exemplo equipas Inglesas típicas), utilizam uma alta velocidade para todos os movimentos efectuados. Equipas com um ritmo de jogo lento (por exemplo a selecção do Brasil ou equipas típicas brasileiras) usualmente movimentam a bola lentamente no campo. Usualmente os jogadores deste tipo de equipas recebem a bola, param a bola e só depois decidem o que fazer a seguir. Os jogadores destas equipas só se movimentam rapidamente no campo em situações críticas para a equipa preferindo em outras situações conservar a sua energia.
- **Risco (Muito Seguro, Seguro, Normal, Arriscado, Muito Arriscado).** Define o risco que os jogadores estão dispostos a assumir nas suas acções defensivas e ofensivas. Se uma tática de alto risco for utilizada, então os jogadores tentarão intercepções arriscadas e passes arriscados. Quando uma equipa está a perder um jogo perto do final, é usual tentar táticas com alto risco de forma a recuperar rapidamente a bola e chegar com perigo à área contrária também rapidamente. A utilização de um baixo grau de risco provoca que os jogadores utilizem movimentos mais seguros e não arrisquem a execução de acções cujo sucesso não é garantido.
- **Utilização do campo.** A utilização do campo refere-se à utilização das diferentes regiões do campo para o desenvolvimento de movimentos ofensivos. As equipas podem utilizar o campo de forma balanceada ou dar preferência à utilização de uma ou ambas as alas. Podem também utilizar prioritariamente o meio do campo como forma de ataque preferencial. Esta situação é típica contra adversários que cobrem bem as alas do terreno, por exemplo utilizando defesas laterais velozes<sup>83</sup>.
- **Estilo de Jogo.** O estilo de jogo decompõe-se em duas partes: o estilo de jogo ofensivo (quando a equipa tem a posse de bola) e defensivo (quando a posse de bola é do adversário). O estilo de jogo define desta forma os estilos de ataque e de defesa utilizado pela equipa.
- **Estilo de Jogo Ofensivo (passes muito curtos, passes curtos, normal, bolas**

---

<sup>83</sup> Esta conclusão é válida não só no futebol real como também no futebol simulado desde a introdução dos jogadores heterogéneos. Diversas equipas como por exemplo os UvaTrilearn utilizaram os jogadores mais rápidos como extremos ou defesas laterais dificultando imenso a tarefa a equipas que utilizavam prioritariamente as alas para atacar.

**longas, bolas muito longas).** O estilo de jogo ofensivo permite definir comportamentos colectivos de ataque distintos. Utilizando novamente como exemplo as equipas da Inglaterra e Brasil, é habitual as equipas Inglesas utilizarem um estilo de jogo com passes longos para as costas dos defesas enquanto as equipas Brasileiras preferem uma progressão mais lenta com fintas e passes curtos.

- **Estilo de Jogo Defensivo.** O estilo de jogo defensivo define o comportamento defensivo da equipa em termos do tipo de marcações utilizadas na zona defensiva;
- **Táctica de Fora de Jogo.** A táctica de fora de jogo está relacionada com as medidas tomadas pelo treinador para apanhar os adversários em fora de jogo e evitar que os jogadores da sua equipa sejam apanhados em fora de jogo. Usualmente a criação de uma armadilha de fora de jogo consiste em criar uma linha composta pelos jogadores mais atrasados da defesa. Os defesas devem manter a linha e avançar rapidamente de forma a procurar colocar os avançados numa posição de fora de jogo.
- **Troca de Posições.** A utilização da troca de posições está relacionada com a tendência que os jogadores terão para trocar de posição com colegas de equipa. Se um elevado grau de trocas de posição for utilizado, os jogadores trocarão de posição com colegas mesmo que o único objectivo de efectuar tal troca seja confundir os adversários. Se um baixo grau de utilização de trocas de posição for utilizado, então os jogadores manterão sempre a mesma posição na equipa.
- **Formações.** Uma das características mais importantes da táctica da equipa está relacionada com as formações a utilizar para as diversas situações de jogo. Por exemplo, uma equipa poderá utilizar uma formação 433 apertada para defender e uma formação larga 235 para atacar. Outras formações poderão ser utilizadas para cantos, pontapés de baliza e outras situações. A linguagem permite ao treinador definir as formações que serão utilizadas em cada situação e a sua forma (em termos de abertura horizontal e vertical).
- **Tipos de Jogadores.** De modo a definir a táctica, é necessário especificar não só as formações utilizadas em cada situação do jogo mas também o comportamento que deve ser assumido por cada jogador em cada posição dessa formação. A definição dos tipos de jogadores permite definir o comportamento individual de cada jogador na formação, atribuindo-lhe um dos comportamentos pré-definidos (*Player Types*). Assim, é possível, por exemplo, definir uma táctica que utiliza um 433 para atacar, cujos jogadores mais avançados da formação são respectivamente um avançado fintador, um avançado rematador e um avançado agressivo.

Embora um conjunto de identificadores seja especificado pela linguagem de forma a ser mais facilmente legível, quase todos os parâmetros permitem utilizar uma escala de valores discretos (*vhi, hi, mid, low, vlow*).

### 9.10.3.6 Formações

As formações descrevem a distribuição espacial dos jogadores em cada situação. COACH UNILANG inclui predefinidas as formações de futebol mais comuns (433, 442, 343, etc.). Inclui ainda um mecanismo de alto-nível de definição de formações baseado num quadro rectangular imaginário de 7\*5 posições. A figura 80 demonstra a utilização deste quadro para definir as formações mais comuns de futebol. As posições por defeito podem também ser definidas sem o auxílio do quadro virtual mas sim, a baixo-nível, utilizando as regiões do campo definidas.

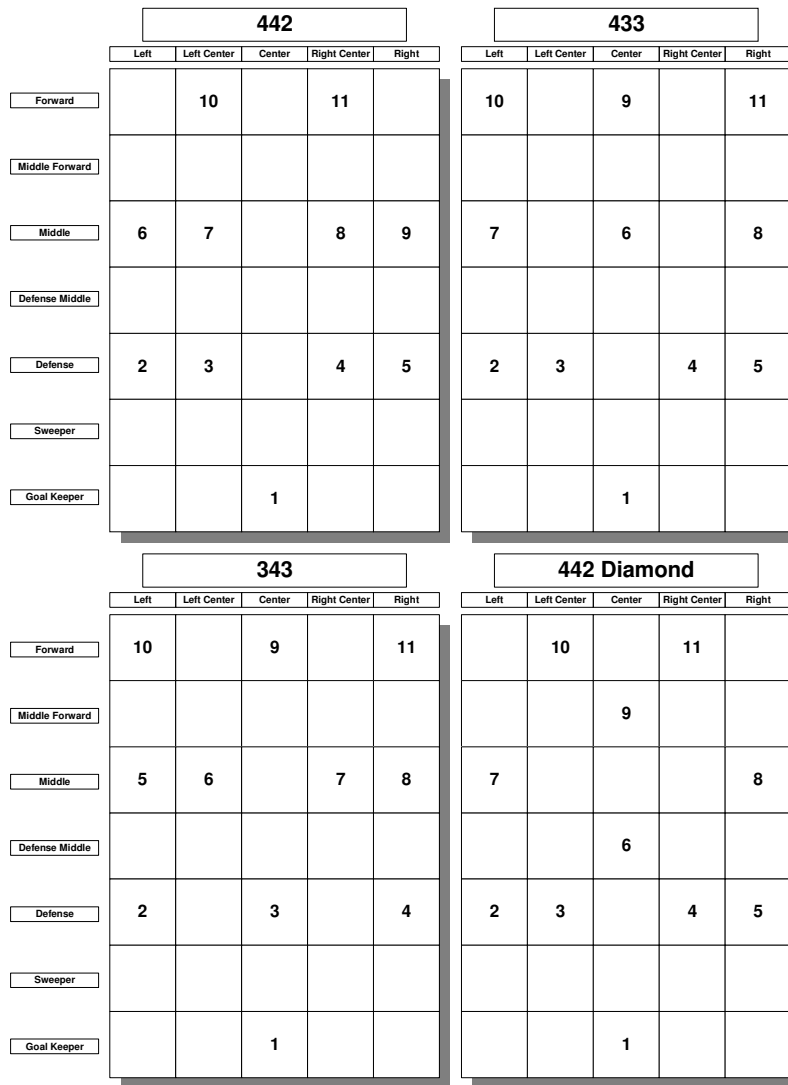


Figura 80: Definição de Formações utilizando o quadro virtual 7\*5

A definição de uma dada formação, implica a definição, para cada jogador, do número do posicionamento a ocupar por defeito na formação<sup>84</sup> e qual a posição correspondente a esse posicionamento no campo. Esta posição pode ser definida com o auxílio do quadro virtual 7\*5 ou utilizando regiões pré-definidas do campo. A definição formal de uma formação é a seguinte:

```
<FORMATION> ::= <PREDEF_FORMATION> | <FORMATION_NAME> | <FORMATION_DEFINITION>
<PREDEF_FORMATION> ::= 433 | 433att | 442 | 442att | 442diamond | 343 | 4123 |
352 | 532 | 41212 | 4132 | 424 | 451 | 541 | 3223 | 3232
<FORMATION_DEFINITION> ::= { (<PLAYER><POSITIONING_NUMBER><PLAYER_POSITIONING> ) }
<PLAYER_POSITIONING> ::= (<VERTICAL_POSITIONING> <HORIZONTAL_POSITIONING> ) |
(<REGION>)
<VERTICAL_POSITIONING> ::= gk | sweeper | defense | defense_middle | middle |
middle_forward | forward
<HORIZONTAL_POSITIONING> ::= left | left_center | center | right_center | right
```

A utilização do número do posicionamento na definição da formação permite descrever a forma de movimentação dos jogadores quando a equipa altera a formação activa. Por exemplo, se a equipa utilizar uma formação 451 para defender e uma formação 433 para atacar, o número do posicionamento permite determinar quais os dois jogadores que passam da linha de meio campo para a linha ofensiva quando a equipa passa para o ataque.

### 9.10.3.7 Tipos de Jogadores

A definição de um tipo de jogador corresponde a definir o comportamento desse jogador quando possui a bola ou quando não é o detentor da posse de bola. A definição dos tipos de jogadores começa pela definição das possíveis acções<sup>85</sup> de posse de bola e de recuperação de bola.

As acções de posse de bola (*action*) incluem as acções definidas anteriormente, ou seja: chutar à baliza (*shoot*), passar a bola directamente a um colega (*pass*) ou para a desmarcação (*forward*), driblar com a bola (*dribble*), correr com a bola (*run*), segurar a bola (*hold*), aliviar a bola (*clear*). É ainda incluída uma acção (*any*) que corresponde a qualquer das acções anteriores.

As acções de recuperação de bola correspondem às acções típicas, sem bola, dos jogadores de futebol, definidas previamente: intercepção (*interception*), corte (*tackle*), marcação de uma linha de passe (*mark\_pass\_line*), marcação de um jogador (*mark\_player*), aproximação da bola (*approach\_ball*) e movimento estratégico no campo respeitando a formação da equipa (*strategical\_move*). É ainda incluída uma acção (*any*) que corresponde a qualquer das acções anteriores.

<sup>84</sup> Eventuais trocas de posicionamentos podem ser permitidas ou mesmo aconselhadas pela tática.

<sup>85</sup> Estas acções são usualmente compostas por um conjunto de comandos básicos (dash, turn, kick, etc.)

Para além da definição das acções de posse e recuperação de bola, a linguagem define ainda os possíveis resultados destas acções. No caso das acções de posse de bola, o resultado pode ser o sucesso (*success*), a interceptção do adversário (*interception*), a perda de bola para um oponente no decurso da execução da acção (*stolen*), bola fora (*out*), a falha na execução da acção (*fail*), a interceptção por parte de um ou outro guarda-redes (*opp\_goalie\_catch* ou *our\_goalie\_catch*) ou qualquer destas acções (*any*). No caso das acções de recuperação de bola o resultado pode ainda ser a interceptção ou o corte por parte de um outro jogador (*stolen\_other* e *interception\_other*) enquanto o jogador considerado estava a efectuar a acção de recuperação de bola.

A execução de uma dada acção implica também a especificação dos parâmetros a serem utilizados para essa acção. Tendo em conta que um dos requisitos da linguagem é o ser genérica e permitir treinar diferentes equipas, e que existe um leque alargado de acções que podem ser executadas pelos agentes, consideram-se unicamente três parâmetros para as acções<sup>86</sup>: distância (*distance*), potência (*power*) e segurança (*safety*).

```
<ACTION> ::= shoot | pass | forward | dribble | run | hold | clear | any
<RECOVERY> ::= interception | tackle | mark_pass_line | mark_player |
    goto_ball | strategical_move | any
<ACTION_RESULT> ::= success | out | interception | stolen | fail |
    opp_goalie_catch | our_goalie_catch | any
<REC_RESULT> ::= <ACTION_RESULT> | stolen_other | interception_other
<ACTION_PARAMETERS> ::= (<DISTANCE> <POWER> <SAFETY>)
<RECOVERY_PARAMETERS> ::= (<DISTANCE> <POWER> <SAFETY>)
```

Para definir um tipo de jogador (*Player\_Type*), o treinador deve definir o comportamento desse jogador com (*Active\_Type*) e sem (*Recovery\_Type*) bola. Alguns comportamentos simples de posse e recuperação de bola encontram-se pré-definidos na linguagem. A definição de novos tipos é bastante simples. No caso dos tipos activos (posse de bola), esta definição baseia-se em definir a tendência (valor numa escala qualitativa) para executar cada possível acção de posse de bola (chutar, passar, driblar, etc.) com início numa dada região do campo (*Region\_From*) e destino numa dada região do campo (*Region\_To*) e determinados parâmetros de execução (distância, potência e segurança). No caso dos tipos de recuperação de bola, estes são definidos de forma similar. Como principal diferença destaca-se que para as acções de recuperação de bola não é considerada a região de destino da acção.

Utilizando tipos de jogadores previamente definidos é possível compôr formações, atribuindo a cada posição da formação um dado tipo de jogador. Uma formação completa de tipos de jogador é composta pela definição dos 11 tipos de jogador que compõe uma dada equipa.

---

<sup>86</sup> Com perda de alguma generalidade e com um aumento significativo da complexidade poderiam ser utilizados os parâmetros utilizados pela equipa FC Portugal para a execução e avaliação de cada uma das acções de recuperação e posse de bola.



```

<FORM_TYPES> ::= { (<PLAYER> <POSITIONING_NUMBER> <PLAYER_TYPE> ) }
<PLAYER_TYPE> ::= <ACTIVE_TYPE> <RECOVERY_TYPE>
<ACTIVE_TYPE> ::= <PREDEF_ACTIVETYPE> | <ACTIVE_TYPE_NAME> | <ACTIVETYPE_DEF>
<PREDEF_ACTIVETYPE> ::= basic | normal | shooter | dribbler | passer
<ACTIVETYPE_DEF> ::= ( { (<ACTION> <REGION_FROM> <REGION_TO>
    <ACTION_PARAMETERS> <VALUE> ) } )
<RECOVERY_TYPE> ::= <PREDEF_RECTYPE> | <RECOVERY_TYPE_NAME> | <RECTYPE_DEF>
<PREDEF_RECTYPE> ::= normal | positional | marker | aggressive
<RECTYPE_DEF> ::= ( { (<RECOVERY> <REGION> <RECOVERY_PARAMETERS> <VALUE> ) } )

```

## 9.10.4 A Linguagem do Treinador Adjunto

O treinador adjunto é responsável por recolher informação estatística do jogo e informação relativa ao comportamento colectivo da equipa adversária e fornecer esta informação, devidamente tratada e formatada, ao treinador principal ou aos jogadores da equipa.

### 9.10.4.1 Informação sobre o Oponente

A informação de modelização do oponente está relacionada com as características detectadas pelo treinador relativas à forma de jogar da equipa ou jogadores adversários [Reis e Lau, 2001c]. Pode também incluir características da própria equipa do treinador (caso este esteja a treinar uma equipa desconhecida).

Na linguagem COACH UNILANG esta informação é dividida em quatro níveis de abstracção [Reis e Lau, 2001c]:

- **Nível Social.** Informação sobre o comportamento colectivo da equipa;
- **Nível Individual.** Decisão individual dos jogadores e guarda-redes;
- **Capacidades de Baixo-nível.** Habilidade na execução das acções sem considerar a adequação da decisão de qual a acção a executar;
- **Características Físicas.** Para os jogadores heterogéneos: velocidade, arranque, inércia, tamanho, etc.

A definição da informação de modelização do oponente pode ser dos quatro tipos descritos e a sua representação formal é a seguinte [Reis e Lau, 2001c]:

```

<OPP_MOD> ::=
    <OPP_TEAM_LEVEL> |
    <OPP_INDIVIDUAL_LEVEL> |
    <OPP_LOW_LEVEL> |
    <OPP_PHYSICAL_LEVEL>

```

No nível social pode ser descrita a qualidade global da equipa adversária (ou da sua própria equipa) e a qualidade numa dada situação específica (defesa, ataque, cantos, etc.). Pode também ser indicada a tática mais próxima da tática utilizada pela equipa que pode incluir, para além dos parâmetros táticos, as formações utilizadas pela equipa para cada situação. A informação é dada na escala qualitativa referida anteriormente (*very\_low ... very\_high*)

```

<OPP_TEAM_LEVEL> ::=
    (opp_team_quality <TEAM> <PERIOD> <VALUE>) |
    (opp_team_situation_quality <TEAM> <SITUATION> <PERIOD> <VALUE>) |
    (opp_team_tactic <TEAM> <PERIOD> <TACTIC>)
    (opp_team_formation <TEAM> <PERIOD> <SITUATION> <FORMATION>)

```

A informação relativa aos jogadores individuais inclui a qualidade global dos jogadores, a sua capacidade para trocar de posicionamento com colegas, capacidade de pressionar oponentes, capacidade de reacção (a eventos no campo) e agressividade. Inclui também informação relativa aos mecanismos de decisão individual dos jogadores com e sem bola e aos mecanismos de decisão individual do guarda-redes: distância usual à baliza (numa escala qualitativa), qualidade de reposição de bola, qualidade de intercepção, qualidade de defesa (de chutos à baliza), qualidade de intercepção de cruzamentos, etc.

```

<OPP_INDIVIDUAL_LEVEL> ::=
    (opp_individual <OPP_INDIVIDUAL_INFO_TYPE> <TEAM> <PLAYER> <PERIOD>
    <VALUE>) |
    (opp_action_quality <ACTION> <TEAM> <PLAYER> <PERIOD> <VALUE>) |
    (opp_recovery_quality <RECOVERY> <TEAM> <PLAYER> <PERIOD> <VALUE>) |
    (opp_goalie <OPP_GOALIE_INFO_TYPE> <TEAM> <PERIOD> <VALUE>)
<OPP_INDIVIDUAL_INFO_TYPE> ::= player_overall_quality |
    player_pos_changing_level | player_pressing_capability |
    player_reaction_capability | player_agressivity
<OPP_GOALIE_INFO_TYPE> ::= goalie_distance_to_goal | goalie_reposition_quality |
    goalie_interception_quality | goalie_defense_quality |
    goalie_leaves_goal_toward_player | goalie_leaves_area |
    goalie_intercept_crosses

```

A informação relativa às capacidades de baixo-nível inclui a indicação (em termos qualitativos) da qualidade de movimentação, intercepção, chuto, drible e visão dos jogadores:

```

<OPP_LOW_LEVEL> ::=
    (opp_low <OPP_LOW_INFO_TYPE> <TEAM> <PLAYER> <PERIOD> <VALUE>)
<OPP_LOW_INFO_TYPE> ::= player_moving_quality | player_dribbling_quality |
    player_kick_quality | player_dribbling_quality |
    player_vision_quality

```

A informação física inclui a indicação do tipo de jogador heterogéneo ou tipo de visão utilizado por um dado jogador. Inclui também os parâmetros individuais relacionados com os jogadores heterogéneos:

```

<OPP_PHYSICAL_LEVEL> ::=
    (opp_physical <OPP_PHYSICAL_INFO_TYPE> <TEAM> <PLAYER> <PERIOD>
    <VALUE>) |
    (player_het_type <TEAM> <PLAYER> <PERIOD> <HET_TYPE>) |
    (player_vision_type <TEAM> <PLAYER> <PERIOD> <VIEW_TYPE>)
<OPP_PHYSICAL_INFO_TYPE> ::= player_physical_condition | player_speed |
    player_recovery | player_decay | player_inertia |
    player_acceleration | player_size | player_kickable_margin |
    player_kick_accuracy

```

Utilizando este formato é possível especificar a todos os níveis (desde o social até ao físico) o comportamento de uma dada equipa e dos seus jogadores.

### 9.10.4.2 Estatísticas de Jogo

A informação estatística do jogo é muito importante para treinar uma equipa de futebol (robótico). Esta informação contrariamente à informação de modelação do oponente não está directamente relacionada com o comportamento do oponente mas sim com o curso do jogo a diversos níveis. A informação estatística incluída em COACH UNILANG baseia-se na informação utilizada por treinadores reais e na informação disponibilizada por sistemas computadorizados de análise de jogos de futebol [Reis e Lau, 2001c] como o *Match Analysis* [Match Analysis, 2001]. A definição da informação estatística em COACH UNILANG é [Reis e Lau, 2001c]:

```
<GAME_PLAYMODE> ::= playon | corner | offside | kickin | kickoff | free_kick |
goalie_free_kick | goal_kick | any
<STATISTICS> ::= (clear) | (game_time <TIME>) |
(game_result <PERIOD> [integer] [integer]) |
(stopped_time <PERIOD> <TIME>) |
(game_occurrence <GAME_PLAYMODE> <REGION> <TEAM> <PERIOD> <COUNT>) |
(action <ACTION> <REGION_FROM> <REGION_TO> <TEAM> <PLAYER> <PERIOD>
<ACTION_RESULT> <COUNT>) |
(recovery <RECOVERY> <REGION> <TEAM> <PLAYER> <PERIOD> <INT_RESULT>
<COUNT>) |
(ball_possession <REGION> <TEAM> <PLAYER> <PERIOD> <COUNT>) |
(player_position <REGION> <TEAM> <PLAYER> <PERIOD> <COUNT>) |
(attack <REGION_FROM> <REGION_TO> <TEAM> <PERIOD> <NPASSES>
<ACTION_RESULT> <COUNT>) |
(assist <REGION_FROM> <REGION_TO> <TEAM> <PLAYER> <PLAYER_TO>
<PERIOD> <ACTION> <ACTION_RESULT> <COUNT>) |
(ball_losses <ACTION> <REGION> <TEAM> <PLAYER> <PERIOD> <COUNT>) |
(ball_recoveries <RECOVERY> <REGION> <TEAM> <PLAYER> <PERIOD>
<COUNT>) |
(action_to_player <ACTION> <REGION_FROM> <REGION_TO> <TEAM> <PLAYER>
<PLAYER_TO> <PERIOD> <ACTION_RESULT> <COUNT>)
```

A informação estatística é muito importante para decidir a tática a aplicar em cada momento. A maioria das equipas do RoboCup utiliza no entanto unicamente o tempo e o resultado do jogo na decisão de qual a tática a utilizar. No entanto, estes dois itens não são suficientes para decidir, dados os objectivos da equipa para o jogo, qual a tática a utilizar. COACH UNILANG inclui a possibilidade de transferir entre o treinador adjunto e o treinador ou de qualquer destes agentes para os jogadores, a seguinte informação estatística, calculada por região do campo, equipa, jogador e período de tempo:

- **Tempo de Jogo (“*Game Time*”).** Tempo total de jogo;
- **Resultado do Jogo (“*Game Result*”).** Resultado do jogo (golos marcados por cada equipa) num dado período;
- **Tempo de Jogo Parado (“*Stopped Game Time*”).** Tempo total em ciclos que o jogo esteve parado;
- **Ocorrências do Jogo (“*Game Occurrences*”).** Ocorrências de situações tais como cantos, lançamentos ou foras de jogo, sua contagem e tempo total. Por exemplo (*game\_occurrence offside their\_penalty\_box our\_first\_half 3*), indica que 3

foras de jogo ocorreram na primeira parte do jogo no interior da grande-área adversária;

- **Número de vezes que cada acção de posse de bola foi executada e resultados (“Action”).** Indica, por região, jogador e período, o número de acções com bola de cada tipo executadas e o seu respectivo resultado. Por exemplo: (*action pass field field our any game success 20*) indica que a nossa equipa executou 20 passes com sucesso durante todo o jogo;
- **Número de vezes que cada acção de recuperação de bola foi executada e resultados (“Recovery”).** Indica, por região, jogador e período, o número de acções sem bola de cada tipo executadas e o seu respectivo resultado. Por exemplo: (*recovery interception their\_middle\_field our 10 last\_300 fail 4*) indica que o jogador 10 da nossa equipa falhou 4 intercepções durante os últimos 300 ciclos de jogo;
- **Posse de Bola (“Ball Possession”).** Estatísticas de posse bola (em termos de número de ciclos) por equipa e jogador, em cada região e período. A frase (*ball\_possession their\_left\_wing our any game 250*) indica que a nossa equipa teve a posse de bola durante 250 ciclos na região *offensive\_left\_wing* durante o jogo;
- **Posições dos Jogadores (“Players Positions”).** Indica o número de ciclos que cada jogador esteve em cada região do campo em cada período.
- **Número de Ataques por Tipo e Resultado (“Attack”).** Fornece a informação relativa aos ataques de cada equipa, suas regiões iniciais e finais e tipo (em termos do número de passes entre jogadores realizados) e o seu resultado final. Por exemplo: (*attack field field our game 3 opp\_goalie\_catch 4*), indica que a equipa realizou 4 ataques no jogo com apenas dois passes cada que resultaram em defesas do guarda-redes adversário;
- **Assistências (“Assist”).** Assistências (i.e. passes para remates à baliza) realizadas por cada equipa e jogador, região de partida, região de destino e período. É ainda permitido incluir a acção que resultou na assistência e o resultado final da acção de remate à baliza. Por exemplo: (*assist their\_left\_wing their\_penalty\_box any any game pass goal 3*) indica que foram realizadas três assistências para remates que resultaram em golo, utilizando passes a partir da ala esquerda do ataque da equipa para o centro da grande-área.
- **Perdas de Bola (“Ball Losses”).** Perdas do controlo da bola (para o adversário) por equipa, jogador, região e período de tempo. Pode também ser incluída a acção executada no momento da perda da bola. A frase (*ball\_lossess pass their\_middle\_field our any last\_1000 10*) indica que 10 bolas foram perdidas no meio campo do adversário, ao realizar passes nos últimos 1000 ciclos.
- **Recuperações de Bola (“Ball Recoveries”).** Informação relativa à forma como

cada recuperação de bola foi executada pelas equipas. Permite incluir informação relativa à forma como cada recuperação de bola foi executada por cada equipa. A frase (*ball\_recoveries tackle our\_middle opponent any game 10*) indica que a bola foi recuperada 10 vezes pelo adversário no nosso meio campo, executando *tackles*, durante o jogo.

- **Circulação de bola (“Action To Player”).** Permite indicar as acções executadas por cada jogador e para cada jogador, por região e período e os seus resultados. Por exemplo: (*action\_to\_player pass field field our 8 9 second\_half successful 10*) indica que o jogador 8 fez 10 passes bem sucedidos para o jogador 9 durante a segunda parte do jogo.

O cálculo desta informação estatística não é no entanto trivial mesmo para um agente com visão global e sem erros como é o caso do treinador. Este cálculo implica transformar as acções de baixo nível executadas pelos jogadores (*dashes, turns, kicks e catches*) e a posição e velocidade da bola, em acções de alto-nível tais como intercepções ou passes.

### 9.10.5 Instruções do Treinador

O treinador pode enviar diversas instruções, nomeadamente tácticas, formações e comportamentos individuais de jogadores.

```
<INSTRUCTION> ::=
    (change_tactic <TACTIC>) |
    (change_mentality <MENTALITY>) |
    (change_game_pace <GAME_PACE>) |
    (change_risk_taken <RISK_TAKEN>) |
    (change_field_use <FIELD_USE>) |
    (change_playing_style <PLAYING_STYLE>) |
    (change_pressure <PRESSURE>) |
    (change_offside_tactic <OFFSIDE_TACTIC>) |
    (change_pos_exchange_use <POS_EXCHANGE_USE>) |
    (change_formation <SITUATION_NAME> <FORMATION> <FORMATION_SHAPE>) |
    (change_formation_pos <FORMATION_NAME> <POSITIONING_NUMBER>
     <PLAYER_POSITIONING> <PLAYER_TYPE>) |
    (change_player_type <PLAYER_TYPE_NAME> <ACTIVETYPE_DEF>
     <RECTYPE_DEF>)
```

A coordenação por controlo parcialmente hierárquico baseou-se na criação de um treinador *online* capaz de enviar instruções aos jogadores da equipa.

### 9.10.6 A Linguagem *COACH UNILANG LIGHT*

De forma a permitir uma utilização mais alargada da linguagem *COACH UNILANG* foi definida uma versão básica da linguagem designada *COACH UNILANG LIGHT*. Esta versão inclui unicamente as características fundamentais da linguagem *COACH UNILANG* e é compatível nas condições de baixo-nível com a linguagem de baixo nível *CLang*. A sua definição completa é a seguinte:

```

<MESSAGE> ::= <ADVICE_MSG> | <DEFINE_MSG> | <FREEFORM_MSG>
<ACTION_TOKEN> ::= mp | md | mc | ms | mf | mr | mh | ma |
    ri | rt | rl | rp | rb | rm | rf | ra
<ACTION> ::= (<ACTION_TOKEN> <REGION> <UNUM_SET>)
<ACTION_LIST> ::= {<ACTION>}
<MODE> ::= highdo | do | neutral | dont | highdont
<DIRECTIVE> ::= (<MODE> <UNUM_SET> <ACTION_LIST>)
<DIRECTIVE_LIST> ::= {<DIRECTIVE>}
<RULE> ::= (<CONDITION> <DIRECTIVE_LIST>) | <RULE_NAME>
<PLAYER_ROLE> ::= {<RULE>}
<FORM_TOKEN> ::= (<POS_NUM> <UNUM> <REGION> <PLAYER_ROLE>) | <FORM_TOKEN_NAME>
<FORMATION> ::= {<FORM_TOKEN>}
<TACTIC_TOKEN> ::= (<CONDITION> <FORMATION>) | <TACTIC_TOKEN_NAME>
<TACTIC> ::= {<TACTIC_TOKEN>}
<TOKEN> ::= <RULE> | <PLAYER_ROLE> | <FORMATION> | <TACTIC> | clear
<ADVICE_TOKEN> ::= (<ID> <UNUM_SET> <TOKEN>)
<ADVICE_LIST> ::= {<ADVICE_TOKEN>}
<ADVICE_MSG> ::= (advice <ADVICE_LIST>)
<DEFINE_MSG> ::= (define <DEFINE_TOKEN_LIST>)
<DEFINE_TOKEN> ::= (definec <CONDITION_NAME> <CONDITION>)
    | (definer <REGION_NAME> <REGION>)
    | (defined <DIRECTIVE_NAME> <DIRECTIVE>)
    | (defineu <RULE_NAME> <RULE>)
    | (definep <PLAYER_ROLE_NAME> <PLAYER_ROLE>)
    | (definef <FORMATION_NAME> <FORMATION>)
    | (definet <TACTIC_NAME> <TACTIC>)
<DEFINE_TOKEN_LIST> ::= {<DEFINE_TOKEN>}
<CONDITION> ::= (true) | (false)
    | (ppos <TEAM> <UNUM_SET> <REGION>) | (bpos <REGION>)
    | (bowner <TEAM> <UNUM_SET>) | (pmode <PLAY_MODE>)
    | (and <CONDITION_LIST>) | (or <CONDITION_LIST>) | (not <CONDITION>)
    | <NAME>
<CONDITION_LIST> ::= {<CONDITION>}
<REGION> ::= <POINT> | (null) | (rec <POINT> <POINT>) #rectangle
    | (arc <POINT> [real] [real] [real] [real]) | (reg <REGION_LIST>)
    | "STRING"
<POINT> ::= (pt [real] [real]) | (add <POINT> <POINT>) | (mult <POINT> <POINT>)
    | (pt ball) | (pt <TEAM> <UNUM>)
<PLAY_MODE> ::= bko | time_over | play_on | ko_our | ko_opp | ki_out |
    ki_opp | fk_our | fk_opp | ck_our | ck_opp | gk_our | gk_opp |
    gc_our | gc_opp | ag_our | ag_opp
<TEAM> ::= our | opp
<POS_NUM> ::= [1..11]
<UNUM> ::= [1..11]
<UNUM_LIST> ::= {<UNUM>} | e
<NAME> ::= "[string]"
<FREEFORM_MSG> ::= (freeform "[string]")

```

Esta linguagem foi utilizada como base para a reformulação da linguagem CLang que foi utilizada na competição de treinadores realizada em Fukuoka no RoboCup 2002.

### 9.10.7 Conclusões

A linguagem COACH UNILANG foi a primeira tentativa para criar uma linguagem que permita o treino a alto-nível de uma equipa de futebol robótico. Claro que ainda existem diversos tópicos abertos e a sintaxe não permite representar tudo o que é necessário para

definir todas as táticas de futebol. Os tipos de jogadores não são completos pois diferentes equipas podem ter acções específicas de posse de bola e recuperação de bola que não estão incluídas na linguagem. A linguagem inclui as acções mais comuns do futebol simulado/real com e sem bola e arquitecturas específicas de jogadores não devem ser incluídas numa linguagem standard com esta.

Pode ser argumentado que a definição de táticas e formações é de muito alto-nível. No entanto a linguagem COACH UNILANG permite a definição de táticas e formações a qualquer nível de detalhe desejado pelo treinador. O treinador pode utilizar unicamente instruções de alto-nível mas pode também definir o posicionamento e comportamento de cada jogador com elevado detalhe.

Outra crítica que pode ser realizada a esta linguagem é que é demasiada complexa. No entanto o futebol é um jogo extremamente complexo e o SoccerServer captura muito bem essa complexidade. Não existe qualquer interesse em definir uma linguagem que permite treinar unicamente equipas com mecanismos de decisão muito simples que não sejam capazes de jogar como equipas reais de futebol.

A linguagem está actualmente em extensão de forma a incluir planos flexíveis, marcadores de livres e cantos, instruções para o guarda-redes, estratégias de visão e estratégias de comunicação.

## **9.11 Coordenação por Controle Parcialmente Hierárquico**

### **9.11.1 Definição de um Treinador on-line**

A coordenação por controle parcialmente hierárquico implica a construção de um treinador para a equipa FC Portugal, capaz de comunicar com os jogadores utilizando a linguagem COACH UNILANG [Reis e Lau, 2003b]. A arquitectura definida para o agente treinador encontra-se representada na figura 81.

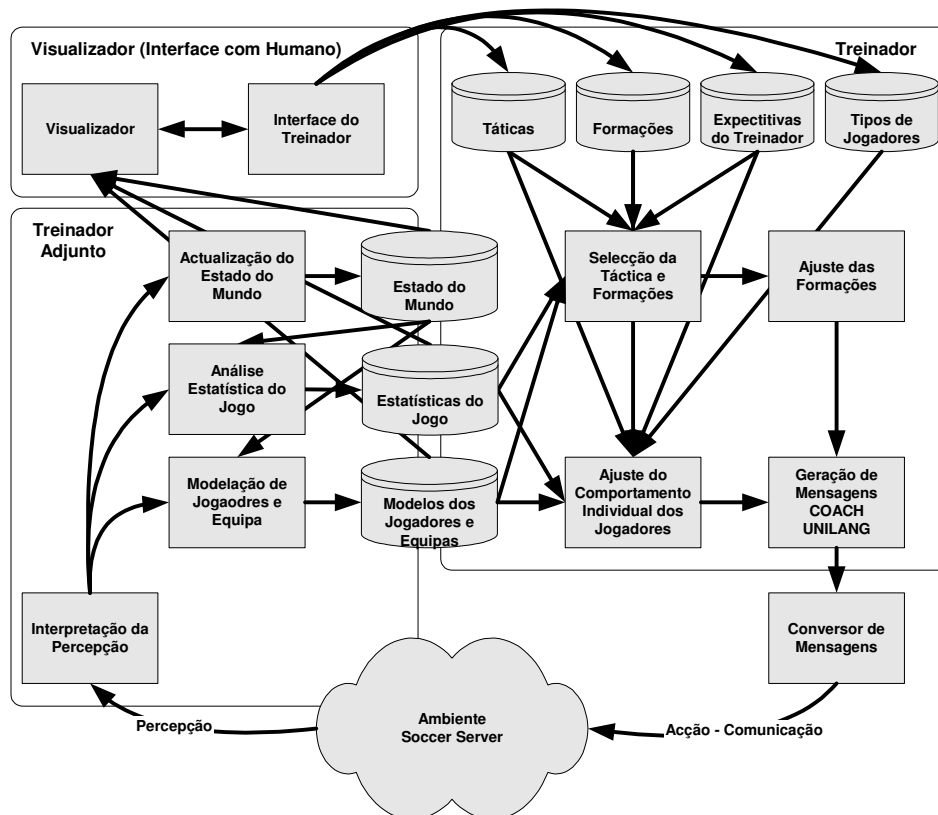


Figura 81: Arquitectura do Treinador da equipa FC Portugal

O Treinador possui uma interface gráfica que permite a um utilizador humano instanciar as suas estruturas de conhecimento e visualizar o seu raciocínio e informação de que dispõe no decurso de cada jogo. As estruturas de conhecimento do treinador são semelhantes às dos agentes jogadores: tática, formações e papéis (tipos de jogadores). O treinador não possui informação sobre estratégias de comunicação ou visão, sendo estas deixadas ao critério dos jogadores. Possui no entanto informação relativa às expectativas para o jogo contendo o resultado final esperado (golos marcados e sofridos) para o jogo e informações diversas sobre o comportamento esperado do oponente.

A informação que o treinador utiliza para o seu processo de decisão é fornecida por um módulo autónomo designado por treinador assistente. Este módulo interpreta a informação de percepção do treinador e as comunicações efectuados pelos jogadores proveniente do simulador. Utilizando esta informação mantém actualizado um estado do mundo semelhante ao dos agentes jogadores e a informação relativa ao comportamento do adversário e estatísticas de jogo. Esta informação é “enviada” (partilhada pois ambos os agentes são implementados como um único agente) ao agente treinador que a utiliza no seu processo de decisão. A informação é também disponibilizada graficamente utilizando a interface com o utilizador.



Utilizando as suas estruturas de conhecimento e a informação que lhe é “enviada” pelo treinador assistente, o treinador principal decide qual a tática a utilizar em cada instante de jogo e realiza ajustes às formações e tipos de jogadores utilizados pela tática e aos parâmetros globais da mesma.

O treinador inclui ainda um gerador de mensagens utilizando a sintaxe da linguagem COACH UNILANG [Reis e Lau, 2002] e um conversor de mensagens que, caso seja activado, permite a simplificação destas mensagens e conversão para outras linguagens de treinador mais simples como a linguagem CLang.

### 9.11.2 Utilização das Instruções do Treinador nos Jogadores

Os jogadores da equipa FC Portugal podem usar as instruções do treinador de três modos distintos:

- **Instruções Interpretadas como Ordens:** As instruções provenientes do treinador são interpretadas como ordens absolutas e substituem as estruturas locais de conhecimento dos jogadores;
- **Instruções Interpretadas como Conselhos:** As instruções provenientes do treinador são fundidas com a base de conhecimento local dos agentes sendo-lhes atribuído um determinado grau de confiança;
- **Instruções Ignoradas:** As instruções provenientes do treinador são ignoradas .

A implementação prática é realizada associando ao treinador um determinado factor de confiança no início do jogo. Quando a equipa FC Portugal é treinada pelo seu próprio treinador, os jogadores são instanciados com uma confiança absoluta de 100% no treinador e como tal as instruções deste serão interpretadas como ordens. Por exemplo, se o treinador solicitar a alteração para uma outra tática pré-definida, todos os jogadores seguirão de imediato as suas instruções e passarão a utilizar os parâmetros táticos, formações e tipos de jogadores definidos nessa tática [Reis e Lau, 2003b].

No caso do factor de confiança no treinador ser nulo, os jogadores ignoram qualquer ordem fornecida pelo treinador. No caso de um qualquer outro factor de confiança, as instruções do treinador relativas ao comportamento colectivo (táticas e formações) são aceites desde que a confiança no treinador exceda um factor mínimo pré-definido. As instruções sobre o comportamento individual são interpretadas como conselhos e adicionados à base de conhecimento com uma confiança idêntica à confiança atribuída ao treinador.

## 9.12 Resultados em Competições RoboCup

Nesta secção apresentam-se os resultados obtidos pela equipa FC Portugal nos dois campeonatos da Europa e nos dois campeonatos do mundo em que participou utilizando as metodologias de coordenação descritas nesta dissertação. Destes resultados destaca-se que a FC Portugal venceu os dois campeonatos da Europa e o campeonato do mundo 2000 e classificou-se em terceiro lugar no campeonato do mundo em 2001 (embora obtendo um *goal-average* de 150-5 neste campeonato). Em todas as competições o FC Portugal obteve o melhor *goal-average*, marcando no total das 4 competições 387 golos, sofrendo unicamente 9<sup>87</sup>.

### 9.12.1 Avaliação dos Resultados Obtidos

O resultado final de um jogo de futebol robótico, embora seja uma medida significativa, não pode ser utilizado como única medida do sucesso de uma determinada equipa ou metodologia específica. Embora o resultado (em termos de golos marcados e sofridos) seja muito importante, nomeadamente em competições oficiais, diversas outras medidas são também importantes, incluindo a posse de bola por região do campo, o número de remates à baliza e as oportunidades de golo. As medidas da qualidade do jogo da equipa utilizadas na avaliação realizada no âmbito desta dissertação foram:

- **Resultado – Golos Marcados (GM-GS):** Golos marcados e sofridos por cada uma das equipas em cada jogo;
- **Remates à Baliza (RE-RS):** Remates efectuados e sofridos em cada jogo. No âmbito desta avaliação, foi considerado um remate um chute efectuado por um jogador da equipa atacante (ou em conjunto por um jogador da equipa atacante e outro jogador), na direcção aproximada da baliza (cuja trajectória intercepte um rectângulo que rodeia em um metro a linha de golo em todas as direcções) e cuja velocidade inicial permitisse atingir a baliza com um velocidade superior a 1 metro/ciclo ou que realmente atinja a baliza.
- **Oportunidades de Golo (OC-OS):** Oportunidades de golo construídas e sofridas em cada jogo. No âmbito desta avaliação foi considerada uma oportunidade de golo um remate ou uma bola que embora não tenha sido rematada em direcção à baliza chegou a ser controlada (estar na área de chute) por um avançado numa zona perigosa. Por zona perigosa foi considerado o rectângulo com largura 30 m

---

<sup>87</sup> Em 2002 a FC Portugal não participou no campeonato europeu e classificou-se em quinto lugar no campeonato do mundo. Neste mesmo campeonato classificou-se em terceiro lugar na competições de visualizadores e venceu a competição de treinadores. O trabalho de investigação que conduziu à elaboração da equipa FC Portugal 2002 e da sua continuação FC Portugal 2003 não é descrito nesta dissertação.

e comprimento 18.5m (até ao limite da área) representado na figura 82. Foram também consideradas como oportunidades de golo bolas enviadas em direcção a um jogador que se encontrasse numa zona muito perigosa mas que tenham sido interceptadas com dificuldade por adversários. O conceito de região muito perigosa corresponde ao rectângulo com largura de 20m e comprimento 15m (ver figura 82). Uma interceptação difícil foi considerada aquela em que se o adversário que interceptou a bola, após ter começado o seu movimento de interceptação, tivesse falhado um comando não teria conseguido interceptar a bola.

- **Posse de Bola (Ataque, Meio-Campo ou Defesa):** Localização da bola (em percentagem) no decurso de cada jogo. Para estimar a posse de bola, o campo foi dividido em três rectângulos com igual área, considerando-se o tempo (em ciclos) que a bola permaneceu em cada um destes rectângulos.

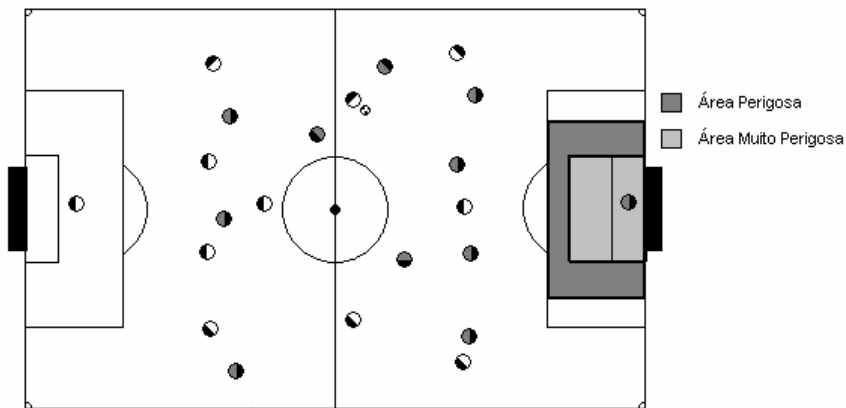


Figura 82: Regiões perigosa e muito perigosa consideradas como conducentes a oportunidades de golo

De forma a efectuar a avaliação, as medidas acima enumeradas foram incorporadas no analisador de jogo contido na ferramenta *Team Designer* e estimadas de forma automática.

### 9.12.2 Campeonato Europeu – Amesterdão 2000

O primeiro campeonato Europeu de RoboCup disputou-se em Amesterdão na Holanda de 29 de Maio a 2 de Junho de 2000. Esta foi a primeira competição oficial em que a equipa FC Portugal participou. Embora tenham sido já utilizados pela equipa versões preliminares do SBSP e DPRE, a equipa ainda não possuía a nível dos *low-level skills* o *Optimization Kick* e *Smart Dribble* e não possuía as metodologias de coordenação por comunicação avançada, percepção inteligente e controlo parcialmente hierárquico. Os resultados obtidos pela equipa no campeonato Europeu de 2000 podem ser analisados na tabela 26.

<i>Adversário</i>	<i>Result.</i>	<i>Remat.</i>	<i>Oport.</i>	<i>Ataque</i>	<i>Meio</i>	<i>Defesa</i>
<b>Primeira Fase – Grupo A</b>						
<i>Essex Wizards (Inglaterra)</i>	3-0	8-0	14-1	57%	32%	11%
<i>Lucky Lubeck (Alemanha)</i>	13-0	19-0	28-1	56%	37%	7%
<i>Cyberoos (Austrália)</i>	4-0	6-1	12-1	54%	43%	3%
<i>Pizza Tower (Itália)</i>	22-0	25-0	30-0	51%	49%	0%
<i>Polytech (Rússia)</i>	19-0	25-0	30-0	54%	32%	14%
<i>PSI (Rússia)</i>	6-0	21-0	36-0	78%	21%	1%
<b>Fase Final</b>						
<i>Wroclaw (Polónia)</i>	13-0	16-0	19-0	53%	40%	7%
<i>Essex Wizards (Inglaterra)</i>	5-0	8-2	11-8	33%	35%	32%
<b>Final</b>						
<i>Karlsruhe Brainstormers (Alemanha)</i>	2-0	8-4	14-5	37%	30%	33%
<i>Total</i>	87-0	136-7	194-16	53%	35%	12%

Tabela 26: Resultados obtidos pela FC Portugal no campeonato Europeu – Amsterdão 2000

Na tabela 26 é visível que a equipa teve um grande domínio em todos os jogos com maior posse de bola no ataque, mais oportunidades de golo, mais remates e mais golos marcados do que qualquer adversário. O carácter um pouco aleatório dos jogos de futebol robótico simulado pode ser verificado comparando os dois jogos realizados entre a equipa FC Portugal e os Essex Wizards. No primeiro jogo a equipa FC Portugal teve um domínio muito superior com mais oportunidades de golo (14 contra 1 do adversário enquanto no segundo jogo as oportunidades foram 11 contra 8) e muito mais ataque (57% contra 11% de ataque no primeiro jogo contra unicamente 33% contra 32% de ataque no segundo jogo). No entanto o resultado final do primeiro jogo foi 3-0 contra 5-0 no segundo jogo. Na final, a capacidade colectiva da equipa FC Portugal superiorizou-se aos excelentes *low-level skills* (chuto, movimentação, drible e interceptação) dos Karlsruhe Brainstormers.

### 9.12.3 RoboCup 2000 - Melbourne

O campeonato do mundo de futebol robótico – RoboCup 2000 foi disputado em Melbourne de 26 de Agosto a 3 de Setembro de 2000. Destaca-se que a equipa FC Portugal foi a única equipa portuguesa a participar neste campeonato<sup>88</sup>. Neste campeonato a equipa FC Portugal utilizou já os mecanismos de visão estratégica, comunicação avançada, *Optimization Kick* e as versões descritas do mecanismo de posicionamento

<sup>88</sup> Embora diversas equipas portuguesas estivessem pré-qualificadas nas modalidades de robôs pequenos e médios.

estratégico - SBSP e troca dinâmica de posições – DPRE. Os resultados obtidos demonstram uma enorme superioridade em todas as estatísticas relativamente aos adversários.

<i>Adversário</i>	<i>Resul.</i>	<i>Remat.</i>	<i>Oport.</i>	<i>Ataque</i>	<i>Meio</i>	<i>Defesa</i>
<b><i>Primeira Fase – Grupo D</i></b>						
<i>Oulu 2000 (Finlândia)</i>	<i>33-0</i>	<i>34-0</i>	<i>37-0</i>	<i>51%</i>	<i>49%</i>	<i>0%</i>
<i>Zeng 2000 (Japão)</i>	<i>18-0</i>	<i>26-0</i>	<i>33-0</i>	<i>65%</i>	<i>33%</i>	<i>2%</i>
<i>Robolog 2000 (Alemanha)</i>	<i>20-0</i>	<i>26-0</i>	<i>30-0</i>	<i>61%</i>	<i>36%</i>	<i>3%</i>
<b><i>Fase Final</i></b>						
<i>Essex Wizards (Inglaterra)</i>	<i>7-0</i>	<i>15-0</i>	<i>24-0</i>	<i>70%</i>	<i>22%</i>	<i>8%</i>
<i>Karlsruhe Brainstormers (Alemanha)</i>	<i>3-0</i>	<i>6-2</i>	<i>15-3</i>	<i>48%</i>	<i>31%</i>	<i>21%</i>
<i>YowAI2000 (Japão)</i>	<i>6-0</i>	<i>19-0</i>	<i>33-0</i>	<i>76%</i>	<i>22%</i>	<i>2%</i>
<i>ATT-CMUnited 2000 (EUA)</i>	<i>6-0</i>	<i>12-1</i>	<i>18-2</i>	<i>53%</i>	<i>36%</i>	<i>11%</i>
<b><i>Final</i></b>						
<i>Karlsruhe Brainstormers (Alemanha)</i>	<i>1-0</i>	<i>6-0</i>	<i>11-2</i>	<i>46%</i>	<i>38%</i>	<i>16%</i>
<i>Total</i>	<i>94-0</i>	<i>144-3</i>	<i>201-7</i>	<i>59%</i>	<i>33%</i>	<i>8%</i>

*Tabela 27: Resultados obtidos pela FC Portugal no campeonato Mundial – Melbourne 2000*

Dos resultados obtidos (tabela 27) realça-se a vitória por 6-0 (12 remates contra 1 e 18 oportunidades contra 2) sobre os anteriores campeões do mundo (ATT-CMUnited) que até esse jogo e nos últimos 3 campeonatos do mundo não tinham sofrido qualquer golo. Destaca-se também que embora o resultado da final tenha sido unicamente 1-0, a vitória foi obtida com grande facilidade (6-0 em remates, 11-2 em oportunidades e 46% contra 16% de ataque). Destaca-se ainda que o guarda-redes da equipa só teve de defender três remates nos oito jogos do torneio.

### **9.12.4 German Open 2001 – Paderborn**

O primeiro German Open foi disputado em Paderborn entre 7 e 10 de Junho de 2001. Este campeonato, desde 2001 funcionou como campeonato europeu, reunindo não só as melhores equipas europeias das várias modalidades RoboCup como também algumas equipas fortes de países não europeus (como por exemplo o Irão).

<i>Adversário</i>	<i>Resul.</i>	<i>Remat.</i>	<i>Oport.</i>	<i>Ataque</i>	<i>Meio</i>	<i>Defesa</i>
<b>Primeira Fase – Grupo A</b>						
<i>Sharif Arvand (Irão)</i>	2-0	3-1	6-5	24%	47%	29%
<i>Aras (Irão)</i>	3-1	3-1	11-2	36%	57%	8%
<i>Osna BallByters (Alemanha)</i>	26-0	32-0	41-0	70%	29%	1%
<i>UvaTrilearn (Holanda)</i>	5-2	6-6	10-7	29%	49%	22%
<i>Lucky Lubeck (Alemanha)</i>	3-1	8-4	11-4	35%	49%	16%
<b>Fase Final</b>						
<i>Dr. Web (Rússia)</i>	6-0	7-1	16-3	51%	41%	9%
<i>Robolog (Alemanha)</i>	9-0	13-2	13-2	28%	60%	12%
<i>Karlsruhe Brainstormers (Alemanha)</i>	1-0*	4-1	10-1	21%	30%	49%
<b>Final</b>				<i>* após prolongamento</i>		
<i>Karlsruhe Brainstormers (Alemanha)</i>	1-0	3-1	10-1	39%	32%	29%
<i>Total</i>	56-4	79-17	128-25	37%	44%	19%

*Tabela 28: Resultados obtidos pela FC Portugal no German Open 2001*

A equipa FC Portugal utilizada neste torneio foi a equipa FC Portugal 2000 cuja configuração se mostrou desadequada às novas regras do RoboCup em 2001, nomeadamente à maior velocidade da bola resultante da maior potência de chute e jogadores com mais energia<sup>89</sup>. Acresce a este facto que a equipa havia disponibilizado em Outubro de 2000 uma parte do código do FC Portugal, contendo uma versão simplificado do algoritmo de posicionamento estratégico e diversos artigos descrevendo as principais metodologias da equipa. Exactamente metade das equipas no German Open 2001 utilizou este código e conseqüente método de posicionamento e basearam a filosofia das suas equipas na do FC Portugal 2000.

Desta desadequação da equipa às novas regras resultou que, na primeira fase, diversos dos seus jogos foram bastante equilibrados com destaque para os jogos com as equipas Sharif Arvand e UvaTrilearn [de Boer et al., 2001].

Para os jogos da fase final a configuração da equipa foi ligeiramente alterada (aumentando a tendência dos jogadores para se deslocarem estrategicamente no campo e para tentarem intercepções menos seguras). Para a final, a configuração foi novamente alterada, aumentando a rapidez e segurança do jogo. O resultado destas alterações tácticas foi uma melhoria clara da qualidade do jogo que se reflecte nas estatísticas de jogo apresentadas na tabela 28. É visível que a equipa FC Portugal, embora jogando contra as melhores equipas da competição, atacou bastante mais na fase final do torneio e criou muito mais oportunidades de golo do que os seus adversários. Do primeiro jogo contra os

<sup>89</sup> Alterações introduzidas pelo comité técnico do RoboCup

Brainstormers para a final é também visível que a maior rapidez de jogo resultou numa passagem da defesa para o ataque mais rápida, fazendo com a equipa FC Portugal atacasse bastante mais neste jogo e o vencesse com mais facilidade.

### 9.12.5 RoboCup 2001 – Seattle

O campeonato do mundo de futebol robótico disputou-se em Seattle nos Estados Unidos de 2 a 10 de Agosto de 2001. Para este campeonato a principal alteração no comportamento da equipa FC Portugal foi a introdução da coordenação por controlo parcialmente hierárquico e diversas melhorias nos restantes mecanismos de coordenação. Os resultados obtidos pela equipa são apresentados na tabela 29.

<i>Adversário</i>	<i>Resul.</i>	<i>Remat.</i>	<i>Oport.</i>	<i>Ataque</i>	<i>Meio</i>	<i>Defesa</i>
<b>Primeira Fase – Grupo E</b>						
<i>11Monkeys (Japão)</i>	29-0	+29-0	+29-0	<i>*Logfile indisponível</i>		
<i>TUT-Grove (Japão)</i>	9-0	+9-0	+9-0	<i>*Logfile indisponível</i>		
<i>RMIT Goannas (Austrália)</i>	32-0	+32-0	+32-0	<i>*Logfile indisponível</i>		
<i>Robolog2001 (Alemanha)</i>	8-0	+8-0	+8-0	<i>*Logfile indisponível</i>		
<b>Segunda Fase – Grupo C</b>						
<i>Helli Respina (Irão)</i>	5-0	6-1	15-1	54%	41%	5%
<i>UtUtd (Irão)</i>	16-0	29-0	42-0	79%	20%	1%
<i>FC Tripleta (Japão)</i>	4-0	12-1	22-1	67%	25%	8%
<i>AT Humboldt (Alemanha)</i>	13-0	22-0	32-0	83%	17%	0%
<i>ATTUnited2001 (EUA)</i>	22-0	24-0	34-0	74%	25%	1%
<b>Fase Final</b>						
<i>YowAI2001 (Japão)</i>	8-0	14-0	23-1	60%	28%	12%
<i>UvaTrilearn (Holanda)</i>	4-1	6-2	12-6	42%	40%	18%
<i>Tsinghuaeolus (China)</i>	0-3	3-3	9-3	37%	54%	10%
<i>Karlsruhe Brainstormers (Alem.)</i>	0-1	1-1	13-2	46%	32%	22%
<i>Total</i>	150-5	195*-8	281*-14	60%	32%	8%

*Tabela 29: Resultados obtidos pela FC Portugal no campeonato Mundial – Seattle 2001*

Analisando os resultados apresentados na tabela 29 conclui-se facilmente que, tal como já tinha sido referido anteriormente, os resultados da competição nem sempre são justos e que o resultado final de um jogo não é a única medida a utilizar para retirar conclusões válidas deste tipo de experiências científicas. Na tabela 29 é visível que a equipa FC Portugal dominou claramente o torneio marcando 150 golos e sofrendo unicamente 5, tendo em todos os jogos um número igual ou superior de remates que os seus adversários, o dobro ou mais de oportunidades de golo e mais do dobro de ataque que os seus oponentes. No entanto, a equipa acabou por perder dois jogos em que teve claro domínio

de jogo e muito mais oportunidades de golo que os seus adversários, sendo eliminada do torneio e classificando-se unicamente em terceiro lugar. Verifica-se ainda que mesmo sem contabilizar os jogos disputados na primeira fase do torneio<sup>90</sup> em que o domínio da equipa FC Portugal foi total, este foi o campeonato em que a equipa mais oportunidades criou (mais de 281), mais rematou à baliza (mais de 195 remates), mais golos marcou (150), mais atacou (60%) e menos defendeu (8%).

Destaca-se que num dos jogos decisivos, os dois últimos golos da equipa Tsinghuaeolus contra o FC Portugal foram marcados no final do jogo quando o treinador da equipa tinha instruído a que a mesma jogasse sem defesas e sem guarda-redes numa tentativa desesperada de inverter o resultado. Esta tática embora não tenha resultado neste jogo, foi de grande eficácia no jogo contra a equipa UvaTrilearn. Neste jogo e após 2/3 do jogo a equipa holandesa estava a vencer por 1-0, altura em que o treinador da FC Portugal instruiu o guarda-redes para ir também para o ataque. Com 11 jogadores no ataque a equipa FC Portugal dominou a partir daí claramente o jogo acabando por o vencer por 4-1.

## 9.13 Resultados em Experiências Controladas

Os resultados das competições oficiais RoboCup são bastante significativos na medida em que demonstram inequivocamente a qualidade global da equipa FC Portugal quando comparada com as equipas desenvolvidas pelas outras Universidades/Empresas participantes. Contudo, de forma a analisar os resultados individuais das diversas metodologias de coordenação desenvolvidas, torna-se necessário realizar experiências controladas que são impossíveis de realizar nas competições oficiais. As competições oficiais não são disputadas em ambientes controlados e cada equipa participante disputa no máximo um ou dois jogos contra o FC Portugal o que não permite realizar testes estatisticamente significativos. Desta forma foi realizado um extenso estudo no laboratório de forma a analisar separadamente a influência de cada metodologia desenvolvida na equipa FC Portugal.

### 9.13.1 Definição das Experiências

Nas experiências realizadas no âmbito desta tese<sup>91</sup>, de forma a testar as metodologias de coordenação propostas, foram utilizadas as seguintes equipas<sup>92</sup>:

---

<sup>90</sup> Devido à indisponibilidade dos respectivos logfiles que não foram gravados correctamente pela organização do torneio

<sup>91</sup> As experiências foram realizadas no período compreendido entre Abril e Junho de 2002.

<sup>92</sup> Os binários utilizados correspondem aos binários disponíveis na Internet em Abril de 2002.



- TSI – Tsinghuaeolus – Tsinghua University - China
- KB – Karlsruhe Brainstormers 2001 – University of Karlsruhe - Alemanha
- UVA - UVA Trilearn 2001 – University of Amsterdam - Holanda
- Yow – YowAI 2001 – Electro Communications University -Japão
- FCM - FC Portugal Melbourne – Universidade do Porto/Aveiro - Portugal
- FA - FCP Agent – Universidade do Porto/Aveiro - Portugal
- ATT – ATTCMUnited 2000 – AT&T e Carnegie Mellon University, E.U.A.
- CMU – CMUnited 99 - Carnegie Mellon University, E.U.A.

A avaliação dos resultados foi efectuada tendo como base as medidas descritas na secção 9.12.1: golos marcados/sofridos, remates, oportunidades e posse de bola por região do campo. Os testes foram realizados utilizando o servidor *soccerserver 7.10* e as configurações por defeito de todas as equipas.

A tabela 30 sumaria os resultados obtidos pela equipa FC Portugal (utilizando a sua configuração por defeito) contra as diversas equipas teste utilizadas.

<i>Equipas</i>	<i>TSI</i>	<i>KB</i>	<i>UVA</i>	<i>YOW</i>	<i>FCM</i>	<i>FA</i>	<i>ATT</i>	<i>CMU</i>
<b>Golos (M-S)</b>	<b>8-4</b>	<b>16-0</b>	<b>35-8</b>	<b>92-0</b>	<b>78-2</b>	<b>165-0</b>	<b>222-0</b>	<b>245-0</b>
<b>Vít–Emp–Der</b>	<b>5-2-3</b>	<b>8-2-0</b>	<b>8-2-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>20-8</b>	<b>28-5</b>	<b>46-18</b>	<b>136-2</b>	<b>114-7</b>	<b>235-1</b>	<b>278-0</b>	<b>296-0</b>
<b>Oportunidades</b>	<b>76-25</b>	<b>95-10</b>	<b>123-52</b>	<b>194-8</b>	<b>148-19</b>	<b>304-3</b>	<b>393-1</b>	<b>404-0</b>
<b>Ataq–Meio-Def.</b>	<b>33-51-16</b>	<b>47-36-17</b>	<b>40-39-21</b>	<b>58-31-11</b>	<b>54-34-12</b>	<b>70-27-3</b>	<b>78-22-1</b>	<b>76-23-1</b>

*Tabela 30: Resultados das Experiências Controladas Simples.*

Analisando a tabela 30 é visível que a equipa FC Portugal vence com facilidade todos os adversários, exceptuando a equipa Tsinghuaeolus, com a qual consegue 5 vitórias contra 3 derrotas, nas 10 experiências efectuadas. Unicamente três equipas conseguem marcar golos à equipa FC Portugal: Tsinghuaeolus, UVA Trilearn e FC Portugal Melbourne. Um resultado interessante é que a equipa Brainstormers, não conseguiu marcar qualquer golo à FC Portugal nas experiências realizadas (tendo perdido 8 jogos e empatado a 0-0 dois jogos). Este resultado mostra que os resultados da competição, tal como foi referido anteriormente, nem sempre são significativos (no RoboCup 2001, a equipa Brainstormers, eliminou o FC Portugal, vencendo por 1-0).

### 9.13.2 Coordenação Estratégica

De forma a averiguar a adequação da coordenação estratégica foram realizadas diversas experiências variando a forma de utilização de táticas, formações, situações, tipos de jogadores e troca dinâmica de posicionamentos.

#### 9.13.2.1 Definição de Táticas, Formações e Papéis

Na primeira experiência foram utilizados três estratégias distintas:

- **STSimple** – Estratégia simples com uma única tática, uma única formação (433) e um único tipo de jogador (com pesos iguais para as diversas acções com bola ou sem bola)
- **STTunned** – Estratégia simples com uma única tática e a formação (433) utilizada por defeito pela equipa FC Portugal com uma única situação. Esta formação foi treinada extensivamente para jogar contra a maior parte das equipas e utiliza diversos tipos de jogadores distintos.
- **STChange** – Estratégia mais elaborada com três táticas (uma tática defensiva, uma normal e uma ofensiva), diversos tipos de jogadores utilizados nas táticas e três formações distintas. Nesta estratégia foi utilizada uma metodologia muito simples para alternar a tática activa: Uma tática normal é utilizada no início do jogo. Caso a equipa esteja a vencer o jogo por menos de três golos de diferença ou a perder por mais de dois, a tática defensiva é utilizada. Caso a equipa esteja a perder por menos de três golos de diferença ou a vencer por mais de dois, a tática ofensiva é utilizada.

Os resultados obtidos nesta experiência encontram-se representados na tabela 31.

<i>Equipas</i>	<i>TSI</i>	<i>KB</i>	<i>UVA</i>	<i>YOW</i>	<i>FCM</i>	<i>FA</i>	<i>ATT</i>	<i>CMU</i>
<i>Resultados</i>								
<b>Estratégia <i>STSimple</i></b>								
<b>Golos (M-S)</b>	<b>4-6</b>	<b>6-2</b>	<b>18-12</b>	<b>76-0</b>	<b>68-2</b>	<b>113-0</b>	<b>196-0</b>	<b>204-0</b>
<b>Vít-Emp-Der</b>	<b>2-4-4</b>	<b>4-5-1</b>	<b>5-3-2</b>	<b>10-0-0</b>	<b>9-1-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>12-10</b>	<b>14-8</b>	<b>28-18</b>	<b>104-5</b>	<b>96-12</b>	<b>152-3</b>	<b>232-0</b>	<b>246-0</b>
<b>Ataq-Meio-Def.</b>	<b>25-48-27</b>	<b>34-40-26</b>	<b>30-41-29</b>	<b>50-36-14</b>	<b>42-37-21</b>	<b>59-33-8</b>	<b>68-27-5</b>	<b>72-25-3</b>
<b>Estratégia <i>STTunned</i></b>								
<b>Golos (M-S)</b>	<b>9-6</b>	<b>12-1</b>	<b>30-7</b>	<b>91-0</b>	<b>81-1</b>	<b>145-0</b>	<b>209-0</b>	<b>236-0</b>
<b>Vít-Emp-Der</b>	<b>4-3-3</b>	<b>7-3-0</b>	<b>8-1-1</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>23-12</b>	<b>22-6</b>	<b>42-18</b>	<b>132-3</b>	<b>108-10</b>	<b>196-1</b>	<b>264-0</b>	<b>256-0</b>
<b>Ataq-Meio-Def.</b>	<b>31-51-18</b>	<b>48-33-19</b>	<b>38-37-25</b>	<b>60-30-10</b>	<b>53-34-13</b>	<b>68-29-3</b>	<b>74-25-1</b>	<b>77-22-1</b>
<b>Estratégia <i>STChange</i></b>								
<b>Golos (M-S)</b>	<b>8-4</b>	<b>15-0</b>	<b>34-10</b>	<b>89-0</b>	<b>75-3</b>	<b>159-0</b>	<b>220-0</b>	<b>244-0</b>
<b>Vít-Emp-Der</b>	<b>5-2-3</b>	<b>8-2-0</b>	<b>8-2-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>18-7</b>	<b>32-3</b>	<b>47-22</b>	<b>138-1</b>	<b>102-6</b>	<b>221-1</b>	<b>268-0</b>	<b>281-0</b>
<b>Ataq-Meio-Def.</b>	<b>28-58-14</b>	<b>45-40-15</b>	<b>35-48-17</b>	<b>58-31-11</b>	<b>50-36-14</b>	<b>69-28-3</b>	<b>75-23-2</b>	<b>74-25-1</b>

*Tabela 31: Resultados das Experiências com Variação de Táticas*

Na tabela 31 é visível que a estratégia *STTunned* permite obter melhores resultados contra todas as equipas do que a estratégia *STSimple*. Este facto deve-se essencialmente a que embora ambas as estratégias utilizem a mesma formação, os tipos de jogadores utilizados pela estratégia *STTunned* são mais adequados à realização da tarefa cooperativa: avançados que rematam mais à baliza, extremos que correm à linha e cruzam a bola, médios mais agressivos, etc.

Uma conclusão interessante e um pouco surpreendente da análise dos resultados obtidos, consiste em que a estratégia *STTunned* permite obter melhores resultados contra algumas equipas do que a estratégia *STChanged*. Nos jogos contra as melhores equipas, nomeadamente no que diz respeito ao número de vitórias, empates e derrotas, a estratégia *STChange* permite claramente obter melhores resultados. Como nesta estratégia a equipa procura defender no caso de estar a ganhar o jogo e atacar no caso de estar a perder, verifica-se que contra a equipa *Tsinghuaeolus* esta estratégia diminui o número de golos marcados, remates e os tempos de ataque de ambas as equipas.

Uma conclusão importante a retirar é que as alterações táticas, pelo menos se forem executadas baseadas unicamente no resultado, não produzem um efeito muito visível no comportamento da equipa, sobretudo contra equipas claramente mais fracas. São no entanto muito úteis contra equipas de um nível semelhante, permitindo melhorar o resultado global da equipa.

### 9.13.2.2 Coordenação por Posicionamento Estratégico

De forma a testar a influência do posicionamento estratégico no comportamento global da equipa foi utilizada uma equipa com uma única tática (*STTunned*) e cinco estratégias de posicionamento distintas:

- **PACTS (Posicionamento Activo Simples)** – Inexistência do conceito de posicionamento estratégico e da distinção entre situação crítica (activa) e situação estratégica. Neste tipo de estratégia, os agentes assumem sempre um comportamento activo sem bola: interceptação, interceptação passiva, aproximação da bola, marcação de linha de passe ou marcação de adversário.
- **PACTF (Posicionamento Activo com Formação Estática)** – Semelhante ao anterior mas todos os jogadores possuem uma posição por defeito estática que permite definir uma formação. No caso de nenhuma acção sem bola ter um valor suficientemente elevado os jogadores regressam para esta posição por defeito.
- **SPAR (Posicionamento por Atracções e Repulsões)** – Posicionamento baseado no algoritmo de Stone et al. [Stone et al., 2000a]. Jogadores são atraídos pela bola, repelidos pelos colegas de equipa e atraídos pelos adversários se estes tiverem a posse de bola e repelidos caso contrário.
- **SP (Posicionamento Estratégico Simples)** – Utilizando unicamente uma situação.
- **SBSP (Posicionamento Estratégico baseado em Situações Elaborado)** - Utilizando situações para ataque, defesa, reposições de bola pelos guarda-redes, pontapés de baliza, livres, ocasiões de golo, cantos e lançamentos.

A tabela 32 sumaria os resultados obtidos nas experiências realizadas.

<i>Equipas</i>	<i>TSI</i>	<i>KB</i>	<i>UVA</i>	<i>YOW</i>	<i>FCM</i>	<i>FA</i>	<i>ATT</i>	<i>CMU</i>
<i>Resultados</i>								
<b>PACTS (Posicionamento Activo Simples)</b>								
<b>Golos (M-S)</b>	<b>0-65</b>	<b>0-22</b>	<b>1-38</b>	<b>6-19</b>	<b>0-34</b>	<b>2-2</b>	<b>14-6</b>	<b>21-0</b>
<b>Vít-Emp-Der</b>	<b>0-0-10</b>	<b>0-2-8</b>	<b>0-0-10</b>	<b>1-3-6</b>	<b>0-0-10</b>	<b>2-6-2</b>	<b>8-1-1</b>	<b>10-0-0</b>
<b>Remates</b>	<b>1-86</b>	<b>2-36</b>	<b>2-72</b>	<b>13-37</b>	<b>2-61</b>	<b>5-16</b>	<b>26-14</b>	<b>37-16</b>
<b>Ataq-Meio-Def.</b>	<b>8-58-34</b>	<b>9-61-30</b>	<b>14-44-42</b>	<b>16-54-30</b>	<b>8-47-45</b>	<b>14-58-28</b>	<b>38-46-16</b>	<b>42-46-12</b>
<b>PACTF (Posicionamento Activo com Formação Estática)</b>								
<b>Golos (M-S)</b>	<b>0-48</b>	<b>1-18</b>	<b>2-46</b>	<b>10-16</b>	<b>4-24</b>	<b>4-3</b>	<b>24-4</b>	<b>32-0</b>
<b>Vít-Emp-Der</b>	<b>0-0-10</b>	<b>0-3-7</b>	<b>0-0-10</b>	<b>3-4-3</b>	<b>0-0-10</b>	<b>3-5-2</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>2-75</b>	<b>3-32</b>	<b>5-89</b>	<b>21-39</b>	<b>6-48</b>	<b>6-20</b>	<b>39-12</b>	<b>54-8</b>
<b>Ataq-Meio-Def.</b>	<b>12-56-32</b>	<b>10-58-32</b>	<b>18-47-35</b>	<b>22-50-28</b>	<b>14-55-31</b>	<b>24-50-26</b>	<b>44-42-14</b>	<b>50-43-7</b>
<b>SPAR (Posicionamento por Atrações e Repulsões)</b>								
<b>Golos (M-S)</b>	<b>0-36</b>	<b>2-12</b>	<b>1-42</b>	<b>8-14</b>	<b>5-25</b>	<b>6-2</b>	<b>33-3</b>	<b>45-0</b>
<b>Vít-Emp-Der</b>	<b>0-0-10</b>	<b>1-4-5</b>	<b>0-0-10</b>	<b>2-4-4</b>	<b>0-0-10</b>	<b>4-5-1</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>2-55</b>	<b>8-26</b>	<b>12-76</b>	<b>16-42</b>	<b>6-46</b>	<b>8-12</b>	<b>52-10</b>	<b>71-5</b>
<b>Ataq-Meio-Def.</b>	<b>10-54-36</b>	<b>13-52-35</b>	<b>19-41-40</b>	<b>25-45-30</b>	<b>16-41-43</b>	<b>29-45-26</b>	<b>50-37-13</b>	<b>58-34-8</b>
<b>SP (Posicionamento Estratégico Simples)</b>								
<b>Golos (M-S)</b>	<b>9-6</b>	<b>12-1</b>	<b>30-7</b>	<b>91-0</b>	<b>81-1</b>	<b>145-0</b>	<b>209-0</b>	<b>236-0</b>
<b>Vít-Emp-Der</b>	<b>4-3-3</b>	<b>7-3-0</b>	<b>8-1-1</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>23-12</b>	<b>22-6</b>	<b>42-18</b>	<b>132-3</b>	<b>108-10</b>	<b>196-1</b>	<b>264-0</b>	<b>256-0</b>
<b>Ataq-Meio-Def.</b>	<b>31-51-18</b>	<b>48-33-19</b>	<b>38-37-25</b>	<b>60-30-10</b>	<b>53-34-13</b>	<b>68-29-3</b>	<b>74-25-1</b>	<b>77-22-1</b>
<b>SBSP(Posicionamento Estratégico baseado em Situações)</b>								
<b>Golos (M-S)</b>	<b>10-3</b>	<b>19-0</b>	<b>32-6</b>	<b>98-0</b>	<b>76-1</b>	<b>158-0</b>	<b>204-0</b>	<b>246-0</b>
<b>Vít-Emp-Der</b>	<b>6-2-2</b>	<b>7-3-0</b>	<b>9-1-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>	<b>10-0-0</b>
<b>Remates</b>	<b>16-4</b>	<b>35-2</b>	<b>45-16</b>	<b>146-0</b>	<b>102-3</b>	<b>209-0</b>	<b>253-0</b>	<b>273-0</b>
<b>Ataq-Meio-Def.</b>	<b>34-50-16</b>	<b>40-41-19</b>	<b>40-35-25</b>	<b>63-25-12</b>	<b>55-32-13</b>	<b>71-27-1</b>	<b>72-26-2</b>	<b>75-24-1</b>

*Tabela 32: Resultados das Experiências com Variação de Esquema de Posicionamento*

Analisando a tabela 32 é visível que os resultados obtidos através da utilização do SBSP são claramente melhores do que os resultados dos restantes esquemas de posicionamento. Verifica-se também que a utilização de diferentes esquemas de posicionamento por situações melhora o comportamento da equipa FC Portugal, sobretudo contra as equipas mais fortes. Verifica-se também que a utilização do SBSP com situações permite à equipa sofrer menos golos do que não utilizando situações. Analisando os jogos disputados verifica-se que este facto está relacionado com o incremento de qualidade da defesa da equipa obtido através da utilização de diferentes esquemas de posicionamento para situações defensivas.

### 9.13.2.3 Coordenação por Troca Dinâmica de Papéis e Posicionamentos

De forma a aferir da influência da troca dinâmica de papéis e posicionamentos no desempenho global da equipa foram realizadas experiências utilizando diferentes esquemas de DPRE:

- **DPRE0:** Trocas dinâmicas de posicionamento não são efectuadas;
- **DPRE1:** Trocas dinâmicas de posição pouco frequentes baseadas unicamente em distâncias às posições estratégicas;
- **DPRE2:** Trocas dinâmicas de posição frequentes baseadas unicamente em distâncias às posições estratégicas.

Não foram efectuadas experiências utilizando a adequação dos agentes aos papéis pois o FC Portugal não dispunha de agentes heterogéneos na data de realização destas experiências. Na realização das experiências foi utilizada a estratégia *STTunned* referida anteriormente. Dado que as diferenças de resultado obtidas com a utilização dos esquemas DPRE1 e DPRE2 não foram significativas, na tabela 33, são apresentados unicamente os resultados comparando a utilização do esquema de troca de posições DPRE2 com o esquema DPRE0.

<i>Equipas</i>	<i>TSI</i>	<i>KB</i>	<i>UVA</i>	<i>YOW</i>	<i>FCM</i>	<i>FA</i>	<i>ATT</i>	<i>CMU</i>
<i>Resultados</i>								
<b>DPRE 0 (Sem Trocas de Posição)</b>								
Golos (M-S)	8-10	10-3	24-12	76-0	65-6	126-0	187-0	225-0
Vít-Emp-Der	2-4-4	6-3-1	6-2-2	10-0-0	9-0-1	10-0-0	10-0-0	10-0-0
Remates	18-14	16-7	34-21	108-2	89-15	170-1	232-0	243-0
Ataq-Meio-Def.	27-50-23	40-36-22	34-39-27	55-34-11	47-33-20	62-33-5	69-28-3	78-21-1
<b>DPRE 2 (Estratégia STTunned)</b>								
Golos (M-S)	9-6	12-1	30-7	91-0	81-1	145-0	209-0	236-0
Vít-Emp-Der	4-3-3	7-3-0	8-1-1	10-0-0	10-0-0	10-0-0	10-0-0	10-0-0
Remates	23-12	22-6	42-18	132-3	108-10	196-1	264-0	256-0
Ataq-Meio-Def.	31-51-18	48-33-19	38-37-25	60-30-10	53-34-13	68-29-3	74-25-1	77-22-1

*Tabela 33: Resultados das Experiências com Variação do Esquema de Troca Dinâmica de Posicionamentos*

Analisando os resultados é visível que a troca dinâmica de posições tem grande influência no desempenho da equipa sobretudo contra as melhores equipas. Embora o número de remates dos adversários não aumente significativamente não utilizando o DPRE, verifica-se que o número de golos dos adversários sofre um acréscimo significativo. Analisando alguns dos jogos em pormenor verifica-se que o acréscimo de golos se deve a situações

defensivas em que a não execução de uma dada troca de posicionamento, deixa um “buraco” na defesa que permite aos adversários marcar um gol de forma fácil.

A utilização do DPRE permite também que uma equipa com um número mais reduzido de jogadores obtenha um bom desempenho. Foram realizadas algumas experiências simples com os seguintes cenários:

- **NODPRE-5PL** - Sem DPRE utilizando uma equipa composta por 5 jogadores (guarda –redes e quatro jogadores de campo);
- **DPRE-5PL** – Com DPRE2 utilizando uma equipa composta por 5 jogadores.

Utilizando como equipa de teste a equipa CMUnited verificou-se que a equipa FC Portugal com cinco jogadores sem DPRE venceu 3 jogos em cinco, empatando os restantes dois contra a equipa CMUnited, marcando um total de 14 golos contra 4 do adversário. No entanto, a mesma equipa utilizando DPRE, consegue vencer todos os jogos marcando um total de 36 golos contra 2 do adversário também em 5 jogos.

### 9.13.3 Coordenação por Controlo Parcialmente Hierárquico

Infelizmente é muito complicado medir quantitativamente no laboratório a influência da coordenação por controlo parcialmente hierárquico. A melhor medida do sucesso desta metodologia foi o seu resultado em competições oficiais, nomeadamente no jogo entre a FC Portugal e a equipa UVATrilearn disputado no campeonato do mundo em 2001. Neste jogo, como foi já descrito, a influência do treinador foi determinante para permitir já perto do final do jogo alterar um resultado desfavorável de 0-1 para um resultado favorável de 4-1.

As experiências controladas da Coordenação por Controlo Parcialmente Hierárquico realizadas utilizaram os seguintes cenários:

- **PH0** – Tática por defeito (*STTunned*) é utilizada;
- **PH1** – Treinador altera a tática (seleccionando outra tática aleatoriamente) quando o resultado for inferior ao esperado;
- **PH2** – Treinador selecciona a tática baseado nas estatísticas do jogo e comportamento do adversário.

Os resultados obtidos podem ser visualizados na tabela 34.

<i>Equipas</i>	<i>TSI</i>	<i>KB</i>	<i>UVA</i>	<i>YOW</i>	<i>FCM</i>	<i>FA</i>	<i>ATT</i>	<i>CMU</i>
<i>Resultados</i>								
<b>PH0 – Tática Única (STTunned)</b>								
<b>Golos (M-S)</b>	9-6	12-1	30-7	91-0	81-1	145-0	209-0	236-0
<b>Vít-Emp-Der</b>	4-3-3	7-3-0	8-1-1	10-0-0	10-0-0	10-0-0	10-0-0	10-0-0
<b>Remates</b>	23-12	22-6	42-18	132-3	108-10	196-1	264-0	256-0
<b>Ataq-Meio-Def.</b>	31-51-18	48-33-19	38-37-25	60-30-10	53-34-13	68-29-3	74-25-1	77-22-1
<b>PH1 – Alteração de Tática em Função do Resultado</b>								
<b>Golos (M-S)</b>	7-8	13-2	34-5	82-0	73-3	147-0	184-0	203-0
<b>Vít-Emp-Der</b>	3-4-3	6-4-0	8-1-1	10-0-0	10-0-0	10-0-0	10-0-0	10-0-0
<b>Remates</b>	18-13	19-7	48-17	118-3	96-16	192-6	232-0	246-0
<b>Ataq-Meio-Def.</b>	28-50-22	49-30-21	40-38-22	56-33-11	45-36-19	64-32-4	65-29-6	71-28-1
<b>PH2 – Tática em Função do Resultado e Estatísticas de Jogo</b>								
<b>Golos (M-S)</b>	8-3	11-1	36-5	99-0	89-2	139-0	203-0	228-0
<b>Vít-Emp-Der</b>	5-3-2	8-2-0	7-2-1	10-0-0	10-0-0	10-0-0	10-0-0	10-0-0
<b>Remates</b>	17-8	21-4	52-14	147-3	138-12	185-3	238-0	262-0
<b>Ataq-Meio-Def.</b>	28-56-16	47-36-17	36-36-28	54-38-8	55-33-12	66-31-3	74-25-1	77-22-1

Tabela 34: Resultados das Experiências com Coordenação Parcialmente Hierárquica

Analisando os resultados conclui-se que a influência da coordenação parcialmente hierárquica nos resultados competitivos da equipa é bastante reduzida. Aliás esta coordenação parece ter um efeito detrimental na equipa quando esta joga com as equipas mais fracas. A explicação parece estar relacionada com o facto de uma tática bem afinada ser melhor para jogar com equipa fracas do que táticas mais elaboradas mas com pormenores menos afinados.

A estratégia PH2 funciona bastante bem contra as melhores equipas, permitindo aumentar o número de vitórias mas diminuindo o número de golos marcados. Este facto prende-se com o facto de o treinador, verificando através das estatísticas de jogo que estes são jogos equilibrados, após a equipa marcar o primeiro golo, ordenar a utilização de táticas mais defensivas. Desta forma o jogo processa-se mais a meio campo e com menos remates a ambas as balizas e consequentemente com menos golos marcados.

### 9.13.4 Coordenação por Comunicação, Percepção Inteligente e Modelização Mútua

De forma a aferir do interesse das metodologias de comunicação avançada, percepção inteligente e modelização mútua foram realizadas diversas experiências analisando os resultados competitivos obtidos através da sua utilização. Em todas as experiências foi utilizada a estratégia *STTunned*:



- **ADVCOM+MM+SLM:** Todas as metodologias (ADVCOM, SLM e modelização mútua) são utilizadas;
- **NOADVCOM** – Sem a utilização de comunicação;
- **NOMM** – Sem a utilização de modelização mútua;
- **NOSLM** – Sem utilização de visão estratégica;
- **NO** - Sem Comunicação, SLM e Modelização Mútua (visão aleatória é utilizada).

Na tabela 35 são sumariados os resultados obtidos em cada estratégia:

<i>Equipas</i>	<i>TSI</i>	<i>KB</i>	<i>UVA</i>	<i>YOW</i>	<i>FCM</i>	<i>FA</i>	<i>ATT</i>	<i>CMU</i>
<i>Resultados</i>								
<b>ADVCOM + MM + SLM (STTunned)</b>								
Golos (M-S)	9-6	12-1	30-7	91-0	81-1	145-0	209-0	236-0
Vít-Emp-Der	4-3-3	7-3-0	8-1-1	10-0-0	10-0-0	10-0-0	10-0-0	10-0-0
Remates	23-12	22-6	42-18	132-3	108-10	196-1	264-0	256-0
Ataq-Meio-Def.	31-51-18	48-33-19	38-37-25	60-30-10	53-34-13	68-29-3	74-25-1	77-22-1
<b>NOADVCOM – Sem ADVCOM</b>								
Golos (M-S)	6-12	8-3	14-15	59-2	44-8	98-0	167-0	174-0
Vít-Emp-Der	2-4-4	5-4-1	3-3-4	10-0-0	9-0-1	10-0-0	10-0-0	10-0-0
Remates	12-18	16-8	26-28	83-12	62-22	125-12	194-1	220-0
Ataq-Meio-Def.	24-50-26	40-37-23	26-44-30	48-36-16	38-40-22	50-40-10	63-31-6	65-31-4
<b>NOMM - Sem MM</b>								
Golos (M-S)	7-5	8-2	23-10	83-0	68-3	138-0	198-0	239-0
Vít-Emp-Der	4-4-2	5-4-1	6-2-2	10-0-0	10-0-0	10-0-0	10-0-0	10-0-0
Remates	19-11	17-5	36-24	111-2	82-16	184-2	225-0	251-0
Ataq-Meio-Def.	32-42-26	46-35-19	32-43-25	56-32-12	46-34-20	71-25-4	70-29-1	76-21-1
<b>NOSLM - Sem SLM</b>								
Golos (M-S)	8-9	10-4	18-12	72-0	72-1	124-0	186-0	211-0
Vít-Emp-Der	3-3-4	8-2-0	6-2-2	10-0-0	10-0-0	10-0-0	10-0-0	10-0-0
Remates	16-14	17-8	34-22	100-6	99-18	160-3	232-2	234-0
Ataq-Meio-Def.	28-48-24	42-41-17	30-40-30	54-36-10	48-36-16	58-36-6	66-31-3	72-27-1
<b>NO - Sem ADVCOM, MM e SLM</b>								
Golos (M-S)	1-45	3-5	8-22	18-6	18-14	64-1	127-0	152-0
Vít-Emp-Der	0-0-10	2-6-2	1-4-5	8-2-0	6-2-2	10-0-0	10-0-0	10-0-0
Remates	3-58	7-13	15-43	42-13	32-26	95-28	171-6	180-0
Ataq-Meio-Def.	14-32-54	11-49-40	17-48-35	24-40-36	24-45-31	21-61-18	42-40-18	52-40-8

*Tabela 35: Resultados das Experiências com Coordenação por Comunicação, Modelização Mútua e Percepção Inteligente*

Analisando os resultados conclui-se que individualmente a metodologia com maior influência no desempenho da equipa é o ADVCOM. Sobretudo contra as equipas Tsinghuaeolus e UvaTrilearn, a simples desactivação da comunicação do FC Portugal faz com a equipa seja derrotada em mais jogos do que os que vence contra estas equipas.

Contra equipas fracas, o efeito detrimental no desempenho não é muito acentuado para nenhuma das metodologias, sendo no entanto mais acentuado também para o ADVCOM.

A influência negativa da desactivação do SLM é mais acentuada em quase todos os casos do que a influência negativa da desactivação da modelização mútua. Exceptuam-se os testes contra a equipa FCP Melbourne em que a desactivação do SLM teve pouca influência competitiva.

A desactivação das três metodologias em simultâneo revela-se desastrosa para o desempenho da equipa. A equipa perde todos os jogos contra os Tsinghuaeolus, sofrendo um total de 45 golos, marcando unicamente um, é derrotada na maioria dos jogos contra os UvaTrilearn e tem inclusivamente muitas dificuldades em vencer o FC Portugal Melbourne. De facto estas três metodologias são complementares permitindo ao agente balancear a informação recebida através da visão, comunicação e predição, na criação de um estado do mundo preciso. A desactivação de uma ou duas metodologias não é ainda crítica pois as metodologias activas permitem ainda isoladamente manter o estado do mundo relativamente actualizado. No entanto, desactivando as três metodologias em simultâneo, o comportamento da equipa altera-se completamente pois o estado do mundo impreciso resultante não permite efectuar correctamente os raciocínios de posicionamento, troca de posições e decisão individual necessários para o seu funcionamento.

### **9.13.5 Análise de Resultados**

Embora o número de experiências realizadas tenha sido bastante elevado, a complexidade do domínio e a dificuldade em encontrar medidas quantitativas que permitam avaliar as metodologias propostas, conduzem à conclusão de que muito mais experiências seriam necessárias para realmente demonstrar a influência das diversas metodologias de coordenação no comportamento global da equipa. Das experiências realizadas conclui-se que as metodologias com maior influência no desempenho global da equipa são: SBSP – Posicionamento Estratégico Baseado em Situações, DPRE – Troca Dinâmica de Posições e Papéis e o ADVCOM – Comunicação avançada. Conclui-se também que metodologias que implicam mudanças táticas e a utilização do treinador têm sobretudo influência contra as melhores equipas de teste.

## 9.14 Conclusões

A aplicação de novas metodologias de coordenação, nomeadamente da coordenação estratégica ao domínio do RoboCup, levou a equipa FC Portugal a vencer um campeonato do mundo e dois campeonatos europeus de futebol robótico simulado. A flexibilidade que esta coordenação permite foi a “arma” principal do FC Portugal nestes campeonatos onde teve de medir forças com equipas compostas por jogadores com muito melhores capacidades individuais.

Através de resultados experimentais obtidos num conjunto alargado de experiências foi comprovado que a capacidade da equipa FC Portugal está directamente relacionada com as suas metodologias de coordenação. Os resultados apresentados demonstram claramente que a utilização de metodologias de coordenação como o SBSP ou o DPRE permitem aumentar a capacidade da equipa para realizar a tarefa cooperativa em causa: jogar um jogo de futebol.

Embora as metodologias de coordenação apresentadas tenham sido testadas através de um conjunto alargado de experiências, muito mais experiências seria necessário realizar de forma a averiguar as interações entre cada tipo de metodologia de coordenação. Por outro lado, a influência de metodologias como a coordenação parcialmente hierárquica ou a utilização de estratégias complexas compostas por múltiplas tácticas com condições de activação baseadas em estatísticas de jogo é muito difícil de analisar. Por vezes, uma única táctica com uma única formação bem definida para um dado jogo, funciona muito bem, sendo muito difícil um treinador conseguir afinar essa táctica e melhorar o desempenho global da equipa.