

4 Simulações

Este trabalho teve três focos de tipos de simulações, com características diferentes, para gerar um processo completo na avaliação do controle baseado em comportamento.

A primeira simulação envolve um processo complexo de controle baseado em comportamento, pois engloba dez comportamentos primários que geram cinco comportamentos complexos executados em paralelo com etapas bem definidas para a execução do objetivo final com sucesso e para verificar todas as vantagens da programação baseada em comportamento.

A segunda simulação demonstra uma situação simples programada com o estilo clássico de controle, e compara a mesma situação programada utilizando o controle baseado em comportamento para verificar as diferenças entre elas.

A terceira simulação apresenta um modelo clássico de programação robótica denominado de modelo predador-presa, que foi totalmente programado baseado em comportamento e comparado experimentalmente com robôs reais.

As simulações não levaram em consideração o conhecimento prévio do ambiente e nenhum dos robôs foi programado para armazenar ações anteriores, com o intuito de tornar ainda mais ágil o processamento dos comportamentos, visto que armazenar todas as ações executadas e processar respostas melhores baseadas nesta função tornariam o processamento muito mais trabalhoso e com um uso de memória muito maior.

4.1. Simulação Completa

Como objetivo deste trabalho, a simulação se utilizará da arquitetura vertical do controle baseado em comportamento denominada esquemas motores, aliada ao uso de campos potenciais.

A tarefa definida para este trabalho foi a de coletar objetos de interesse, na cor azul, e descarregá-los em uma área de depósito, de cor amarela. Este processo consiste em desviar de obstáculos, de cor cinza, paredes, na cor preta e de outro

robô, de cor vermelha, vide figura 36. Este robô vermelho também foi programado utilizando o controle baseado em comportamento para perseguir o robô de coleta. Este, quando executa ações de fuga se comporta como um robô presa que tenta escapar de um robô predador. O robô vermelho apresenta apenas comportamentos de desviar de obstáculos e perseguir o robô verde.

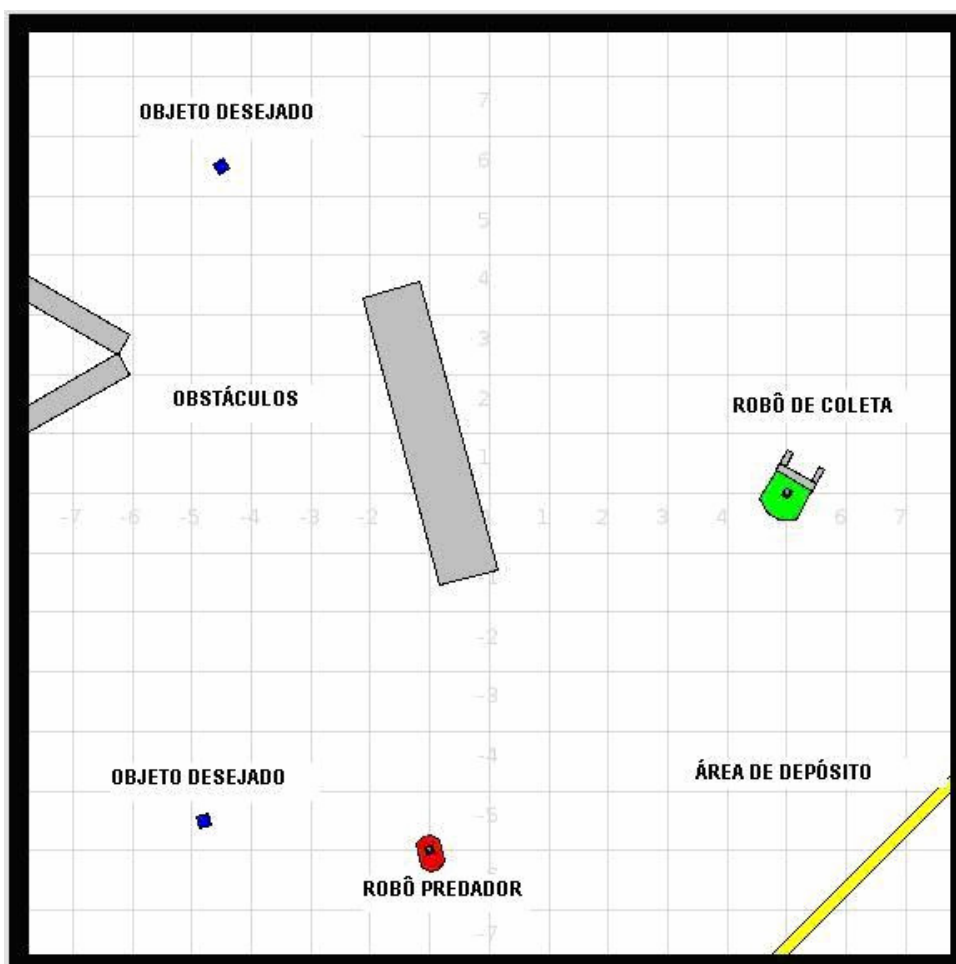


Figura 36 – Ambiente de simulação.

O ambiente de simulação Player/Stage fornece todas as ferramentas necessárias para a simulação do controle baseado em comportamento.

O modelo utilizado usa uma representação em duas dimensões do robô *Pioneer 2-dx*, muito comum em aplicações de robôs pequenos. Foram agregados a ele uma variedade de sensores capazes de perceber o ambiente de várias formas diferentes, como descrito a seguir.

Os sensores e atuadores acoplados e suas funções idealizadas são:

Sensor LASER – Cobre 180° à frente do robô com 360 pontos de medição, o que dá uma precisão angular de uma leitura a cada 0,5°. Programado com a funcionalidade exclusiva de identificar o robô predador, de cor vermelha. Seu alcance foi definido para leitura de até 4 unidades de medida.

Sensor SONAR – Total de 16 sensores dispostos ao redor do robô com a finalidade de detectar obstáculos e paredes. Configurado para uma distância máxima de 4 unidades de medida.

Sensor *BLOBFINDER* – Este sensor é responsável por detectar os objetos de interesse, cor azul, e detectar o local de despejo, na cor amarela.

GRIPPER – O *gripper* adiciona a funcionalidade de capturar e liberar os objetos de interesse.

BUMPER – Os *bumpers* sinalizam para o código do programa que houve uma colisão. Nesta simulação, existem três na parte traseira e dois na parte frontal do robô.

A partir dos sensores programados, devem-se criar os comportamentos primários, ou esquemas perceptivos que, agregados, geram os esquemas motores, ou comportamentos complexos.

Todos os comportamentos são executados simultaneamente e estão sempre contribuindo para o somatório vetorial final.

Os esquemas perceptivos definidos e suas respectivas funções são:

AndaGira – Este comportamento determina que o robô deve andar em linha reta durante um determinado tempo, depois girar em seu próprio eixo aproximadamente 270°, andar novamente em linha reta, e logo após girar no sentido oposto, ou seja, -270°. Esta instrução garante que o robô navegue cobrindo a maior parte do terreno sem retornar pelo mesmo caminho. O valor de 270° é heurístico, não foi feita nenhuma otimização relativa à exploração.

Busca – Ao identificar o objeto azul, o comportamento busca tem seu ganho elevado e começa a contribuir de forma muito significativa para o somatório geral a fim navegar o robô para próximo do objeto. É o campo potencial atrativo gerado pelo objeto que causa esta resposta.

Fechar Garra – Este comportamento só tem sua contribuição iniciada quando os sensores da garra detectam que o objeto azul se encontra posicionado

corretamente para ser capturado. Só então a garra é fechada, e logo após este comportamento deixa de contribuir.

Foge – Este comportamento contribui significativamente quando detecta o robô predador e o campo potencial repulsivo, gerado pelo mesmo, faz com que o robô se afaste do robô predador. As componentes vetoriais geradas por este comportamento variam de intensidade conforme a proximidade do predador, seguindo o equacionamento descrito para campos potenciais repulsivos.

Desvia – Tem a função de desviar dos obstáculos estáticos do ambiente como as paredes e os objetos cinza dispostos no mundo simulado. Este comportamento também utiliza o campo potencial repulsivo para desviar dos obstáculos, e tem constantemente sua contribuição alterada durante a execução. Pela grande quantidade de sensores e obstáculos existentes, o robô está constantemente se acomodando da melhor maneira possível para evitar obstáculos estáticos.

Colisão – Este comportamento foi programado para detectar colisões iminentes, ou seja, quando os sensores de proximidade detectarem que algum objeto, tanto estático quanto dinâmico, está a menos de 1,5 unidades de medida; sua contribuição se torna alta para tentar evitar ao máximo que haja alguma colisão.

Ruído – O comportamento ruído está constantemente contribuindo com uma pequena parcela na soma vetorial, apenas para garantir que o robô escape de situações singulares onde 2 campos potenciais que estejam atuando sobre o robô tenham forças exatamente opostas e se anulem. Com este comportamento, pode-se garantir que algum dos campos irá prevalecer, fornecendo alguma orientação sobre o robô.

Entrega – Uma vez que o objeto azul foi capturado pela garra e o local de depósito foi detectado, este comportamento é responsável por contribuir para que o robô navegue diretamente para ele.

Abrir Garra – Este comportamento só tem sua contribuição iniciada quando o robô se encontra dentro da área de depósito do objeto azul. Só então a garra é aberta para liberação do objeto. Após a liberação, a garra retorna à posição aberta, o robô gira 180° em seu próprio eixo, e então este comportamento deixa de contribuir.

Batida – Quando efetivamente ocorrer uma colisão indicada por algum dos sensores do tipo *bumper*, todas as contribuições dos outros comportamentos têm seu ganhos zerados para que este comportamento de emergência possa recuperar a trajetória do robô. Este processo acontece muito rapidamente, exatamente para representar a reação de girar para o lado oposto de onde ocorreu a colisão.

Todos os comportamentos definidos até agora são esquemas perceptivos puramente reativos, mas não conseguiriam executar a tarefa por completo se não houvesse o complemento da teoria do controle baseado em comportamento que orienta a criação dos esquemas motores, ou comportamentos complexos e um arbitrador central para determinar quais comportamentos complexos devem ser ativados em cada momento. Este arbitrador, em sistemas com dezenas de comportamentos, pode alternar entre os esquemas motores utilizando funções lineares ou não lineares, porém neste trabalho executar esta troca de forma binária e sequencial trouxe resultados satisfatórios, tornando a programação mais simples.

Os esquemas motores estão definidos da seguinte forma:

- **EXPLORAR:**

Este é o esquema motor inicial quando o robô se encontra no ambiente e não está detectando nem está carregando o objeto azul. Este comportamento complexo é gerado pela contribuição dos comportamentos AndaGira, Foge, Desvia, Colisão, Ruído e Batida. Ou seja, o robô está explorando o ambiente em busca do objeto desejado, mas com as funcionalidades que os comportamentos citados lhe fornecem. Apenas estes esquemas perceptivos contribuem para a soma vetorial quando este comportamento é ativado pelo arbitrador.

- **AQUISITAR:**

O arbitrador só aciona este comportamento quando o robô está com o comportamento EXPLORAR ativado e detecta um objeto azul. Isto determina que o robô não esteja mais explorando o ambiente em busca do objeto, e sim que deve entrar no processo de aquisitá-lo. É a junção dos esquemas perceptivos Busca, Foge, Desvia, Colisão, Ruído e Batida.

- **PEGAR:**

Assim que o objeto azul entra na área de captação da garra, o arbitrador ativa este comportamento. Ele é composto apenas pelos esquemas perceptivos Fechar Garra e Foge, pois no momento em que ocorre a captação do objeto pela

garra o robô está parado, portanto não precisa desviar de nenhum obstáculo nem se preocupar com colisões, apenas em caso de aproximação do robô predador é que ocorrerá a soma vetorial.

- **ENTREGAR:**

É similar ao comportamento EXPLORAR, porém ao invés de estar explorando o ambiente à procura do objeto azul, procura-se a área de depósito amarela. Possui os comportamentos simples Entrega, Foge, Desvia, Colisão e Batida. Este esquema motor é ativado assim que a garra termina de capturar o objeto azul.

- **SOLTAR:**

Assim que o robô entra na faixa de depósito, o arbitrador ativa este comportamento, que tem como única finalidade abrir a garra para despejar o objeto azul na zona amarela, e logo após girar 180° em seu próprio eixo. Após este giro, o arbitrador ativa novamente o esquema motor EXPLORAR, fechando o ciclo de organização dos comportamentos complexos. Apenas os comportamentos simples Abrir Garra e Foge contribuem no somatório vetorial geral enquanto este esquema motor está ativo.

A combinação e codificação dos esquemas perceptivos e motores, assim como o funcionamento do arbitrador podem ser visto no apêndice II.

A descrição do controle do robô predador é uma simplificação do robô de coleta, pois este não possui as tarefas de captura e entrega de objetos, ele apenas é programado para explorar e perseguir o robô de coleta. Portanto, o robô predador apresenta apenas os esquemas perceptivos AndaGira, Desvia, Colisão, Ruído e Batida. Os esquemas motores gerados a partir destes esquemas perceptivos são o EXPLORAR, que na essência é o mesmo comportamento do robô de coleta, mas buscando o robô verde ao invés do objeto azul, e o esquema motor PERSEGUIR, que se assemelha ao comportamento AQUISITAR. Contudo se diferencia no objetivo a ser alcançado e também no campo potencial gerado, que no objeto azul é um campo atrativo decrescente com a proximidade do objeto e, no robô verde, um campo atrativo uniforme, tentando maximizar a velocidade da perseguição para aumentar as chances de cumprir o objetivo.

As figuras 37 e 38 apresentam a topologia dos esquemas perceptivos e motores, e sua organização frente ao arbitrador de cada robô.

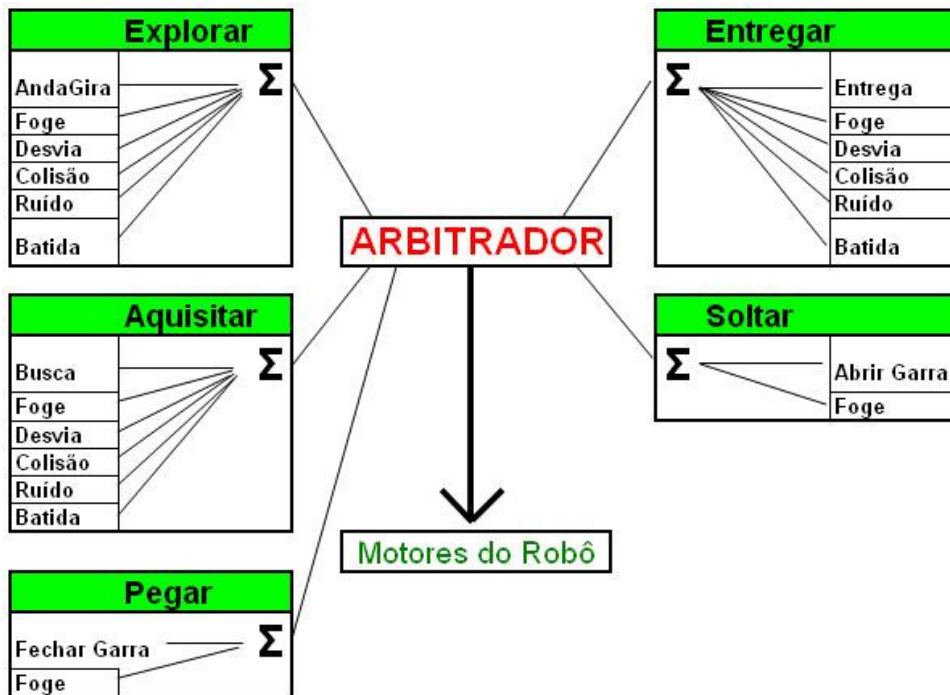


Figura 37 – Topologia dos esquemas motores e perceptivos do robô coleta.

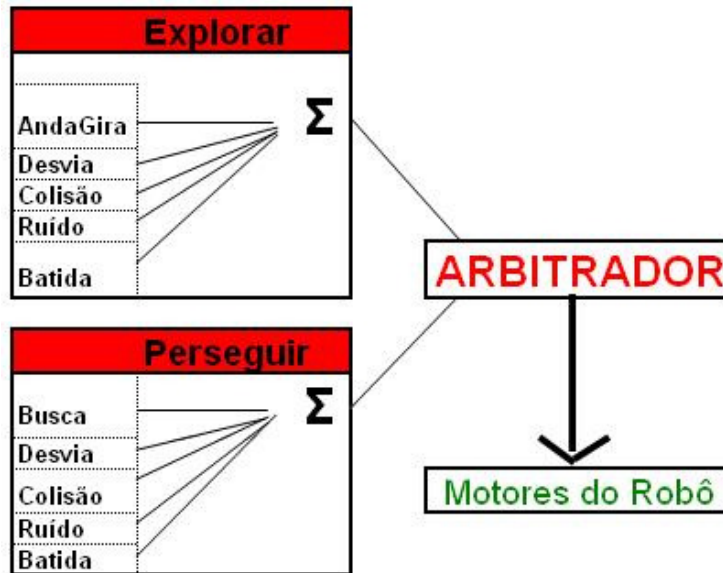


Figura 38 – Topologia dos esquemas motores e perceptivos do robô predador.

O player/stage trabalha com unidades adimensionais para os parâmetros dos robôs. As velocidades máximas do robô de coleta variam entre -0,7 (para trás)

e 1 (em frente) e do robô predador variando -0,3 (para trás) e 0,61 (em frente). A velocidade angular de ambos foi definida entre -5 (para esquerda) e 5 (para a direita). O início das simulações mostrou que as equações que geram os campos potenciais conseguem executar todas as tarefas de maneira satisfatória por isso, os ganhos relativos aos esquemas perceptivos foram definidos de forma binária, 0 e 1. Desta forma o arbitrador determina quais esquemas perceptivos estarão ativos em cada esquema motor. Esta configuração torna a programação mais transparente, pois é possível identificar de forma clara qual comportamento é o predominante em cada momento da simulação.

Exemplos de trajetórias do robô de coleta, para cada esquema motor em separado, são apresentadas a seguir. (Figura 39).

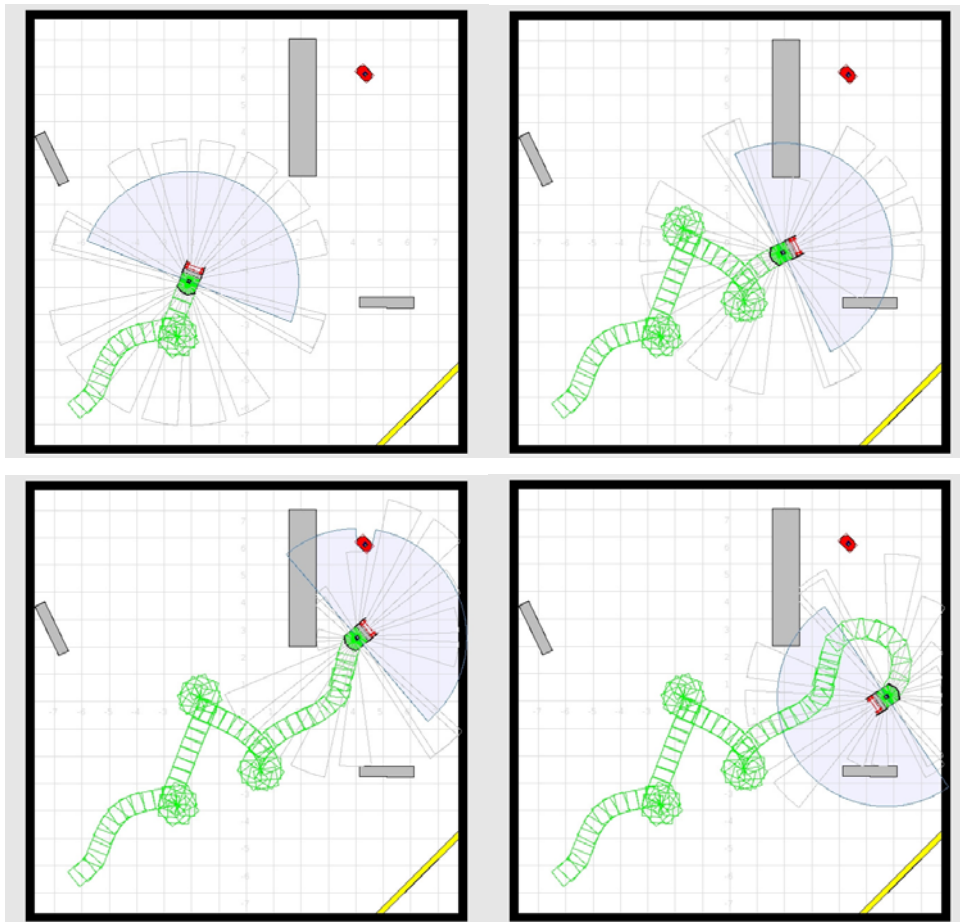


Figura 39 – Trajetória executada pelo robô de coleta apenas com o comportamento EXPLORAR ativo.

Na figura 39, o robô de coleta, em sua situação inicial tem o esquema motor EXPLORAR ativo e inicia o processo de exploração do ambiente à procura do objeto azul, com os outros esquemas perceptivos sempre contribuindo para evitar colisões e fugir do predador. A situação representada na figura 39 não possui os objetos azuis para garantir que apenas o comportamento EXPLORAR esteja ativo. Inicialmente o robô se encontra na parte inferior esquerda do ambiente e começa a se locomover. Logo no início da locomoção já é possível ver que o esquema perceptivo Desvia está contribuindo significativamente, orientando o robô para a melhor trajetória que se afaste das paredes. É uma reação ao campo potencial repulsivo das paredes.

Assim que o processo é iniciado, todos os esquemas perceptivos agregados ao esquema motor EXPLORAR contribuem para a locomoção do robô, e logo após o início da trajetória o esquema perceptivo AndaGira orienta o robô para girar 270° em seu próprio eixo para verificar se existe algum objeto desejado ou predador ao seu redor.

A segunda parte da figura mostra que a trajetória do robô já explorou mais da metade da área do mapa e está caminhando para uma área onde há um robô predador.

Assim que o robô detecta o robô predador através do sensor LASER, ou seja, assim que o robô vermelho entra na faixa de detecção do sensor, o comportamento fuge contribui com grande importância para a soma vetorial e se torna praticamente dominante. Caso esta contribuição não fosse alta, o campo potencial do obstáculo cinza, à esquerda, e da parede, à direita, faria com que o robô fosse diretamente de encontro ao predador. Ou seja, o campo potencial repulsivo do robô predador gera componentes vetoriais de alta magnitude e fazem com que o robô desvie corretamente, tomando uma trajetória contrária como visto na última etapa.

O robô inimigo, chamado aqui de predador, não é considerado um membro nocivo nesta simulação, ou seja, ele é apenas um obstáculo dinâmico a ser evitado. Uma analogia para esta escolha seria o robô de coleta ser um robô de escritório e o robô predador representar uma pessoa caminhando. O robô de coleta não tem como objetivo se desvencilhar do robô inimigo e sim desviar para evitar colisões. Comportamentos no qual um robô deve atacar e o outro fugir são discutidos na simulação predador/presa.

O robô predador tem seu alcance de visão propositalmente maior que o do robô de coleta e sua velocidade menor, para agregar diferenças significativas entre ambos.

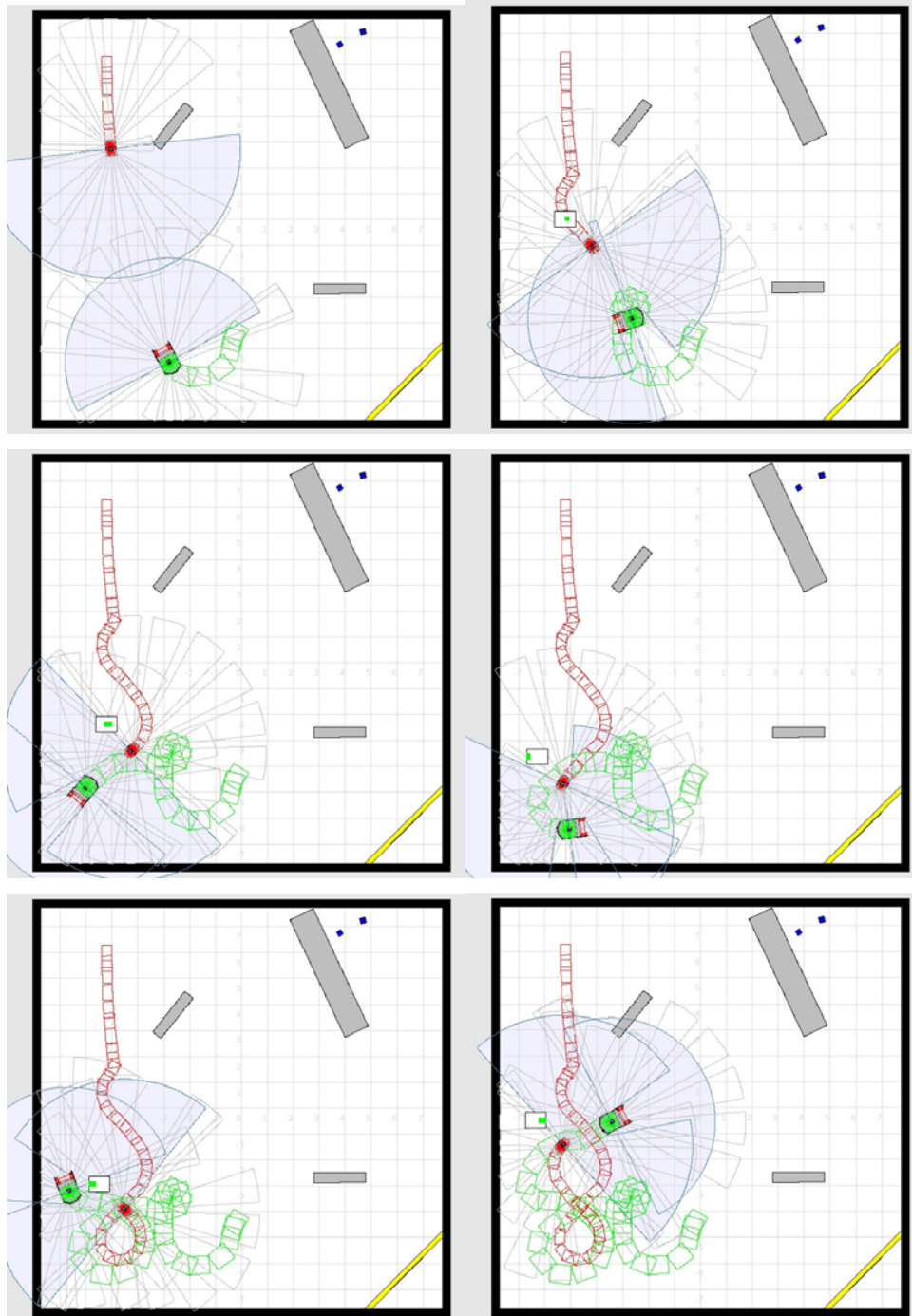


Figura 40 – Trajetória de desvio do robô predador feita pelo robô de coleta.

Na trajetória da figura 40, os robôs estão em rota de encontro a partir do momento em que o robô de coleta entra na faixa de sensoriamento do robô predador. Automaticamente o esquema motor PERSEGUIR é ativado pelo arbitrador do robô predador, fazendo com que o campo atrativo gerado pelo robô de coleta exerça forças diretamente para sua posição. Até este momento, o robô de coleta não identificou o predador por ter um alcance menor em seus sensores, e prossegue explorando sem contribuições do esquema perceptivo FOGUE.

Após esta parte, começa o processo no qual os robôs se identificam e só então o robô de coleta inicia uma trajetória para desviar do robô predador, agregando as contribuições dos esquemas perceptivos ativos.

Nas quatro partes restantes da figura 40 nota-se o robô de coleta sendo guiado pelo somatório dos campos potenciais das paredes próximas e do forte campo repulsivo gerado pelo robô predador, até conseguir ir para uma área aberta.

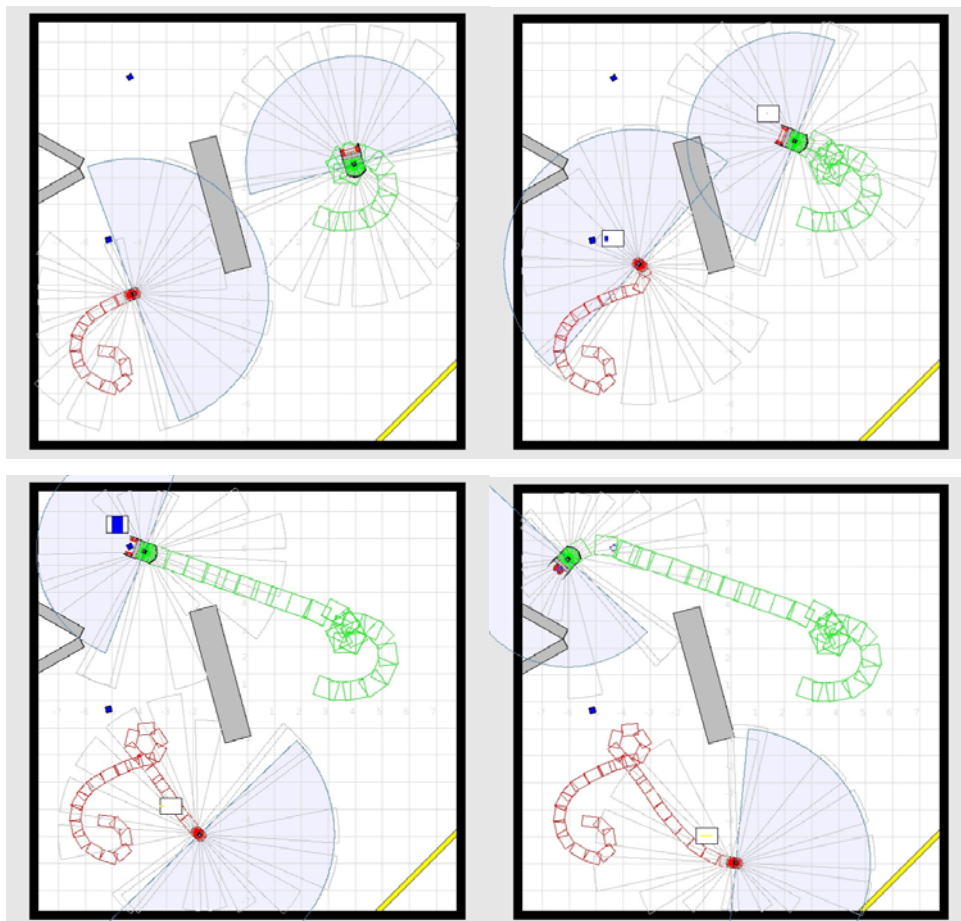


Figura 41 – Trajetória dos esquemas motores AQUISITAR, PEGAR e a ativação do ENTREGAR.

Na figura 41, o robô de coleta está executando o giro em torno de seu eixo explorando o ambiente, até que o sensor *BLOBFINDER* detecta a presença do objeto desejado e a trajetória circular é interrompida imediatamente, pois neste momento o arbitrador ativa o esquema motor AQUISITAR. O objeto desejado agora exerce um campo potencial atrativo decrescente sobre o robô, gerando uma trajetória praticamente linear em direção ao objeto. O termo AQUISITAR é sinônimo do verbo adquirir e amplamente utilizado no meio técnico.

Percebe-se que a influência dos esquemas perceptivos de desvio e contra colisões tem sua contribuição diminuída, pois o robô passa bem próximo ao obstáculo cinza e não sofre alteração significativa em sua trajetória. Esta é uma característica projetada, pois o objetivo principal da simulação é o objeto azul e, com isso, o campo potencial gerado pelo objeto azul é tão forte que os vetores que fariam o robô desviar não têm magnitude suficiente para mudar a trajetória do robô, a não ser que este entre em rota de colisão com algum obstáculo ou predador. Neste caso, como as equações que estabelecem os campos potenciais repulsivos geram magnitudes inversamente proporcionais ao quadrado da distância, em proximidades pequenas os vetores de desvio corrigiriam a trajetória do robô.

Uma vez que o campo potencial atrativo do objeto azul conduz o robô a parar com o objeto dentro da área de captação de sua garra, o arbitrador ativa o esquema motor PEGAR. Neste comportamento, o robô fecha a garra para poder carregar o objeto, e a única contribuição possível neste momento é do esquema perceptivo Foge, pois não há necessidade de desviar de obstáculos visto que o robô está parado.

Ao fechar a garra, tem-se a ativação do esquema motor ENTREGAR, e se inicia a navegação em busca do local para depósito.

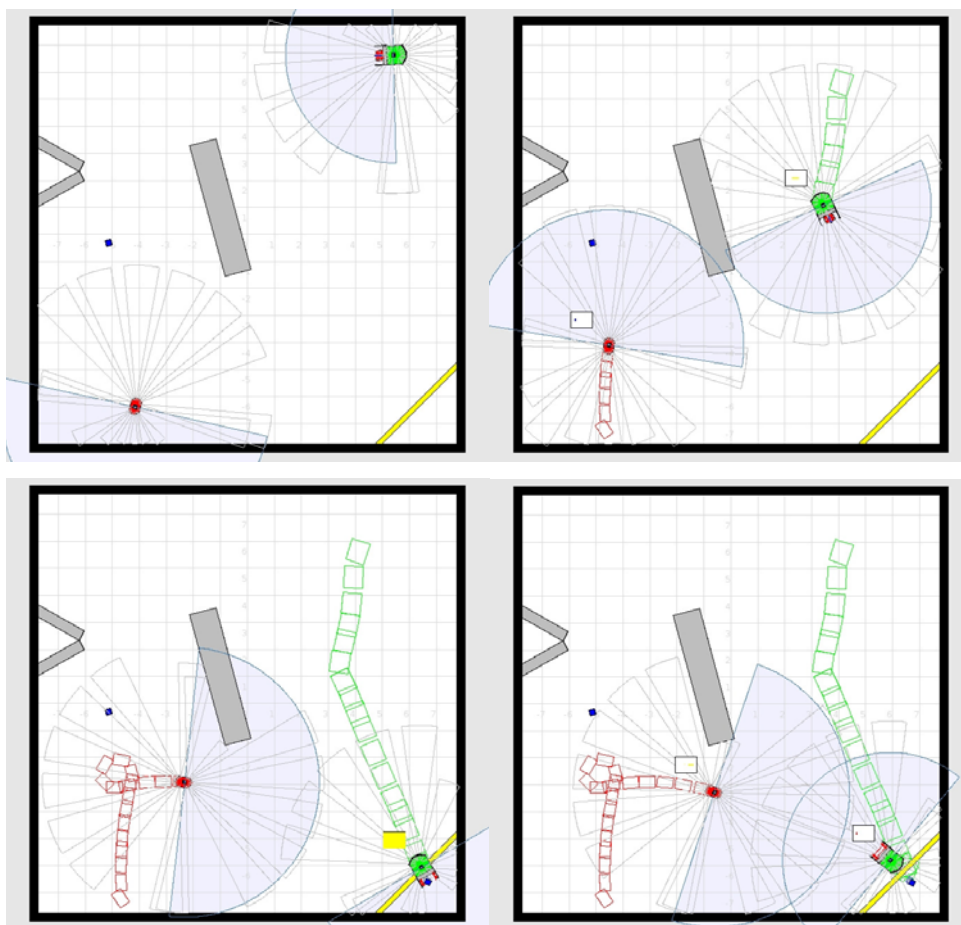


Figura 42 – Trajetória dos esquemas motores ENTREGAR, SOLTAR e a ativação do EXPLORAR.

O processo ENTREGAR procura pela área de depósito como visto na figura 42, e assim que a detecta entra em uma trajetória similar à descrita no comportamento AQUISITAR, onde os campos potenciais de desvio, fuga e contra colisões só influem de maneira significativa em casos iminentes de colisão. Desta maneira, o robô ao detectar a zona amarela é imediatamente atraído pelo campo potencial atrativo decrescente gerado por ela e, ao entrar na zona de depósito, o arbitrador aciona o esquema motor SOLTAR. Este esquema tem como finalidade abrir a garra para depositar o objeto azul e logo após girar 180° sob seu eixo para sair da zona amarela. Logo após esta ação, o arbitrador torna a ativar o esquema motor EXPLORAR, reiniciando o ciclo a procura de outro objeto desejado.

Todos os esquemas motores desta simulação executaram suas tarefas com êxito, o robô alcançou o objetivo mesmo com as dificuldades neste ambiente.

4.2. Comparativo Clássico X Comportamento

A estratégia do controle baseado em comportamento se mostra muito útil na programação de robôs móveis autônomos, contudo faz-se necessária uma comparação com a técnica tradicional de programação que não leva em consideração os modelos comportamentais e utiliza a arquitetura horizontal para executar as tarefas.

Nesta simulação, utilizaram-se os mesmo dois robôs da simulação anterior, porém o objetivo do robô de coleta, de cor verde, foi reduzido para apenas navegar pelo ambiente desviando de obstáculos e do robô vermelho. O objetivo do robô vermelho também não foi modificado, permanecendo explorar e perseguir o robô verde.

4.2.1. Controle Baseado em Comportamento

A parte da simulação programada utilizando o controle baseado em comportamento contempla um único esquema motor NAVEGAR, com esquemas perceptivos AndaGira, Foge, Desvia, Ruído e Batida.

A programação do robô vermelho continua com dois esquemas motores EXPLORAR e PERSEGUIR com seus respectivos esquemas perceptivos, vide figura 43.

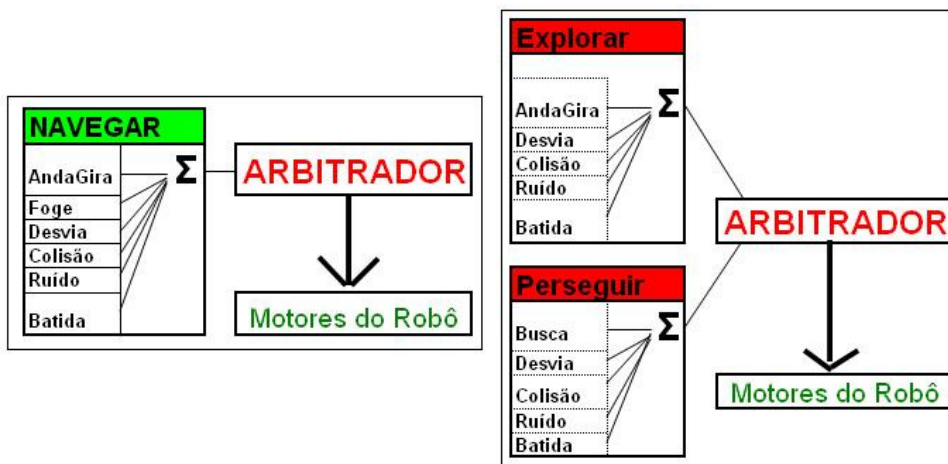


Figura 43 – Topologia do robô verde e do robô vermelho programados com controle baseado em comportamento.

A figura 44 apresenta o comportamento que os robôs apresentam em uma simulação simplificada.

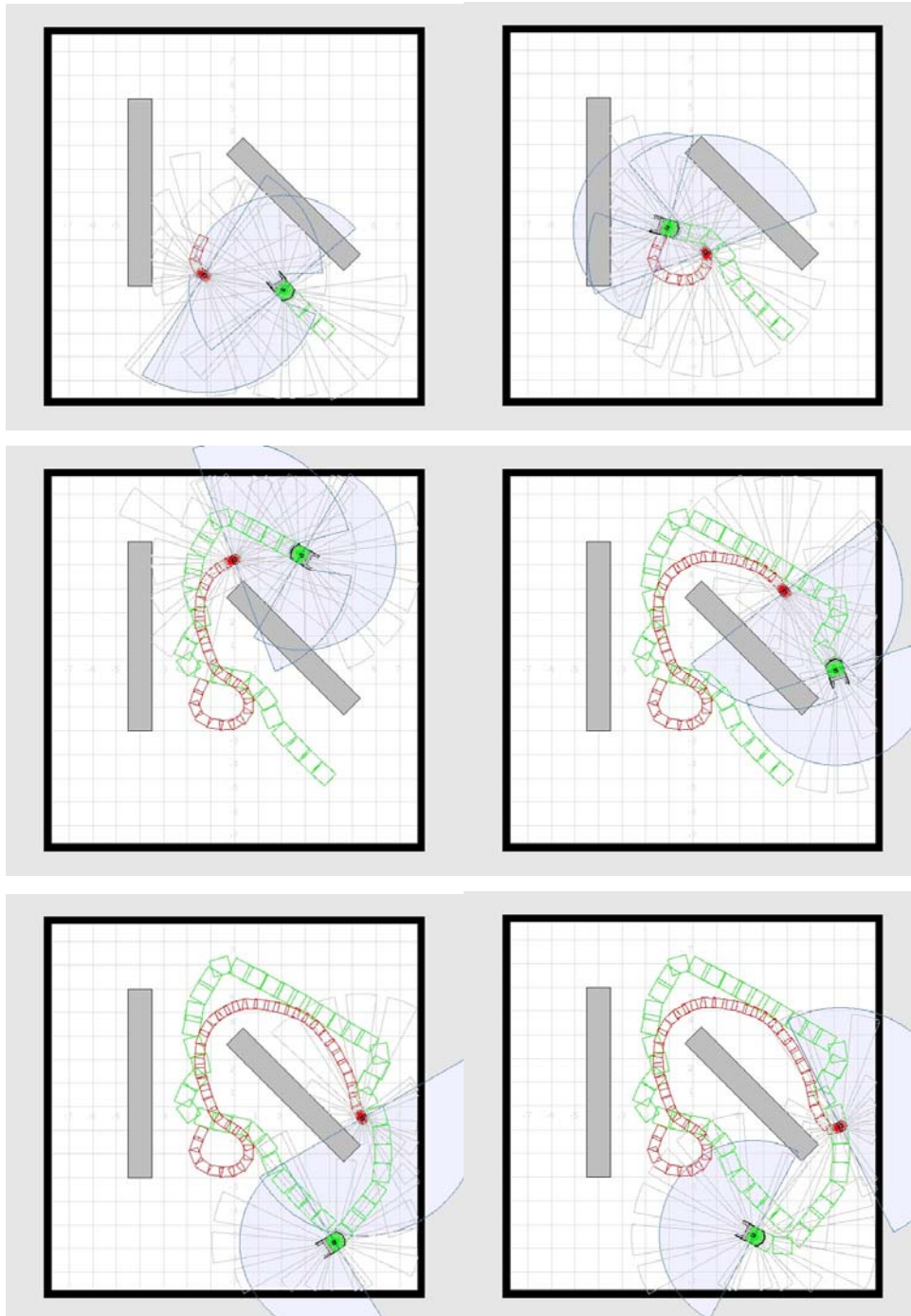


Figura 44 – Trajetória dos robôs controlados por comportamento. Robô verde desvia e escapa do robô vermelho.

Esta simulação (Figura 44) demonstra o bom desempenho do controle baseado em comportamento, utilizando campos potenciais, que foram responsáveis pela correta trajetória do robô verde ao desviar dos obstáculos estáticos e dinâmicos do ambiente.

O estado inicial mostra que os robôs estão em rota de colisão e o campo potencial repulsivo crescente gerado pelo robô vermelho gera forças vetoriais que forcem o correto desvio por parte do robô verde. Após a detecção do robô verde, o robô vermelho tem seu esquema motor PERSEGUIR ativado pelo arbitrador, e inicia o processo de perseguição por todo o ambiente. Apesar de possuir sensores com maior alcance, o robô vermelho é mais lento e por isso, o robô verde consegue sair do alcance do vermelho, apesar da influência dos obstáculos.

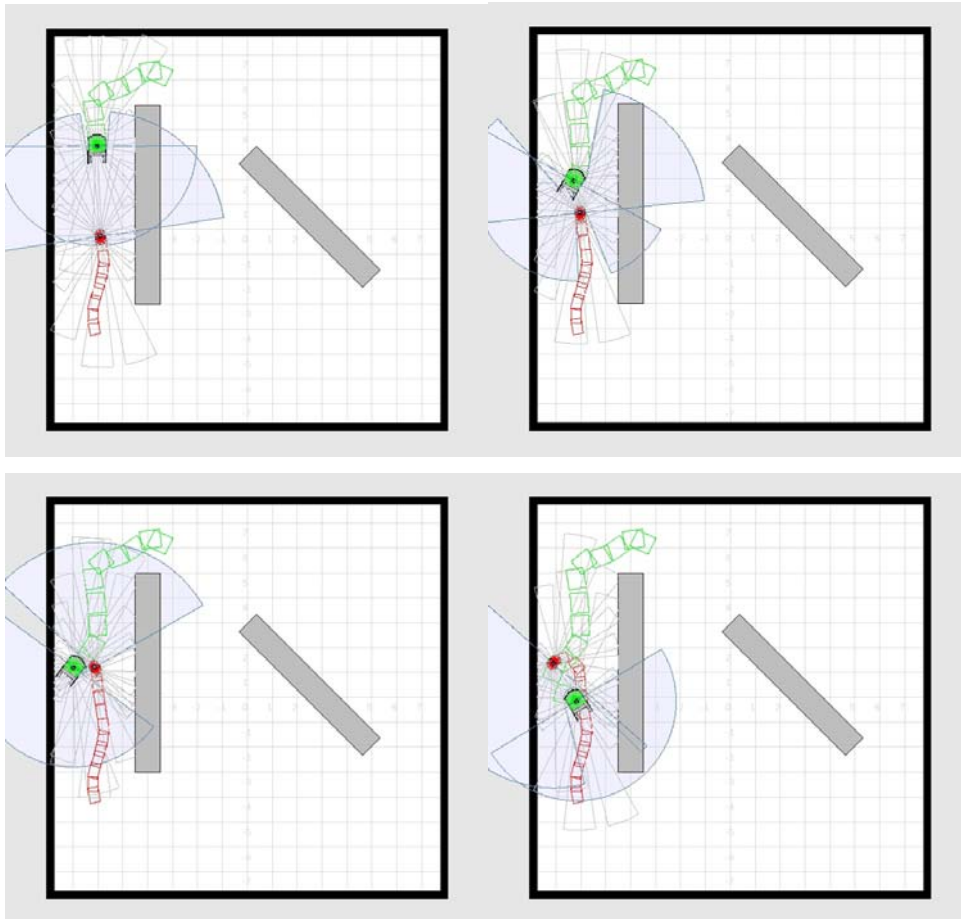


Figura 45 – Trajetória dos robôs verde e vermelho (baseados em comportamento) em rota de colisão em um corredor estreito.

Na figura 45, um corredor estreito é utilizado para verificar a trajetória realizada pelos robôs. Assim que entram no corredor, ambos os robôs sofrem as mesmas forças geradas pelos campos potenciais das paredes, e admitem uma trajetória linear e igualmente afastada de ambos os lados. Esta trajetória gerada cria uma rota de colisão entre os robôs. Cabe ao robô verde a tarefa de tentar desviar do robô vermelho.

O campo potencial repulsivo crescente criado pelo robô vermelho começa a influenciar o robô verde, que pouco altera sua rota, pois está sofrendo as forças laterais das paredes. O campo repulsivo cresce inversamente com o quadrado da distância entre os robôs, ou seja, quando o robô verde se aproxima muito do vermelho, as forças repulsivas tornam-se maiores que as forças das paredes, forçando o desvio.

Uma observação sobre o campo repulsivo é que a magnitude da velocidade também é uma grandeza que cresce inversamente com o quadrado da distância. Este fato é um dos responsáveis para o sucesso na tarefa de desviar do robô vermelho.

Uma possível interpretação de fuga seria fazer com que o robô verde girasse 180° assim que avistasse o robô vermelho. Este fato não acontece pois a simulação foi concebida com objetivo de capturar objetos e ao mesmo tempo desviar de obstáculos. Ainda neste caso o campo potencial repulsivo tende a girar o robô verde mas o campo repulsivo das paredes é dominante e logo após o desvio os campos potenciais das paredes retomam a rota centralizada de ambos.

4.2.2. Controle Clássico de Trajetórias

A fim de realizar a comparação, as duas programações, baseada em comportamento e clássica, foram totalmente refeitas utilizando a teoria clássica de programação horizontal. Esta divide o controle em unidades de função dependentes umas das outras, onde as informações captadas pelos sensores são utilizadas para representar o ambiente a cada instante. A partir desta representação, ocorre o planejamento da ação a ser executada para atingir seu objetivo. O planejamento é executado em tempo real e utilizado para navegar o robô até encerrar o ciclo de controle e, logo após, reiniciá-lo.

Nesta simulação com controle clássico não foi utilizado o conceito de campos potenciais visto que sua implementação em um sistema onde as funções são dependentes umas das outras dificulta o processo de somatório dos campos vetoriais.

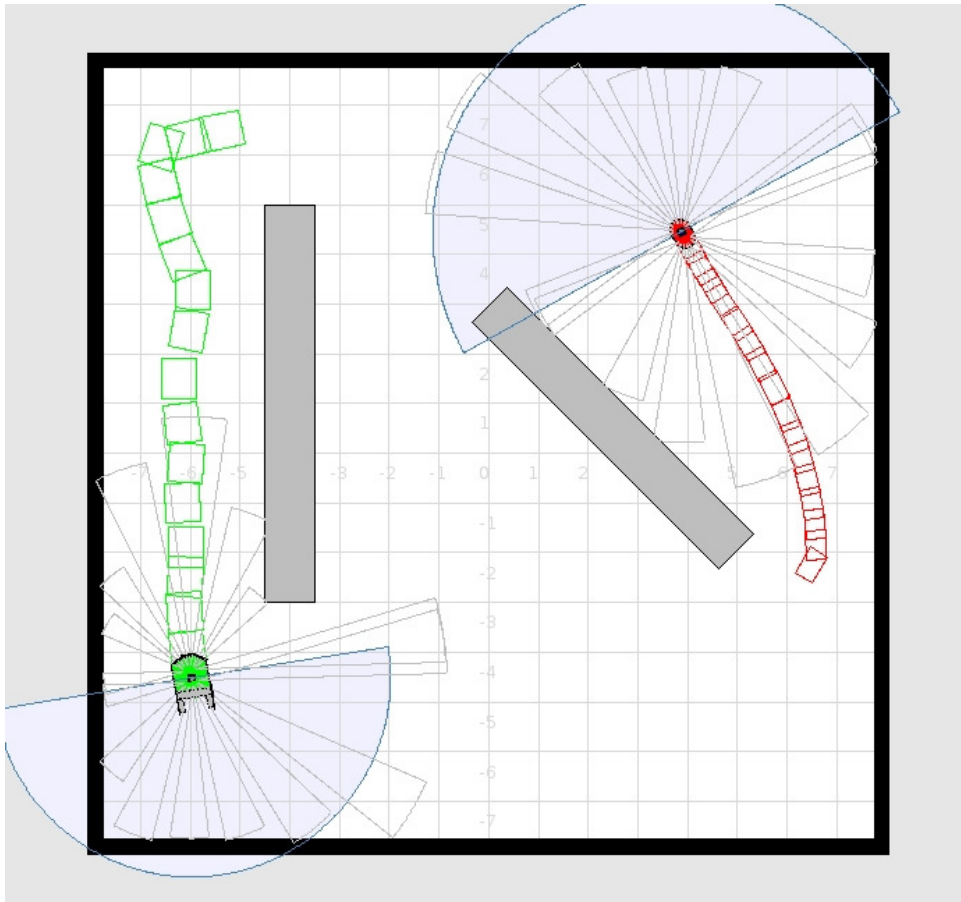


Figura 46 – Trajetória não linear do robô verde dentro do corredor.

A reprogramação dos robôs utilizando o método clássico trouxe características diferentes.

Primeiramente, vê-se na figura 46 que o robô verde não consegue executar uma trajetória perfeitamente linear quando se encontra sozinho dentro de um corredor estreito. Este fato se deve ao estilo da programação clássica, que executa módulos de função separadamente e sequencialmente; isto significa que quando o robô entra no corredor, o módulo responsável pelo desvio detecta que o obstáculo à direita é o que está mais próximo, sendo assim executa ordem para o robô girar à esquerda. Esta ordem de execução faz com que o robô gire para a esquerda e comece a navegar para longe do obstáculo à direita, porém, logo depois que esse

processo se inicia, o obstáculo à esquerda se torna o mais próximo ao robô, fazendo com que o módulo de desvio dê a ordem inversa.

Esta troca de operações faz com que o robô não obedeça a uma trajetória tão retilínea quanto à vista no controle utilizando campos potenciais.

Este problema poderia ser solucionado realizando um ajuste fino para cada situação específica que o robô vier a enfrentar, porém isto demanda mais gasto computacional, tempo de programação e aumento da complexidade do código gerado.

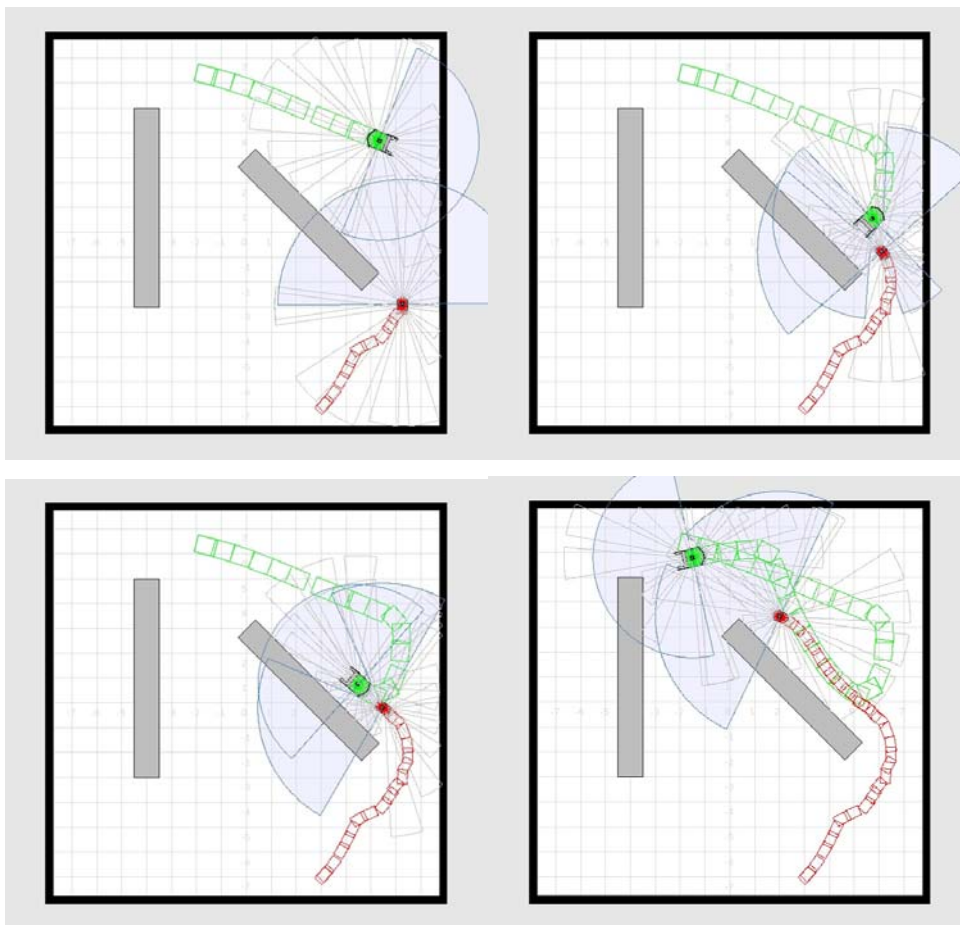


Figura 47 – Trajetória de desvio dos robôs programados com método clássico.

As curvas dos robôs são mais bruscas (Figura 47) pelo fato de não haver a soma vetorial repulsiva e atrativa. Esta característica não impede seu funcionamento, mas perde-se na fluidez do movimento como se vê,

provavelmente gerando maior gasto de energia em um sistema real. Ao aproximar-se da parede o módulo responsável pelo desvio executa a ordem de giro assim que detecta a parede, isso faz com que o robô gire rapidamente 90° para evitar a colisão, e segue em frente.

À frente, ao detectar o robô vermelho, o módulo de fuga executa a ordem de giro, que nesta situação é no mesmo sentido do desvio do obstáculo que está próximo. Neste caso, a tarefa de desvio foi realizada com sucesso, mas com movimento abruptos que em experimentos reais podem causar incertezas de medições.

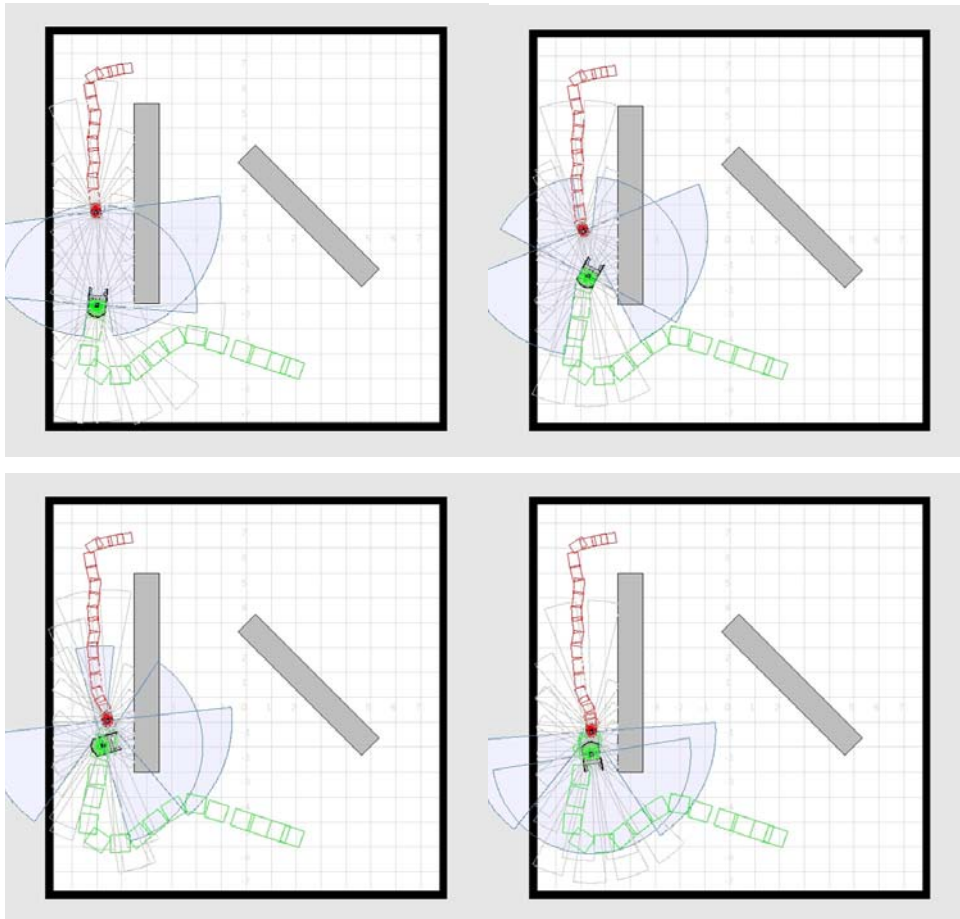


Figura 48 – Colisão entre os robôs programados com método clássico.

A mesma situação do corredor estreito é aplicada ao controle clássico (Figura 48). A aproximação do robô verde ocorre de maneira não retilínea como

descrito anteriormente, este fato contribui para a incapacidade de realizar o desvio.

O módulo responsável pelo desvio das paredes opera normalmente até que o robô vermelho é detectado. Neste momento, o módulo responsável por fugir atua sobre os motores para que girem à direita com grande magnitude. Logo após esta ordem, o módulo responsável por desviar das paredes também atua no robô para afastá-lo da parede, contudo como no modelo clássico não existe o método dos campos potenciais crescentes, a intensidade do giro dos motores para fugir do robô vermelho é maior do que a de desviar das paredes. Isto causa uma colisão com as paredes e conseqüentemente com o robô vermelho, até que depois de três colisões o robô consegue girar e tomar o caminho contrário.

4.2.3. Comparação

A programação clássica apresentou desvantagens significativas em relação ao controle baseado em comportamento aliado aos campos potenciais:

A interdependência dos módulos de funções do controle clássico deixa o código “amarrado” em relação à sua execução. Os módulos são executados sequencialmente, gerando respostas conflitantes que podem causar incertezas nos experimentos reais. Perde-se a estrutura modular. Caso seja necessário adicionar um novo módulo de função, toda a estrutura do código fonte deve ser alterada, causando retrabalho desnecessário.

No controle por comportamento, todos os módulos comportamentais são executados em paralelo, o que traz alta modularidade. É possível adicionar um novo comportamento sem praticamente nenhuma alteração no código original.

As derivadas de primeira e segunda ordem da posição com relação ao tempo fornecem, respectivamente, a velocidade e a aceleração do objeto. Menos conhecida é a derivada terceira da posição com relação ao tempo, isto é, a taxa de variação da aceleração com relação ao tempo que, em inglês, recebe o nome de *jerk*. Esta derivada é útil, por exemplo, quando se deseja estudar o desgaste pela ação do movimento de um mecanismo sensível ou o desconforto dos passageiros dentro de um veículo. A ausência dos campos potenciais deixa a trajetória dos robôs grosseira, com movimentos bruscos, ou seja, com valores altos da terceira

derivada da posição, causados pelo conflito de respostas diferentes originadas pelos módulos de função.

Esta ausência causa problemas mais frequentes de colisões e indecisões sobre qual melhor direção seguir por parte do robô.

O advento dos campos potenciais trouxe suavidade para o controle do robô, visto que apenas uma força resultante atua no robô o tempo todo, gerada a partir do somatório vetorial de todos os esquemas perceptivos ativos no momento.

4.3. Predador / Presa

Na ecologia, a interação entre diferentes espécies é foco de estudo de diversas áreas da ciência. O modelo predador / presa, no qual um ou mais agentes do tipo predador tem como objetivo caçar um ou mais agentes do tipo presa, é discutido como o problema predador / presa. Este problema é naturalmente visto como uma tarefa cooperativa entre agentes. A estratégia mais eficaz para a solução exige comportamentos cooperativos entre os agentes individuais. O modelo predador/presa foi proposto por Benda (BENDA, 1986) [44] e se tornou referência para pesquisadores avaliarem as técnicas de comportamentos cooperativos.

Neste trabalho, a simulação realizada contempla a perseguição de um robô presa por um robô predador. A programação baseada em comportamento se encaixa no escopo desta tarefa por reproduzir o comportamento animal que pode ser verificado experimentalmente. Esta simulação é a representação de um modelo real existente. Dois robôs identicamente projetados foram programados de acordo com a teoria do controle baseado em comportamento com o método dos campos potenciais, e utilizando o modelo predador / presa simulados.

Os robôs possuem três sensores ultrassom com a finalidade de detecção do outro robô, sendo que o robô predador possui os sensores localizados a esquerda, direita, e à frente, mas o robô presa tem o terceiro sensor posicionado na parte traseira com o intuito de poder detectar a aproximação do robô predador e poder fugir da melhor maneira possível. Caso o ultrassom fosse instalado na parte da frente, o robô presa não teria como fugir eficientemente. Uma vez que este execute um giro no próprio eixo para iniciar a fuga, o robô predador sairia do

ângulo de visão da presa, fazendo com que o robô presa, que não possui memória, acredite que não está mais sendo perseguido.

Existem também quatro sensores infravermelhos para detecção de superfícies brancas. Localizados nas extremidades da base do robô, estes sensores estão apontados para o solo, para detectar regiões e linhas brancas que delimitam o ambiente.

O ambiente simulado contempla um robô predador programado para detectar e perseguir o robô presa, que por sua vez foi programado para detectar e fugir do robô predador. Existem faixas largas nas bordas do ambiente que representam áreas de risco, ou seja, paredes virtuais das quais o robô deve desviar assim que as detectar. O campo potencial gerado pelas paredes virtuais é um campo repulsivo constante; como a largura da faixa de desvio é pequena, o robô deve desviar de maneira rápida para não sair da arena com nenhuma de suas rodas. Ambos têm a mesma capacidade de desvio, contudo o robô presa foi programado para ser mais rápido que o robô predador, pois os comportamentos de fuga e desvio deixam o robô presa muito vulnerável a ser encurralado. A proporção do quanto a velocidade do robô presa é superior foi obtida empiricamente para tentar representar melhor o comportamento animal no qual o predador nem sempre obtém sucesso ao perseguir uma presa.

Os comportamentos definidos em cada robô seguem a teoria do controle baseado em comportamento, utilizando a arquitetura dos esquemas motores e esquemas perceptivos. Os dois robôs diferem no esquema motor CAÇAR do predador e do FUGIR da presa, mais especificamente no esquema perceptivo Busca do predador e Foge da presa. Todos os esquemas são variações dos esquemas utilizados na simulação completa e, detalhados a seguir:

AndaGira – Determina que o robô deve andar em linha reta durante um determinado tempo, depois girar em seu próprio eixo aproximadamente 90° , andar novamente em linha reta, e logo após girar no sentido oposto, ou seja, -90° . Como o terreno tem dimensões reduzidas, esta instrução garante que o robô explore uma área considerável em pouco tempo.

Busca – Presente apenas no robô predador. Ao identificar o robô presa, tem seu ganho elevado e começa a contribuir de forma muito significativa para o somatório geral a fim de navegar o robô para próximo do objeto. É o campo potencial atrativo gerado pelo objeto que causa esta resposta.

Foge – Presente apenas no robô presa. Contribui com alta relevância quando detecta o robô predador e o campo potencial repulsivo gerado pelo mesmo faz com que o robô se afaste do robô predador. As componentes vetoriais geradas por este comportamento variam de intensidade conforme a proximidade do predador, seguindo o equacionamento descrito para campos potenciais repulsivos crescentes.

Desvia – É um comportamento associado a cada um dos quatro sensores infravermelhos, que contribui para o somatório quando cada um deles é ativado individualmente. Tem a função de desviar das paredes virtuais representadas pelas faixas nas bordas do chão da arena. Este comportamento também utiliza o campo potencial repulsivo para desviar dos obstáculos. A presença do robô posicionado com apenas um sensor infravermelho detectando a presença da faixa de desvio indica que o robô deve efetuar um leve desvio, por isso a contribuição deste comportamento tem sua magnitude reduzida. À medida que o robô entra mais na faixa de desvio, outros sensores detectam a faixa branca, acionando outros esquemas perceptivos e contribuindo cada vez mais para o desvio.

Colisão – Este comportamento foi programado para detectar quando o robô entre de frente ou de costas na faixa de desvio, ou seja, quando os dois sensores frontais ou traseiros estão acionados ao mesmo tempo, e os outros dois estão desativados. Nesta situação o campo potencial repulsivo contribui com grande intensidade para que o robô recue e gire, ou avance e gire, para evitar uma invasão por completo na faixa de desvio.

Batida – Quando três sensores infravermelhos estiverem acionados simultaneamente, significa que o robô está em uma situação crítica, ou seja, está posicionado quase que por completo dentro da faixa de desvio. O campo potencial repulsivo neste caso contribui significativamente para navegar o robô de volta para a arena.

Ruído – O esquema perceptivo ruído está constantemente contribuindo com uma pequena parcela na soma vetorial, apenas para garantir que o robô escape de situações singulares onde dois campos potenciais que estejam atuando sobre o robô tenham forças exatamente opostas e se anulem. Com este comportamento, pode-se garantir que algum dos campos irá prevalecer, fornecendo alguma orientação sobre o robô.

Os esquemas motores estão definidos da seguinte forma: (Figura 49).

- **EXPLORAR:**

Este é o esquema motor inicial quando o robô se encontra no ambiente e não está detectando o outro robô. Este comportamento complexo é gerado pela contribuição dos comportamentos AndaGira, Desvia, Colisão, Ruído e Batida. O robô explora o ambiente com as funcionalidades que os comportamentos citados lhe fornecem. Apenas estes esquemas perceptivos contribuem para a soma vetorial quando este comportamento é ativado pelo arbitrador.

- **CAÇAR:**

Presente apenas no robô predador. O arbitrador só aciona este comportamento quando o robô está com o comportamento EXPLORAR ativado e o robô presa à frente ou nas laterais. Isto determina que o robô passe do estado exploratório e passa a ter o objetivo de caçar a presa. Compõem-se dos esquemas perceptivos Busca, Desvia, Colisão, Ruído e Batida.

- **FUGIR:**

Presente apenas no robô presa. O arbitrador só aciona este comportamento quando o robô está com o comportamento EXPLORAR ativado e o robô predador na parte traseira ou nas laterais. Isto determina que o robô passe do estado exploratório a ter o objetivo de fugir do predador, se posicionando de maneira que o robô predador fique diretamente atrás, a fim de atingir a maior velocidade possível de afastamento. Compõem-se dos esquemas perceptivos Busca, Desvia, Colisão, Ruído e Batida.

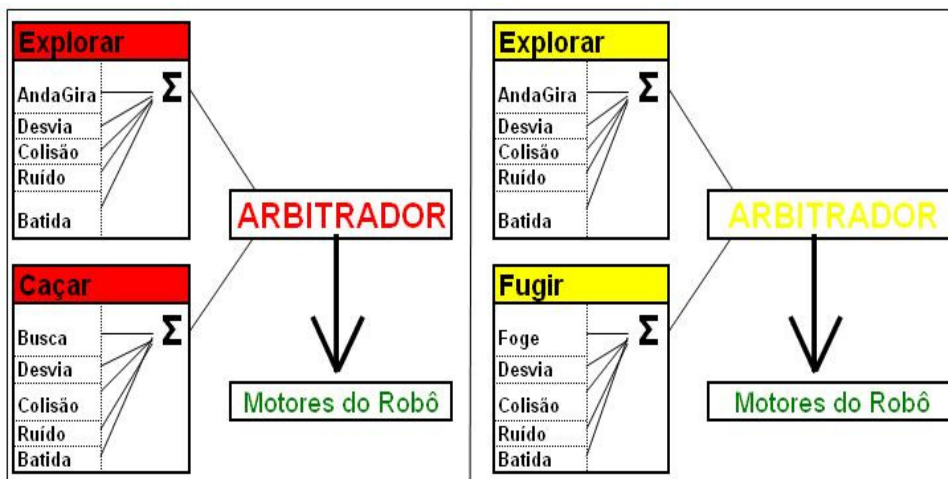


Figura 49 – Topologia do robô predador e do robô presa programados com controle baseado em comportamento.

As simulações apresentam o comportamento dos robôs interagindo na área retangular que também é usada no experimento real. (Figura 50).

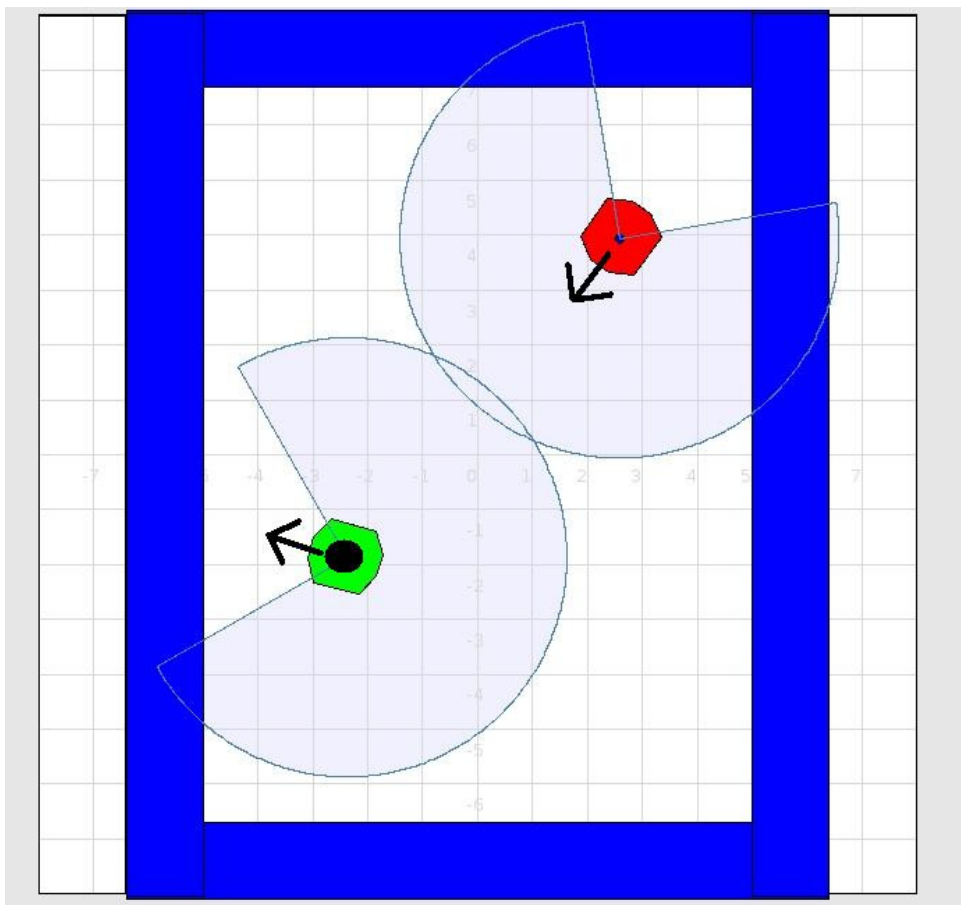


Figura 50 – Robôs predador e presa no ambiente.

A área azulada em volta dos robôs na figura 50 representa a área de detecção dos sensores ultrassom.

No sistema experimental estas áreas/linhas azuladas receberão a cor branca e a parte interna será preta para mais fácil detecção pelos sensores infravermelhos.

Inicialmente vêem-se o ambiente simulado com as paredes virtuais, o robô predador, em vermelho, e o robô presa, em verde e com um círculo preto em seu centro. As respectivas áreas cobertas pelos sensores, no robô predador, localizadas à frente, e no robô presa, na traseira, são representadas na figura 50, assim como as direções de locomoção, indicadas por setas.

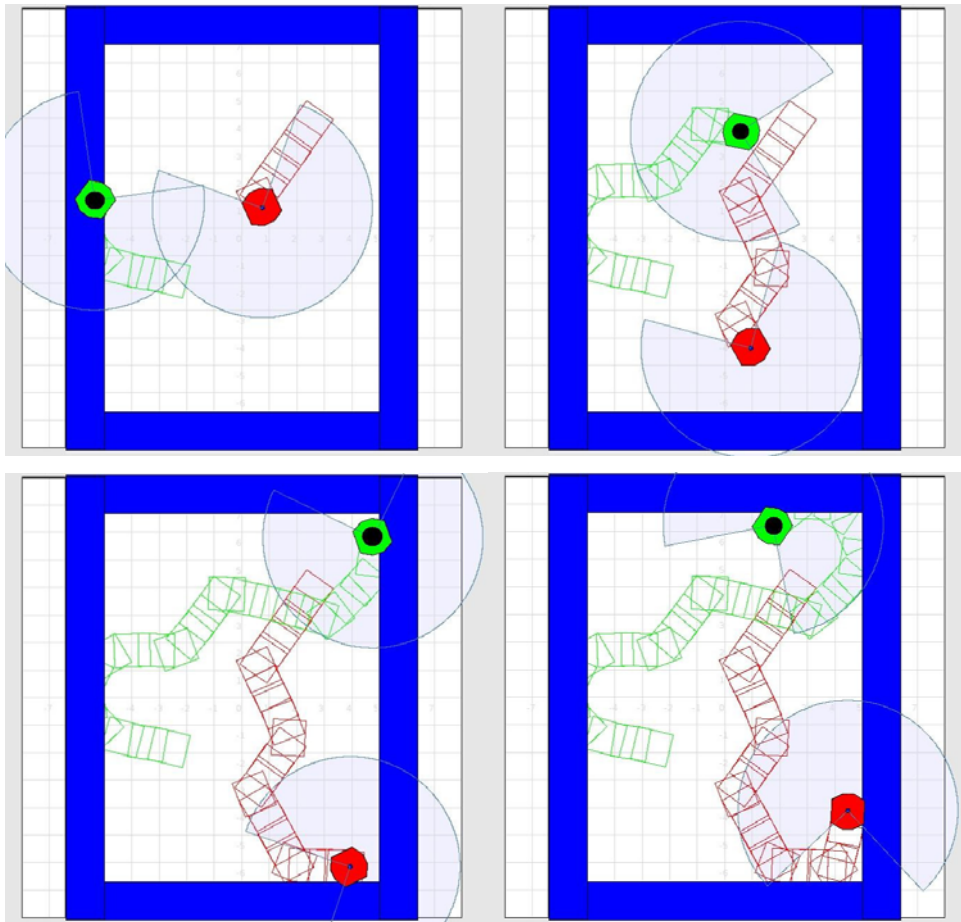


Figura 51 – Trajetória dos robôs predador e presa navegando pelo ambiente.

Iniciado o processo, vêem-se na figura 51 os robôs com o esquema motor EXPLORAR habilitado pelos respectivos arbitradores. O esquema perceptivo do predador inicia seu funcionamento, executando a navegação programada para andar durante um determinado tempo e alternar entre os lados de giro. Já no robô presa, o campo potencial repulsivo gerado pela faixa de desvio produz forças de repulsão que fazem com que o robô gire e se afaste da mesma. Nota-se que o robô presa entrou na faixa de desvio quase que de frente, o que fez com que os três esquemas perceptivos Desvia, Colisão e Batida contribuíssem consideravelmente para o sucesso do desvio.

Logo após, ainda na figura 51, é possível ver a trajetória executada pelos robôs executadas a partir dos respectivos esquemas perceptivos AndaGira, até que ambos se deparam com a faixa de desvio e próximos dos cantos da arena. O robô predador se aproxima com ângulo bem aberto, o que torna o desvio mais suave,

pois se tem tempo suficiente para executar a manobra. Já a presa entra mais uma vez com angulação acentuada, o que faz com que a manobra de desvio seja uma curva suave.

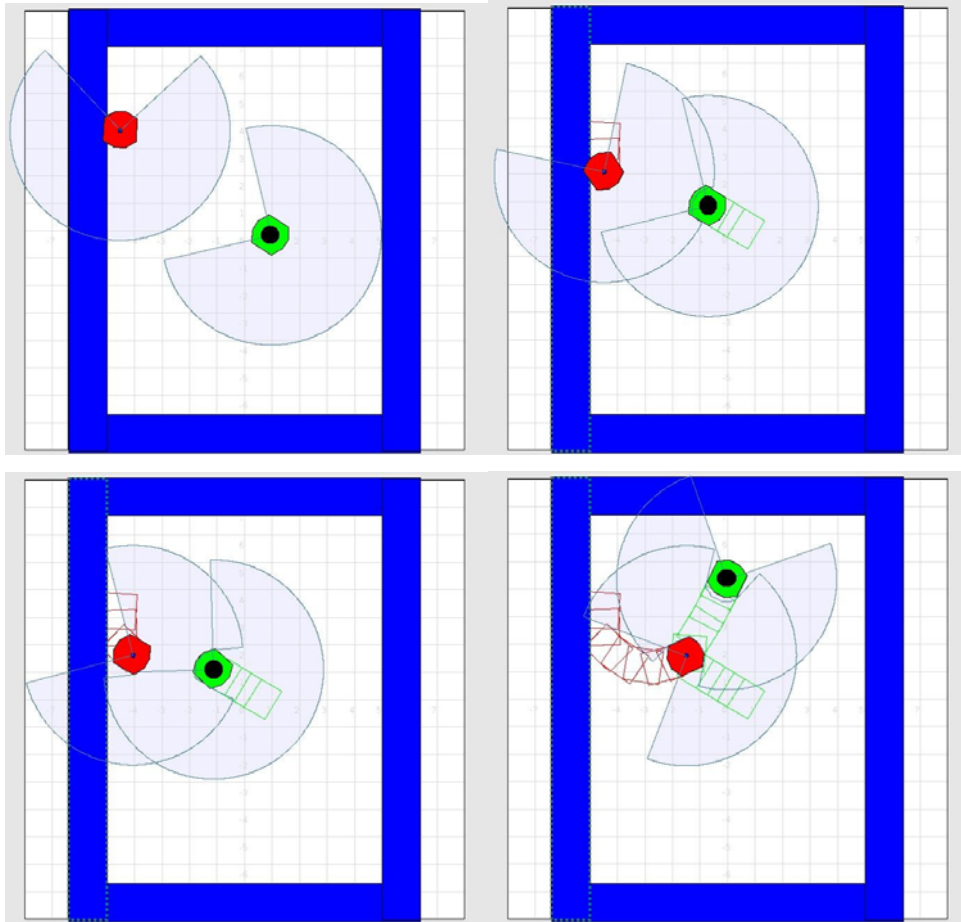


Figura 52 – Trajetória dos robôs presa desviando do robô predador.

Na figura 52, o robô presa está em rota de encontro com o robô predador. Assim que entra no alcance dos sensores do predador, este tem seu esquema motor CAÇAR ativado pelo arbitrador, e o campo potencial atrativo da presa exerce as forças necessárias para que direcione o predador diretamente para a presa. No momento seguinte a presa detecta o predador, o esquema motor FUGIR é habilitado, e o campo potencial repulsivo da presa gera as forças necessárias para realizar o desvio com sucesso.

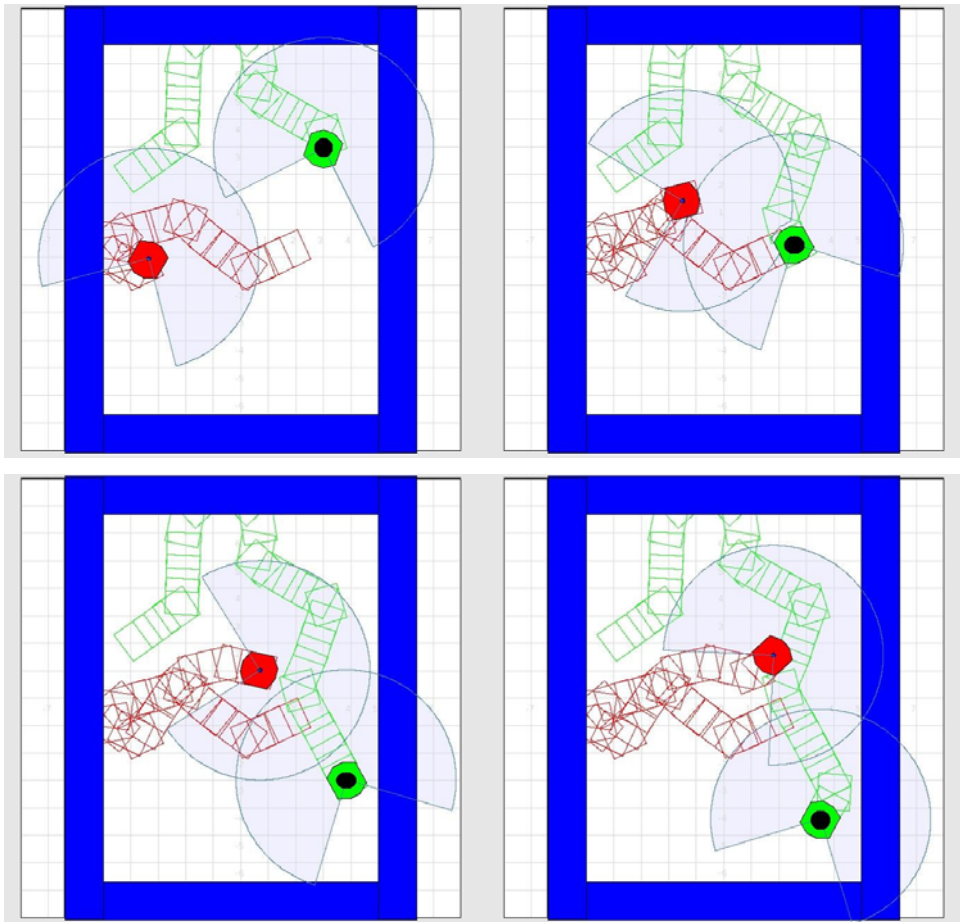


Figura 53 – Trajetória de fuga do robô presa.

Na figura 53, as trajetórias executadas pelos robôs resultam em uma rota na qual os robôs se identificam no mesmo instante, e a partir deste momento ambos têm seu respectivo esquema motor acionado.

A presa executa um leve desvio, enquanto que o robô predador é forçado a girar em um ângulo maior para poder perseguir a presa. Este giro faz com que o predador perca tempo e, devido à maior velocidade de presa, esta consegue sair do alcance dos sensores do predador, executando a fuga com sucesso.

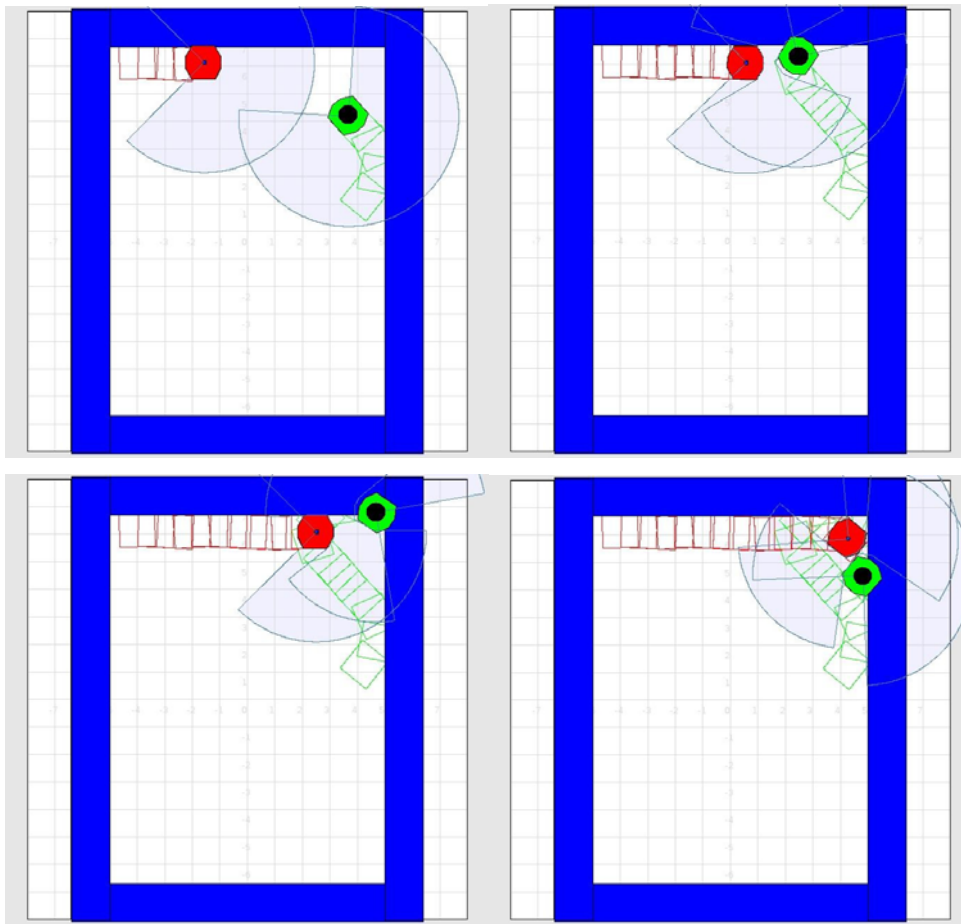


Figura 54 – Robô predador atinge robô presa.

Na figura 54, observa-se um exemplo de quando a presa é alcançada pelo predador. O predador mantém uma trajetória retilínea, pois quando seu esquema perceptivo tende a girá-lo para a esquerda, o esquema perceptivo de desvio corrige sua trajetória. Este fato faz parecer que ele obedece a uma rota constante.

O robô presa está localizado no canto superior direito da arena e com ângulo aberto. Quando detecta o predador com seu sensor esquerdo, ele reage para girar à direita. Esta ação coloca o robô presa em uma situação de risco, pois sobra pouco espaço para uma manobra evasiva sem sair da arena. Conseqüentemente, a perda de tempo gerada pelo desvio da faixa de perigo faz com que o robô predador consiga atingir o robô presa.

As simulações predador/presa realizadas são validadas experimentalmente com a utilização de robôs reais que foram utilizados como modelo para esta simulação. Os experimentos são descritos a seguir.