

Pioneer 3-DX Básico

Prof. Guilherme Sousa Bastos

1	Introdução.....	1
2	Histórico	2
3	Definição de Robô.....	3
3.1	Manipulador mecânico	4
3.1.1	Elos	4
3.1.2	Junta.....	4
3.1.3	Sistema de Transmissão	5
3.2	Atuadores.....	5
3.2.1	Atuadores Hidráulicos e Pneumáticos.....	5
3.2.2	Atuadores Eletromagnéticos.....	6
3.3	Sensores.....	6
3.4	Unidade de Controle.....	7
3.5	Unidade de Potência	7
3.6	Efetuator	7
4	Características Básicas de Robôs	8
4.1	Resolução Espacial	8
4.2	Precisão.....	9
4.3	Repetibilidade.....	9
4.4	Complacência	10
5	Robôs Móveis.....	11
5.1	Evolução dos Robôs Móveis	11
5.2	Robôs com Esteiras	12
5.3	Robôs com Patas.....	12
5.4	Robôs de Limpeza	13
5.4.1	Limpeza de Pisos Domésticos e Industriais	13
5.4.2	Limpeza de Dutos para Petróleo.....	14
5.4.3	Limpeza de Tubulações em Geral	14
5.4.4	Limpeza de Cascos de Navio.....	15
5.5	Robôs Cortador de Grama	15
5.6	Robô Submarino	16
5.7	Cinemática de Robôs Móveis.....	18
6	Robótica Cooperativa	19
7	Robô Móvel Pioneer 3-DX.....	21
7.1	Família Pioneer.....	22
7.1.1	Pioneer 1	22
7.1.2	Pioneer 2	22
7.1.3	Pioneer 3	23
7.2	Controles e Especificações	24
7.2.1	Mesa	25
7.2.2	Botão de Parada do Motor	26
7.2.3	Painel de Controle do Usuário.....	26
7.2.4	Corpo, Nariz e Painel de Acessórios	29
7.2.5	Sonares	30
7.2.6	Motores, Rodas e Encoders	30
7.2.7	Bateria e Alimentação	31
7.3	Acessórios.....	32
7.3.1	Conexão cliente-servidor.....	32

7.3.2	PC Integrado	32
7.3.3	Rede	33
7.4	ARIA	34
7.5	MobileSim	42
7.6	Mapper3Basic	47
7.7	Exemplos de Programas utilizando o ARIA.....	48
7.7.1	Exemplo 1	49
7.7.2	Exemplo 2.....	51

1 INTRODUÇÃO

A robótica é uma ciência da engenharia aplicada que é tida como uma combinação da tecnologia de máquinas operatrizes e ciência da computação. Inclui campos aparentemente tão diversos quanto projeto de máquinas, teoria de controle, microeletrônica, programação de computadores, inteligência artificial, fatores humanos e teoria da produção.

Automação e robótica são duas tecnologias intimamente relacionadas. Num contexto industrial podemos definir a automação como uma tecnologia que se ocupa do uso de sistemas mecânicos, eletrônicos e à base de computadores na operação e controle de processos e da produção. Exemplos dessa tecnologia incluem máquinas de montagem mecanizadas, sistemas de controle de realimentação (aplicados a processos industriais), máquinas operatrizes dotadas de comandos numéricos e robôs. Conseqüentemente, a robótica é uma forma de automação industrial.

O maciço investimento em robôs industriais no processo produtivo observado nas últimas décadas deve-se principalmente às crescentes necessidades impostas pelo mercado em se obter sistemas de produção cada vez mais automatizados e sistemas integrados de manufatura, o robô industrial tornou-se um elemento importante nesse contexto.

Um sistema de produção tem por objetivo agregar valor a produtos. Ou seja, a partir de uma *entrada de materiais* a serem processados – matérias-primas, peças básicas ou conjunto de peças (subgrupos) – o sistema de produção irá executar algum *processo de transformação* sobre esses materiais, resultando em *produtos processados*, com valor comercial mais elevado. Estes podem ser produtos acabados aptos a serem comercializados diretamente no mercado ou, ainda, produtos intermediários, que serão utilizados posteriormente na construção de produtos acabados.

O uso de robôs industriais no chão de fábrica de uma empresa está diretamente associado aos objetivos da produção automatizada a qual visa:

- reduzir custos dos produtos fabricados, através de: diminuição de número de pessoas envolvidas na produção, aumento da quantidade de produtos em um dado período (produtividade), melhor utilização de matéria-prima (redução de perdas, otimização do aproveitamento), economia de energia, etc.;

- melhorar as condições de trabalho do ser humano, por meio da eliminação de atividades perigosas ou insalubres de seu contato direto;
- melhorar a qualidade do produto, através do controle mais racional dos parâmetros de produção;
- realizar atividade impossíveis de serem controladas manual ou intelectualmente como, por exemplo, a montagem de peças de peças em miniatura, a coordenação de movimentos complexos e atividades muito rápidas (deslocamento de materiais).

2 HISTÓRICO

O termo *robô* teve sua origem em uma peça teatral de nome "RUR" (*Rossum's Universal Robot*) de Karel Capek, criada em 1921. Na peça, Capek utiliza a palavra *robotas*, que em tcheco significa "escravo, trabalhador compulsório". A intenção do autor era idealizar uma fábrica de trabalhadores, ou seja, uma fábrica que produzisse *robotas*, tipo andróides, que mais tarde acabavam se rebelando contra o ser humano.

Na década de 40, o escritor Isaac Asimov tornou popular o conceito de robô como uma máquina de aparência humana, não possuidora de sentimentos, cujo comportamento seria definido a partir de programação feita por seres humanos, de forma a cumprir determinadas regras éticas de conduta. O termo *robótica* foi criado por Asimov para designar a ciência que se dedica ao estudo dos robôs e que se fundamenta pela observação de três leis básicas:

1ª.) "Um robô não pode ferir um ser humano ou, permanecendo passivo, deixar um ser humano exposto ao perigo".

2ª.) "O robô deve obedecer às ordens dadas pelos seres humanos, exceto se tais ordens estiverem em contradição com a primeira lei".

3ª.) "Um robô deve proteger sua existência na medida em que essa proteção não estiver em contradição com a primeira e a segunda leis".

4ª.) "Um robô não pode causar mal à humanidade nem permitir que ela própria o faça".

A quarta e última lei foi escrita por Asimov em 1984.

A base tecnológica para os atuais robôs industriais foi desenvolvida a partir de pesquisas iniciadas logo após a Segunda Guerra Mundial, quando foi construído um equipamento denominado *teleoperador mestre-escravo*

(*master-slave*), empregado em atividades de manipulação de materiais radioativos. O sistema era formado de um manipulador mestre, movido diretamente por um operador humano responsável pelas seqüências de movimentos desejados, e um manipulador escravo, capaz de reproduzir os movimentos realizados remotamente pelo mestre. Os vínculos entre os manipuladores mestre e escravo eram realizados através de sistemas de transmissão mecânicos.

O primeiro robô industrial moderno foi desenvolvido por George Devol e Joe Engelberger no final dos anos 50. Engelberger criou a empresa Unimation Inc, que iniciou a comercialização de robôs industriais, sendo esse o motivo pelo qual é chamado o de “pai da robótica”.

A Unimation Inc. instalou o primeiro robô industrial, denominado Unimate, no chão-de-fábrica de uma empresa em 1961. O projeto desse robô resultou da combinação entre os mecanismos articulados e garras usados no teleoperador mestre-escravo e a tecnologia de controle desenvolvida em máquinas operatrizes com comando numérico. Desde então, o constante desenvolvimento tecnológico nos ares de mecânica, eletrônica digital, ciência da computação, materiais e logística da produção contribuiu para o aumento da confiabilidade nos componentes empregados em projetos de robôs e a redução dos custos para a sua implementação em atividades industriais.

3 DEFINIÇÃO DE ROBÔ

A Robotic Industries Association (RIA), define *robô industrial* como um “manipulador multifuncional reprogramável, projetado para movimentar materiais, partes, ferramentas ou peças especiais, através de diversos movimentos programados, para a realização de uma variedade de tarefas”.

Uma definição mais completa é apresentada pela norma ISO10218 como “uma máquina manipuladora, com vários graus de liberdade, controlada automaticamente, reprogramável, multifuncional, que pode ter base fixa ou móvel para utilização em aplicações de automação industrial”.

Um robô industrial é formado pela integração de vários componentes, conforme segue:

3.1 Manipulador mecânico

Refere-se principalmente ao aspecto mecânico e estrutural do robô. Consiste na combinação de elementos estruturais rígidos (corpos, ou elos), conectados entre si através de articulações (juntas), sendo o primeiro corpo denominado base e o último extremidade terminal, onde será vinculado o componente efetuator (garra ou ferramenta).

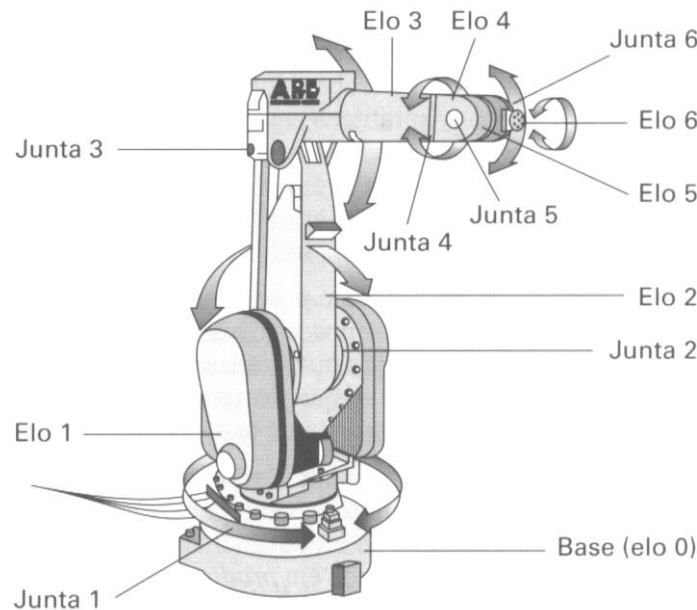


Figura 1 – Robô Industrial de 6 graus de liberdade

3.1.1 Elos

É inevitável que os elos rígidos apresentem algum grau de flexibilidade quando submetido a esforços durante a realização de uma tarefa, sejam estes de natureza estática ou dinâmica. Portanto, nos robôs, a estrutura deve ser projetada para apresentar elevada rigidez aos esforços de flexão e torção. Os materiais mais empregados nas estruturas são alumínio e aço. Mais recentemente têm sido utilizados fibras de carbono e de vidro, materiais termoplásticos e plásticos reforçados.

3.1.2 Junta

Em robótica, geralmente se utilizam dois tipos básicos de junta para compor um par cinemático formado por dois elos adjacentes: junta de rotação ou junta prismática (translação). O uso dessas juntas visa tornar mais simples o processo de montagem e/ou fabricação dos componentes mecânicos que compõem uma junta.

O número de graus de liberdade que um robô apresenta é o número de variáveis independentes de posição que precisam ser especificadas para

se definir a localização de todas as partes do mecanismo, de forma inequívoca. O robô industrial é normalmente uma combinação de elos e juntas em forma de cadeia cinemática aberta. Portanto o número de juntas equivale ao número de graus de liberdade. A Figura 1 ilustra um robô industrial de seis juntas, portanto, seis graus de liberdade.

3.1.3 Sistema de Transmissão

A movimentação de cada corpo ocorre devido à transmissão de potência mecânica (torque/força e velocidade angular/linear) originada num atuador. Os sistemas de transmissão são componentes mecânicos cuja função é transmitir potência mecânica dos atuadores aos elos.

3.2 Atuadores

São componentes que convertem energia elétrica, hidráulica ou pneumática, em potência mecânica. Através dos sistemas de transmissão, a potência mecânica gerada pelos atuadores é enviada aos elos para que se movimentem.

3.2.1 Atuadores Hidráulicos e Pneumáticos

Os atuadores hidráulicos e pneumáticos podem ter a forma de cilindros lineares para gerar os movimentos lineares, ou motores para proporcionar deslocamentos angulares. Ambos são conectados a válvulas direcionais (pré-atuadores), que gerenciam a direção do deslocamento do fluido nos atuadores, a partir de sinais gerados de uma unidade de comando. O custo das válvulas direcionais de alto desempenho ainda é muito elevado.

– Atuadores hidráulicos

Permitem a implementação do controle contínuo e acurado de posicionamento e velocidade, devido a incompressibilidade do fluido (óleo hidráulico), resultando numa elevada rigidez. Outra característica é a elevada relação entre a potência mecânica transmitida pelo atuador e o seu peso, o que possibilita a construção de unidades compactas de alta potência. Uma bomba fornece o óleo para o atuador através das válvulas direcionais.

– Atuadores Pneumáticos

São utilizados em robôs industriais que operam com movimentação de cargas entre posições bem definidas, limitadas por batentes mecânicos, o que caracteriza o movimento ponto-a-ponto. A baixa rigidez desses atuadores devido à compressibilidade do fluido (ar comprimido), permite que sejam obtidas operações suaves. A natureza binária do movimento desses atuadores (posição estendida ou retraída) implica em um controle

simples e de baixo custo. Utiliza-se um compressor para fornecer o ar comprimido ao atuador pneumático, através das válvulas direcionais.

3.2.2 Atuadores Eletromagnéticos

São os atuadores mais utilizados em robôs, principalmente os motores de corrente contínua e de passo. Como vantagens, pode-se citar a grande variedade de fabricantes disponíveis no mercado, o fato de os motores elétricos, quando associados a sensores, poderem ser empregados tanto para o controle de força, quanto da posição do robô, e facilidade de se programar seus movimentos, já que podem ser controlados por sinais elétricos, o que permite a utilização de controladores de movimento.

- Motores de Corrente Contínua (CC)

São compactos e geralmente o valor de torque mantém-se numa faixa constante (com controle PWM) para grandes variações de velocidade, porém necessitam de sensores de posição angular (encoder) e de velocidade (tacômetro) para controle de posicionamento em *malha fechada* (servocontrole). A máxima eficiência mecânica desses motores normalmente ocorre a velocidades elevadas, portanto é comum o uso de redutores de velocidade para se obter a redução de velocidade e, ao elemento movido. Atualmente os fabricantes de robôs utilizam os motores CC sem escovas (brushless), devido à reduzida manutenção, decorrente da diminuição de desgastes e otimização da dissipação térmica entre o rotor e estator.

- Motores de Passo

Podem funcionar em controle de *malha aberta*, em posição e velocidade, e são facilmente interligados a unidades de comando de baixo custo, porém a curva de torque decresce com o aumento da velocidade e, em baixas velocidades, podem gerar vibrações mecânicas. São mais empregados na movimentação de garras.

- Motores de Corrente Alternada (CA)

Utilizados em conjunto com inversores para o controle de velocidade e torque.

3.3 Sensores

Fornecem parâmetros sobre o comportamento do manipulador, geralmente em termos de posição e velocidade dos elos em função do tempo e do modo de interação entre o robô e o ambiente operativo (força, torque, sistema de visão) à unidade de controle. As juntas

utilizadas para vincular os elos de um robô são normalmente acopladas a sensores.

3.4 Unidade de Controle

Responde pelo gerenciamento e monitoração dos parâmetros operacionais requeridos para realizar as tarefas do robô. Os comandos de movimentação enviados aos atuadores são originados de controladores de movimento (computador industrial, CLP, placa controladora de passo) e baseados em informações obtidas através de sensores.

3.5 Unidade de Potência

É responsável pelo fornecimento da potência necessária à movimentação dos atuadores. A bomba hidráulica, o compressor e a fonte elétrica são as unidades de potência associadas aos atuadores hidráulico, pneumático e eletromagnético, respectivamente.

3.6 Efetuador

É o elemento de ligação entre o robô e o meio que o cerca. Pode ser do tipo *garra* ou *ferramenta*. A principal função de uma garra é pegar um determinado objeto, transportá-lo a uma posição pré-estabelecida e após alcançar tal posição, soltá-lo. A ferramenta tem como função realizar uma ação ou trabalho sobre uma peça sem necessariamente manipulá-la.

As ferramentas são em geral, rigidamente fixas às extremidades terminais dos robôs, não possuindo movimentação relativas a estes. A função primordial do robô nesses casos é posicionar e orientar a ferramenta em relação à peça que será trabalhada.

O uso de ferramentas está associado diretamente às tarefas a serem realizadas. Dentre as ferramentas mais tradicionais utilizadas em operações de processamento estão: o porta-eletrodo, a pistola de aspersão (para pó, jateamento de superfícies, etc.), a pistola de pintura, as tochas para soldagem TIG e MIG/MAG, o dispositivo para soldagem/corte a plasma, o conjunto de pinças para soldagem por pontos, o dispositivo para soldagem/corte a laser, o porta-rebolo, o maçarico para corte oxiacetilênico, a pistola para limpeza por jato de água, a pistola para corte por jato de água, etc.

4 CARACTERÍSTICAS BÁSICAS DE ROBÔS

Os robôs são dotados de diversas características básicas, as quais são de extrema importância na seleção do robô em determinada aplicação.

4.1 Resolução Espacial

A resolução espacial de um robô é o menor incremento de movimento em que o robô pode dividir seu volume de trabalho. A resolução espacial depende de dois fatores: a resolução de controle do sistema e as imprecisões mecânicas do robô. O mais fácil é conceituar esses fatores em termos de um robô com um grau de liberdade.

A resolução de controle é determinada pelo sistema de controle de posição do robô e seu sistema de medição de realimentação. Faz parte da capacidade do controlador dividir o deslocamento total de uma junta particular em incrementos individuais, que podem ser endereçados no controlados. Os incrementos são às vezes designados como "ponto endereçável". A capacidade de dividir o deslocamento total da junta em incrementos depende da capacidade de armazenamento de bits na memória de controle. O número de incrementos identificáveis (pontos endereçáveis) para um eixo particular é dado por 2^n .

Um robô com vários graus de liberdade teria uma resolução de controle para cada junta de movimento. Para se obter a resolução de controle para todo o robô, as resoluções componentes teriam de ser somadas vetorialmente. A resolução de controle total dependeria dos movimentos do punho, bem como dos movimentos do braço e do corpo. Já que algumas juntas podem ser rotativas, enquanto outras deslizantes, a resolução de controle do robô pode ser uma tarefa complicada de se determinar.

Imprecisões mecânicas nos elos e componentes de juntas do robô e seu sistema de medição de realimentação (se for um robô servocontrolado) constituem outro fator que contribui para a resolução espacial. Imprecisões mecânicas resultam de flexão elástica nos elementos estruturais, folga nas engrenagens, estiramento de cordões de polias, vazamentos de fluidos hidráulicos e outras falhas no sistema mecânico. Essas imperfeições tendem a piorar no caso de robôs de maior porte, simplesmente porque os erros são ampliados pelos componentes de maior porte. As imprecisões seriam também influenciadas por fatores tais como a carga que está sendo manipulada, a velocidade à qual o braço está se movendo, a condição de manutenção do robô e outros fatores similares.

A resolução espacial do robô é a resolução de controle degradada por essas imprecisões mecânicas. A resolução espacial pode ser melhorada aumentando-se a capacidade de bits da memória de controle. Todavia, é

atingido um ponto em que se acrescenta pouco benefício aumentar ainda mais a capacidade de bits, porque as imprecisões mecânicas do sistema se tornam fator dominante na resolução espacial.

4.2 Precisão

A precisão refere-se à capacidade de um robô de posicionar a extremidade de seu punho num ponto meta desejado dentro do volume de trabalho e pode ser definida em termos de resolução espacial, porque a capacidade de atingir um dado ponto-meta depende de quão proximamente o robô pode definir os incrementos de controle para cada um dos movimentos de suas juntas. No pior caso, o ponto desejado se situaria entre dois incrementos de controle adjacentes. Ignorando no momento as imprecisões mecânicas que reduziriam a precisão do robô, poderíamos inicialmente definir precisão nessa hipótese do pior caso como a metade da resolução de controle. De fato, as imprecisões mecânicas afetariam a capacidade de alcançar a posição meta. Conseqüentemente, definimos a precisão do robô como metade de sua resolução espacial.

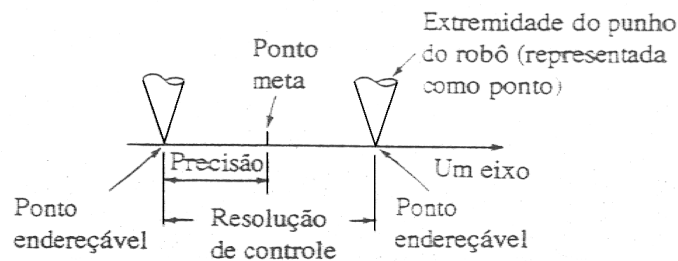


Figura 2 – Precisão e resolução de controle quando as informações são desprezadas

Um terceiro fator que influencia a precisão é a carga que está sendo carregada pelo robô. Cargas mais pesadas causam maior deflexão dos elos mecânicos do robô, resultando em menor precisão.

4.3 Repetibilidade

Repetibilidade diz respeito à capacidade do robô posicionar seu punho ou um órgão terminal ligado ao seu punho num ponto no espaço previamente indicado. O verdadeiro ponto programado provavelmente será diferente do ponto-meta, devido a limitações de controle. A repetibilidade refere-se à capacidade do robô de retornar ao ponto programado quando comandado.

O ponto-meta é designado pela letra T. Durante o procedimento de instrução, o robô é comandado a mover-se para o ponto T, mas, devido às limitações em sua precisão, a posição programada torna-se o ponto P. A distância entre os pontos T e P é uma manifestação de precisão do robô, neste caso. Subseqüentemente, o robô é instruído a retornar ao

ponto programado P; todavia, não retorna à posição exata. Em vez disso, vai para a posição R. A diferença entre P e R é o resultado de limitações à repetibilidade do robô. O robô nem sempre retorna à mesma posição R em subseqüentes repetições do ciclo de movimento. Ao contrário, forma um agrupamento de pontos de ambos os lados da posição P.

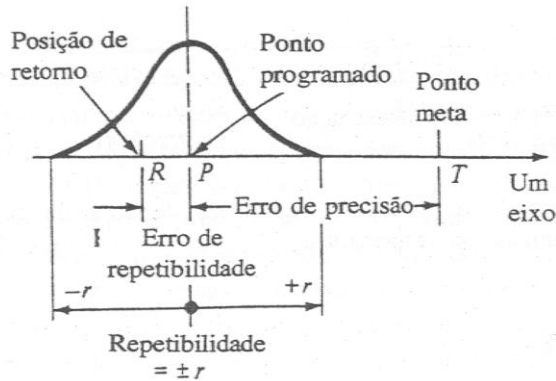


Figura 3 – Repetibilidade e Precisão

4.4 Complacência

A complacência do robô manipulador refere-se ao *deslocamento* da extremidade do punho em resposta a uma força ou torque exercido contra ele. Uma alta complacência significa que o punho é deslocado em grande medida por uma força relativamente pequena. O termo “elástico” é usado às vezes para descrever um robô com alta complacência. Uma baixa complacência significa que o manipulador é relativamente rígido e não é deslocado em medida significativa.

A complacência do manipulador é uma característica direcional. Isto é, a complacência do braço do robô será maior em certas direções, devido à construção mecânica do braço.

A complacência é importante porque reduz a precisão de movimento do robô sob carga. Se o robô estiver manipulando uma carga pesada, o peso da carga fará com que seu braço sofra deflexão. Se o robô estiver pressionando uma ferramenta contra uma peça de trabalho, a força de reação da peça poderá causar flexão do manipulador. Se o robô foi programado em condições sem carga para posicionar seu órgão terminal e a precisão de posição for importante na aplicação, o desempenho do robô será reduzido devido à complacência quando operar em condições carregadas.

5 ROBÔS MÓVEIS

Desde a década de 60, quando a empresa Unimation instalou o primeiro robô manipulador industrial, já se contabilizam mais de 800 mil robôs em todo o mundo. E se prevê um rápido crescimento desse número, principalmente devido ao custo da mão-de-obra, à necessidade de aumento de produtividade, melhoria da qualidade final, realização de tarefas perigosas, que o homem não pode realizar, e à queda do custo dos robôs.

Um outro tipo de robô, o chamado robô móvel, também vem sendo amplamente pesquisado, e utilizado. Esses robôs são empregados numa enorme variedade de tarefas, do transporte de peças em uma indústria, até para substituir o homem na exploração de locais perigosos, como águas profundas, áreas radioativas, crateras de vulcões, ambientes espaciais e mesmo outros planetas. Diferentemente dos robôs manipuladores, cuja base fixa serve de referencia tanto para os movimentos do robô quanto para a área de trabalho, um robô móvel autônomo é livre para mover-se em todas as direções, portanto sua controlabilidade exige um maior grau de interação com o ambiente para que seja continuamente redefinido o referencial que irá permitir a realização de uma dada tarefa. As três formas mais comuns de movimentos são por rodas, esteiras e patas, embora existam robôs que saltam, escalam, voam e até se deslocam por propulsão no fundo do mar.

5.1 Evolução dos Robôs Móveis

Na década de 70 começaram os primeiros desenvolvimentos de robôs móveis: surgiram os robôs não autônomos, que se movem sobre trajetórias fixas, restritas de alguma forma (por exemplo, trilhos), ou seguindo marcas contínuas ou espaçadas, pintadas no solo. Esses robôs são normalmente classificados como veículos autoguiados, ou AGV, de caminho fixo, e realizam operações de transporte automatizado, com diferentes graus de complexidade, em fábricas e armazéns.

Na década de 80 apareceram os primeiros *robôs autônomos*, ou AGV de caminho indefinido, que são restritos a um espaço de trabalho fixo. Os robôs autônomos são mais versáteis, mas requerem um sistema de sensoriamento mais sofisticado, exigindo computadores de bordo poderosos, capazes de atender às exigências dos sistemas de sensoriamento, navegação e controle. Esse tipo de robô é atualmente objeto de amplo estudo: pretende-se chegar a sistemas capazes de operar autonomamente nas mais diversas situações, desde ambientes fechados e estruturados, até locais abertos, desconhecidos, acidentados e dinâmicos. Para isso, seria necessário reproduzir praticamente toda capacidade humana de percepção sensorial, habilidade motora e discernimento abstrato, obtendo sistemas capazes de operar sem a

presença humana, ou com sua mínima intervenção. O exemplo mais recente de aplicações desse tipo de robô é o *Sojourner* que a Nasa enviou a Marte em 1997.

Entre os mecanismos de movimentação utilizados em robôs, as rodas são os mais populares. Isso ocorre por razões práticas, já que robôs com rodas são mais simples e fáceis de construir. Os robôs com patas ou com esteiras normalmente requerem hardware mais complexo e têm maior peso que os correspondentes com rodas, projetados para a mesma carga útil. A principal desvantagem dos robôs móveis com rodas é seu baixo desempenho em terrenos acidentados, já que, como regra geral, um veículo com rodas tem problemas quando a altura do obstáculo se aproxima do raio de suas rodas.

5.2 Robôs com Esteiras

Em um ambiente natural, os robôs com esteiras são mais interessantes porque elas permitem ao robô transpor obstáculos relativamente mais altos. Assim, esses robôs são menos suscetíveis que os de rodas a terrenos acidentados, como terra solta e pedras. A principal desvantagem das esteiras é sua ineficiência, pois o atrito nelas dissipa potência, desperdiçando energia sempre que o robô se movimenta.



Figura 4 – Robô Hazbot III

5.3 Robôs com Patas

Os robôs com patas, ou caminchantes, podem atuar em meios submarinos, espaciais e terrestres. Também podem superar a maioria dos problemas de locomoção em terrenos acidentados mais do que qualquer robô com rodas ou esteiras. Entretanto um dos fatores que dificultam o desenvolvimento dos robôs caminchantes é a complexidade de

coordenação dos movimentos de suas patas. Isso se deve ao modo de caminhar, à estabilidade do corpo sob a ação do movimento, e à arquitetura das patas. O modo de caminhar está associado à coordenação harmônica das patas no movimento de locomoção; a estabilidade do corpo está associada à distribuição de massas em relação às patas, quando em movimento; e a arquitetura mecânica se deve principalmente ao número de graus de liberdade, tipo de configuração, e formação mecânica das juntas.

Embora exista um grande interesse no desenvolvimento de sistemas práticos, o robô caminhante ainda tem que superar alguns desafios. Muitos desses desafios se devem ao grande número de graus de liberdade requerido por esse tipo de robô. Como cada para de ter pelo menos dois motores, o custo de um robô com patas é relativamente mais alto que o dos robôs com rodas ou esteiras. Além disso, o mecanismo de caminhar é mais complexo e, portanto, mais propenso a falhas, sendo que o algoritmo de controle fica comprometido, dado que existem mais movimentos a coordenar.

É importante ressaltar o enorme interesse despertado ultimamente nesse tipo de robô por pesquisadores da área. Isso se deve principalmente pela capacidade que esse tipo de robô tem em se locomover em terrenos acidentados, subir e descer escadas, o que os habilita a aplicações em situações de emergência, onde a intervenção humana é impossível ou arriscada. O interesse da Nasa em robôs de patas está na utilização em tarefas de busca entre escombros de sobreviventes de um desastre ou terremoto; para a exploração de áreas incendiadas com radioatividade, sob bombardeios e com contaminação química ou biológica.

5.4 Robôs de Limpeza

5.4.1 Limpeza de Pisos Domésticos e Industriais

Existem trabalhos pioneiros no uso de robôs de limpeza de piso, para ambientes domésticos, tal como o robô móvel com rodas *RoboScrub*, criado em 1991, desenvolvido pelas empresas Denning Mobile Robotics e Windsor Industries. Entretanto esse pioneirismo não garantiu um sucesso comercial, por várias razões. Primeiro, o robô às vezes se chocava com os móveis, danificando-os. Segundo, o custo do robô era bastante alto, bem acima do salário de um empregado de limpeza. Por estas razões, o *RobScrub* foi posteriormente destinado à limpeza de grandes salões, de forma a justificar seu alto custo, embora não se garantisse a limpeza completa do ambiente (certos locais fora do alcance do robô).



Figura 5 – Robô de Limpeza Robosoft

5.4.2 Limpeza de Dutos para Petróleo

Na indústria do petróleo, verifica-se nos dutos para o escoamento do óleo e seus derivados, um grande número de entupimentos causados por formação de hidratos e de parafina, além de outros problemas como amassados nas paredes e vazamentos.

O grupo de robótica do CENPES/Petrobrás desenvolveu um novo conceito de tele-robô para atividades de inspeção e intervenção capaz de se deslocar pelo interior de dutos, independente da direção do fluido, denominado GIRINO (Gabarito Interno Robotizado de Incidência Normal a Oleodutos). Seu movimento é baseado nos anfíbios, em especial os girinos, sendo acionado por sistema hidráulico.

O protótipo pode dispor de acessórios de inspeção de injeção química para desobstrução e dispositivos porta-ferramentas voltados a reparos estruturais feitos de dentro para fora nos dutos.



Figura 6 – Robô Rubio V II (Tecnomecanica)

5.4.3 Limpeza de Tubulações em Geral

Robôs comerciais para limpeza de tubulações de esgoto vêm sendo utilizados em diversas cidades. Nessa aplicação, geralmente um operador humano controla remotamente os movimentos do robô enquanto recebe as imagens internas da tubulação. Existem pesquisas para

desenvolvimento de robôs que realizem essa limpeza de forma autônoma. Outra aplicação desses robôs é na limpeza de dutos de ar condicionado e de usinas nucleares.



Figura 7 – Robô limpador de dutos de esgoto

5.4.4 Limpeza de Cascos de Navio

Há robôs que podem ser utilizados para a limpeza de cascos de navios. É o caso, por exemplo, do produto da empresa Barnes and Reineke, um robô manipulador de grandes dimensões para efetuar esse tipo de limpeza.

5.5 Robôs Cortador de Grama

Uma outra aplicação de robôs destinados a ambientes domésticos é para o corte de grama, como o robô *Lawn Nibbler*, que corta a grama numa área definida, cujo perímetro é demarcado por emissores de radiofrequência. Enquanto corta a grama, o robô desvia-se de obstáculos (árvores, brinquedos, animais, crianças, etc.).



Figura 8 – Robô RL500

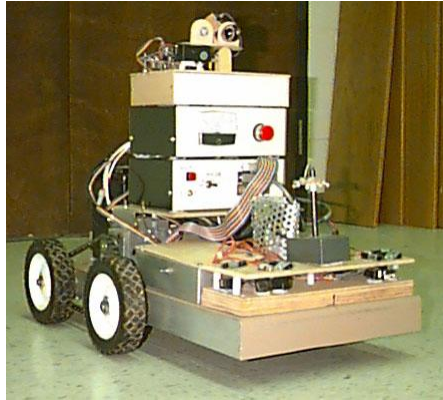


Figura 9 – Robô Lawn Nibbler

5.6 Robô Submarino

Existem basicamente dois tipos de robôs submarinos: o ROV (Remotely Controlled Vehicle), que pode ser controlado remotamente, e o AUV (Autonomous Underwater Vehicle), o qual opera autonomamente. Estes robôs têm sido aprimorados para operar em grandes profundidades e áreas de risco onde os mergulhadores não podem operar. Os ROVs podem realizar diversas tarefas, utilizando para isso braços mecânicos e ferramentas especializadas. Entre as aplicações realizadas por esses robôs podem-se citar:

- investigação visual das partes submersas de navios, pontes e plataformas;
- monitoramento de colônias de peixes;
- controle de poluição subaquática;
- controle de devastação da vida marinha;
- localização de depósitos de combustíveis nucleares;
- execução de operações tecnológicas em plataformas marítimas;
- inspeção visual de estruturas subaquáticas de oleodutos e gasodutos;
- inspeção de barragens;
- explorações marítimas para fins de pesquisa.

Tipicamente, os ROVs operam conectados a uma embarcação de superfície através de um cabo umbilical. Por esse cabo, o ROV recebe a energia necessária para sua operação e se comunica com a unidade de comando enviando sinais de vídeo e recebendo sinais de controle. Entretanto, o cabo limita a velocidade e o alcance do robô, além da possibilidade de enredar-se tanto no robô quanto em algum obstáculo. Os ROVs normalmente são movimentados por motores elétricos que acionam hélices ou turbinas.



Figura 10 – ROV Scorpio

Os AUVs, de tecnologia mais recente, operam de forma autônoma, dispensando os cabos umbilicais. Entretanto essa autonomia é limitada pela duração de suas baterias. Isso pode ser contornado, mantendo-se um navio de recarga de bateria próximo ao ponto onde o AUV realiza suas operações.



Figura 11 – AUV Odyssey IIc

5.7 Cinemática de Robôs Móveis

Cinemática é o estudo de como sistemas mecânicos se comportam. Na robótica móvel, é necessário que se entenda o comportamento mecânico do robô para que se possa projetar robôs móveis apropriados para tarefas e também a criação de softwares de controle.

O estudo da cinemática de robôs móveis se dá com a análise de suas configurações (Figura 12), para que se possa determinar o Pose (posição e orientação) do robô (Figura 13) e realizar deslocamento em trajetórias pré-definidas (Figura 14).

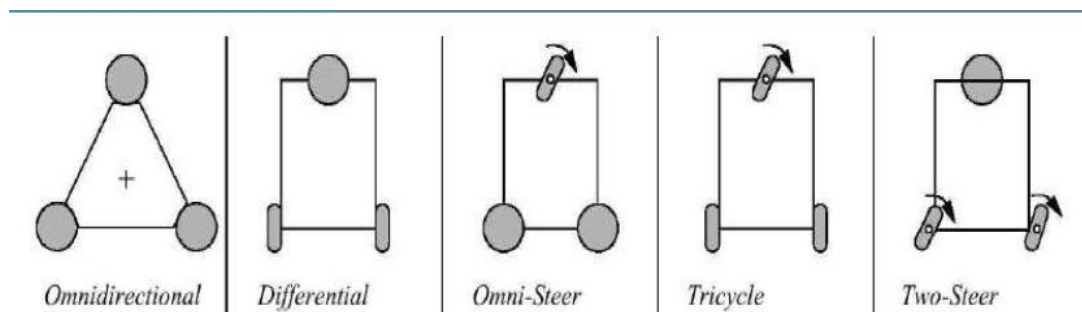


Figura 12 – Configuração de robôs móveis com tração em duas rodas

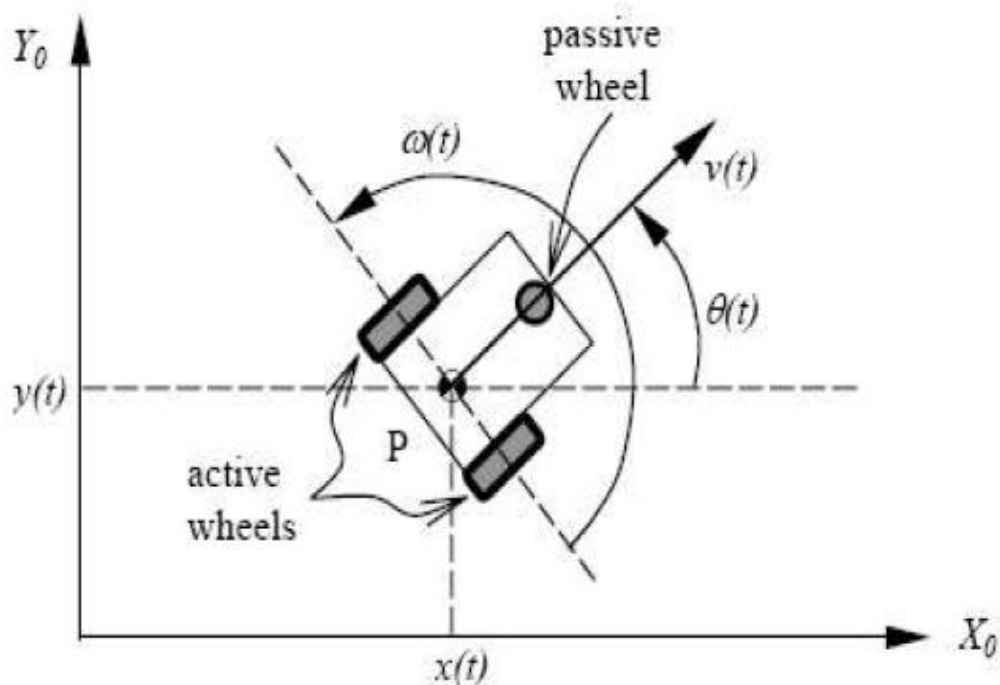


Figura 13 – Posição e orientação (Pose) de robô móvel

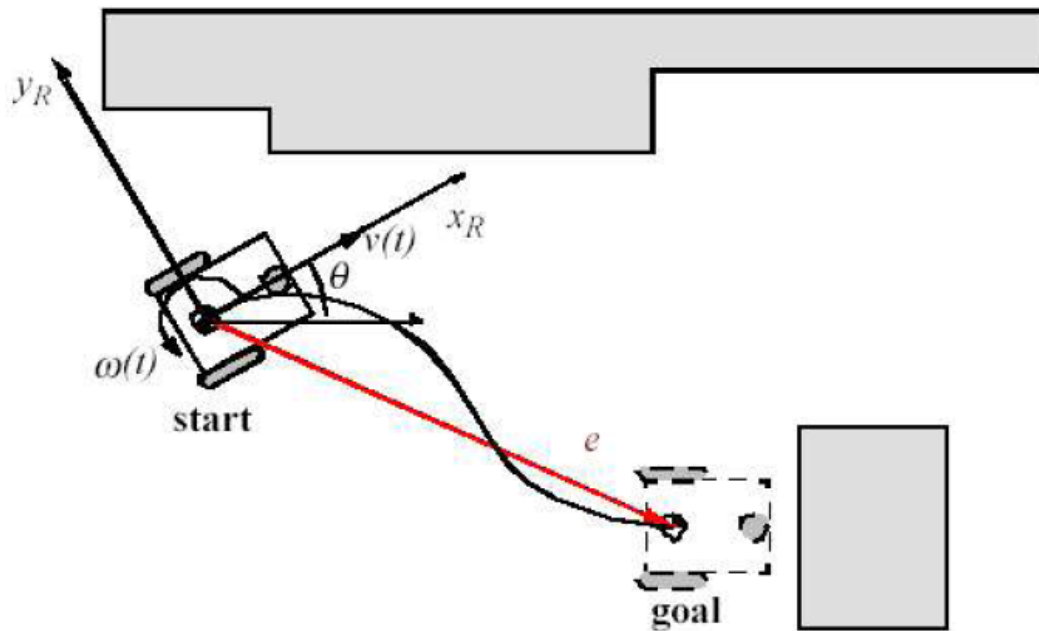


Figura 14 – Controle de trajetória de robô móvel

6 ROBÓTICA COOPERATIVA

A Robótica Cooperativa é aplicada quando se têm vários robôs (móveis ou de base fixa) trabalhando em conjunto e de forma autônoma. As figuras a seguir mostram alguns exemplos de Robótica Cooperativa.

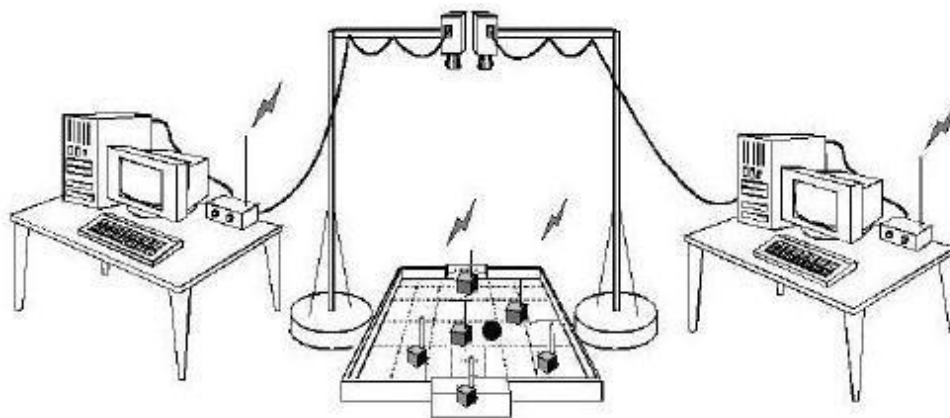


Figura 15 – Futebol de Robôs

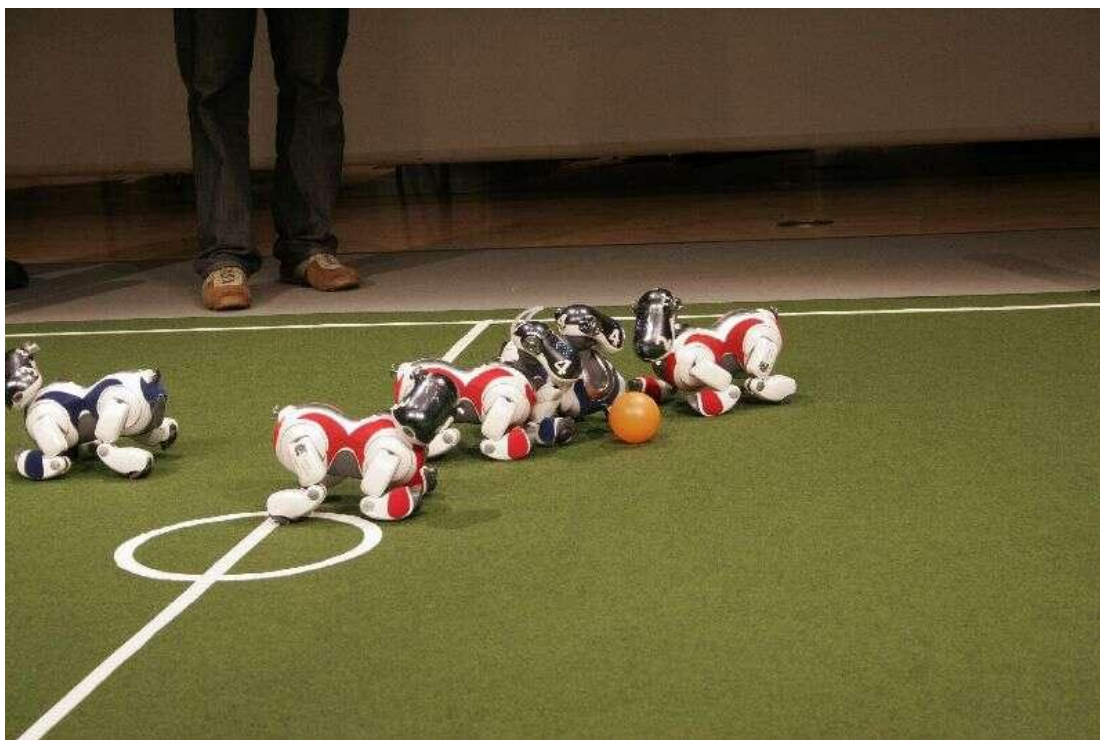


Figura 16 – Futebol de robôs - AIBO



Figura 17 – Resgate com robôs móveis

7 ROBÔ MÓVEL PIONEER 3-DX

Pioneer é uma família de robôs, com tração em duas ou quatro rodas, incluindo o *Pioneer 1* e *Pioneer AT*, *Pioneer 2-DX*, *-DXe*, *-DXf*, *-CE*, *-AT*, o *Pioneer 2-DX8/Dx8 Plus* e *-AT8/AT8 Plus*, e os mais novos *Pioneer 3-DX* e *-AT*. Estas pequenas plataformas de pesquisa e desenvolvimento compartilham uma arquitetura comum e software com todas as outras plataformas MOBILEROBOTS, incluindo os robôs móveis *AmigoBot*, *PeopleBot V1*, *Performance PeopleBot* e *PowerBot*.

A plataforma MOBILEROBOTS define os padrões para robôs móveis inteligentes por conter os componentes básicos para sensoramento e navegação em um ambiente de mundo real.



Figura 18 – Plataforma MOBILEROBOTS

Todas as plataformas MOBILEROBOTS vêm completas com um corpo forte de alumínio, sistema de tração balanceado, motores DC reversíveis, eletrônica de potência e controle, encoders de alta resolução e baterias, tudo controlado por um microcontrolador *onboard* e um software servidor.

7.1 Família Pioneer

7.1.1 Pioneer 1

Comercialmente introduzido em meados de 1995, o *Pioneer 1* (Figura 17) foi a plataforma original do MOBILEROBOT. Ele foi desenvolvido para uso *indoor* em superfícies planas e rígidas. Seu microcontrolador é o 68HC11 e utiliza o software PSOS (*Pioneer Server Operating System*). O *Pioneer 1* vem também, no modo padrão, com sete sonares (dois aos lados, e cinco a frente) e encoders de roda integrados. O seu baixo custo e alto desempenho causaram uma explosão no número de pesquisadores e desenvolvedores que passou a ter acesso a uma plataforma robótica móvel inteligente real.



Figura 19 – *Pioneer AT* e *Pioneer 1*

O *Pioneer AT* (Figura 19), que foi introduzido no mercado em meados de 1997 para operação em ambientes *indoor* e *outdoor*, conta com tração nas quatro rodas e é funcionalmente idêntico ao *Pioneer 1* (incluindo a mesma programação).

7.1.2 Pioneer 2

A próxima geração do *Pioneer*, incluindo o *Pioneer 2-DX*, *-CE*, e *-AT* foram introduzidos entre o final de 1998 e meados de 1999, realizando uma melhora nos modelos *Pioneer 1* enquanto mantinha as suas várias vantagens. Com relação às aplicações de software, o *Pioneer 2* trabalha de forma idêntica aos modelos *Pioneer 1*, mas oferecia várias opções de expansão, incluindo um PC cliente *onboard* no robô.

Os modelos *Pioneer 2* usavam um microcontrolador Siemens 88C166 20MHz, com alimentação independente dos motores e sonares microcontrolados para uma operação versátil no ambiente. Buscando uma arquitetura mais holonômica, rodas maiores e motores mais fortes foram introduzidos para um desempenho *indoor* melhor.

O robô *PeopleBot* (Figura 20), baseado no modelo *Pioneer 2*, tem motores mais fortes e capacidades integradas de interação com humanos,

incluindo extensão do pedestal, voz integrada e síntese e reconhecimento de sons, o que é ideal para estudos de interação com o homem.



Figura 20 – PeopleBot

7.1.3 Pioneer 3

O *Pioneer 3* foi lançado em meados de 2003 e utiliza o microcontrolador Hitachi H8S, com o novo software de controle de sistema ARCOS (*Advanced Robot Control & Operations Software*) e capacidade de expansões de I/O. Em meados de 2004, seu microcontrolador foi substituído pelo revolucionário Renesas SH2. Apresentavam também novos motores (mais potentes) e sistemas de potência para melhor controle de navegação e maior capacidade de carga. A Figura 21 mostra o robô *Pioneer 3-DX*, o qual será utilizado no curso.

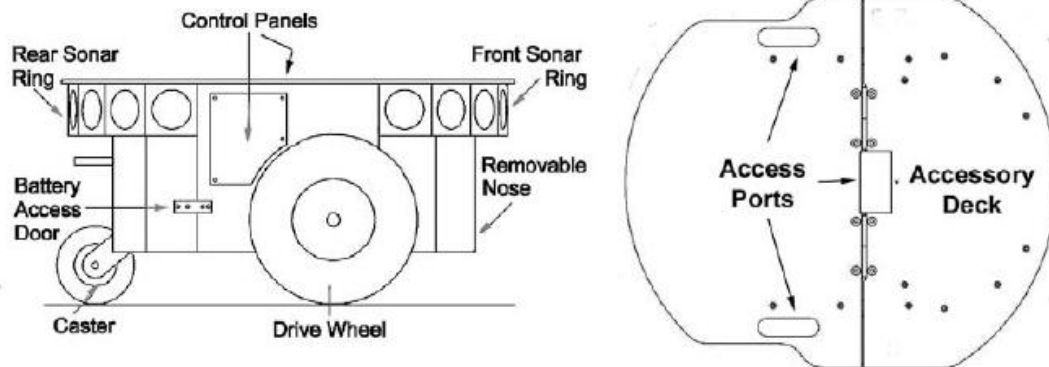


Figura 21 – Pioneer 3-DX

7.2 Controles e Especificações

Os robôs *Pioneer* são menores do que a maioria, mas apresentam uma impressionante lista de capacidades inteligentes de robôs móveis comparada a robôs maiores e mais caros. O *Pioneer 3-DX* com PC *onboard* é um robô móvel inteligente totalmente autônomo. Diferentemente de outros robôs comerciais disponíveis, o tamanho modesto do *Pioneer* permite sua navegação em espaços pequenos e restritos, como salas de aulas, laboratórios e pequenos escritórios. Com o poderoso servidor ARCOS e o avançado software cliente MOBILEROBOTS, o robô *Pioneer 3* é completamente capaz de mapear o seu ambiente, encontrando o caminho para “casa” e executando outras tarefas sofisticadas de planejamento de rota (requer localizador laser e o software ANRL – *Advanced Robotics Navigation and Localization*).

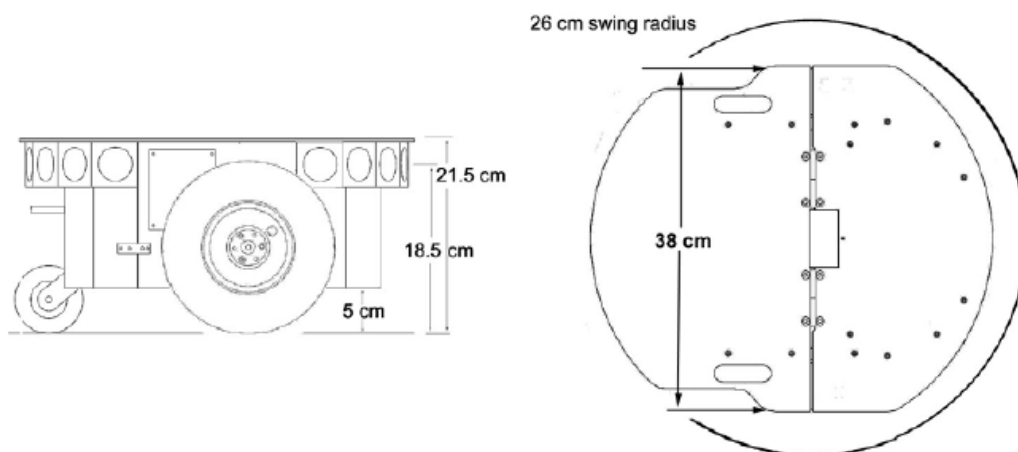


Figura 22 – Dimensões físicas do *Pioneer 3-DX*

Pesando apenas 9 kg (com uma bateria), os robôs móveis Pioneer 3-DX podem ser classificados como “peso leve”, mas seus corpos fortes de alumínio e construções sólidas os fazem virtualmente indestrutíveis. Estas características também permitem que eles carreguem extraordinárias cargas. O Pioneer 3-DX pode carregar até 23 kg extras e o 3-AT até 35 kg extras!

Os robôs Pioneer são compostos pelas partes principais:

- Mesa
- Botão de Parada de Motor
- Painel de Controle do Usuário
- Corpo, Nariz e Painel de Acessórios
- Anel de Sonares
- Motores, Rodas e Encoders
- Baterias e Alimentação

7.2.1 Mesa

A mesa do robô é simplesmente o seu topo composto por uma superfície plana para a montagem de projetos e acessórios, tais como a câmera robótica PTZ e um localizador laser. As Figura 23 e Figura 24 mostram acessórios montados sobre a mesa.



Figura 23 – Robô *Pioneer* com câmera PTZ, localizador laser e caixas de som



Figura 24 – Robô *Pioneer* com manipulador articulado, localizador laser, localizador GPS e câmera PTZ

Na montagem de acessórios, deve-se tentar centralizar a carga sobre as rodas de tração. Caso se tenha um acessório pesado na extremidade da mesa, deve-se contrabalancear o peso com um objeto pesado na outra extremidade. O uso de todas as baterias ajuda a balancear também o robô.

7.2.2 Botão de Parada do Motor

Todos *Pioneer 3-AT* e, sob requisição, alguns *Pioneer 3-DX* têm um Botão de Parada na parte traseira da mesa. Pressionando este botão tem-se o imediato desligamento da alimentação dos motores do robô. Deve-se considerar este botão como um sistema de segurança contra possíveis problemas de programação ou de hardware.

7.2.3 Painel de Controle do Usuário

O Painel de Controle do Usuário (Figura 25) é onde se tem acesso ao ARCOS-microcontrolador (Figura 26). Localizado no painel lateral do *DX*, ele consiste de botões de controle, indicadores e uma porta serial RS-232 compatível.

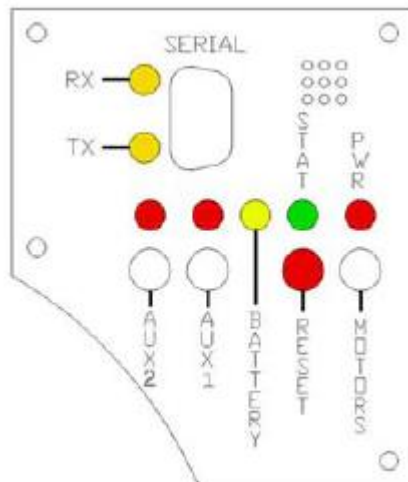


Figura 25 – Painel de Controle do Usuário

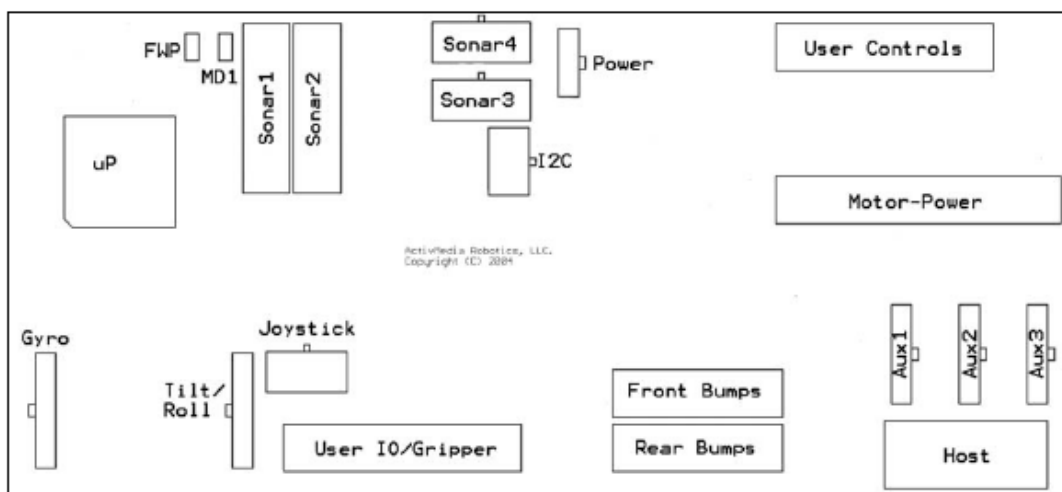


Figura 26 – Microcontrolador SH2

- Indicadores de Força e Status

O led vermelho PWR está aceso quando a alimentação principal é aplicada ao robô. O led verde STAT depende do modo de operação e outras condições. Ele pisca devagar quando o microcontrolador está esperando uma conexão com um cliente e pisca rapidamente quando está no modo *joydrive* ou quando está conectado a um cliente e os motores estão prontos. Ele também pisca moderadamente rápido quando o microcontrolador está no modo de manutenção.

- Buzzer

Um buzzer (audível através dos furos logo acima dos leds STAT e PWR) informa por sinais audíveis o estado do robô, tal como um início com

sucesso do microcontrolador e uma conexão de um cliente. Os sons emitidos pelo buzzer também podem ser reprogramados via um cliente ARCOS para tocar áudio do tipo MIDI.

- Conector SERIAL

O conector SERIAL, com leds de recepção (RX) e transmissão (TX) de dados, permite interação com o microcontrolador ARCOS de um computador *offboard* para um controle cliente-servidor e para manutenção do software do microcontrolador. A porta é compartilhada internamente pela porta serial HOST, onde é conectado o computador *onboard* ou um dispositivo Ethernet-Serial. Tanto o conector SERIAL quanto o HOST podem ser utilizados para controle cliente-servidor e modo de manutenção com o microcontrolador.

- Botões de Alimentação

Os botões AUX1 e AUX2 são *pushbuttons* que energizam ou desenergizam a alimentação de 5 e 12 VDC dos conectores na placa de controle dos motores onde se pode conectar a alimentação de vários acessórios. Por exemplo, 12 VDC para uma câmera PTZ é tipicamente acionada via botão AUX1. Os respectivos leds vermelhos indicam que a alimentação está ligada.

- RESET e MOTORS

O botão vermelho RESET age para resetar o microcontrolador, desabilitando quaisquer conexões ativas ou dispositivos conectados, incluindo os motores.

A ação do botão branco MOTORS depende do estado do microcontrolador. Quando conectado a um cliente, o seu pressionamento habilita e desabilita os motores manualmente.

Para colocar o ARCOS em modo de manutenção, deve-se pressionar e manter o botão MOTORS, pressionar e soltar o botão RESET, e depois soltar MOTORS. Note que esta operação manual era necessária para entrar em modo de manutenção com os microcontroladores anteriores, não é necessária com o ARCOS. No modo de manutenção pode-se reprogramar a memória FLASH do microcontrolador SH2 e também realizar *update* e *upgrade* do ARCOS. Pode-se também ajustar parâmetros que o ARCOS utiliza como valores *default* no início ou reset.

7.2.4 Corpo, Nariz e Pannel de Acessórios

- Corpo

O corpo do *Pioneer* acomoda as baterias, motores de tração, eletrônica e outros componentes comuns, incluindo o anel de sonares. O corpo tem espaço suficiente, com conectores de força e sinal, para suportar uma variedade de acessórios robóticos, incluindo sistema de vigilância A/V sem fio, Ethernet wireless, computador onboard, localizador laser, etc. Uma porta traseira fornece fácil acesso às baterias, onde se pode fazer troca a quente de qualquer uma das três baterias.

- Nariz

O nariz é onde está colocado o PC *onboard*. O acesso é feito através da remoção de alguns parafusos e abertura do nariz, dando acesso a placas de acessórios e o HD do PC *onboard*. Também se tem acesso ao ajuste de ganho do anel de sonares. O nariz também é um local ideal para se colocar sensores e acessórios, como se pode visualizar a câmera PTZ na Figura 24 e garra na Figura 27



Figura 27 – *Pioneer 3-AT* com garra

- Pannel de Acessórios

Todos os *DX* vem um painel removível no lado direito, através do qual pode-se instalar conectores de acessórios e controles. Um painel lateral especial vem com a opção PC *onboard*, que contém conectores para um monitor, teclado, mouse e 10Base-T Ethernet, e botões de força e reset do computador *onboard*.

Todos os modelos vêm com uma porta de acesso próxima ao centro da mesa, através da qual se tem acesso a componentes internos.

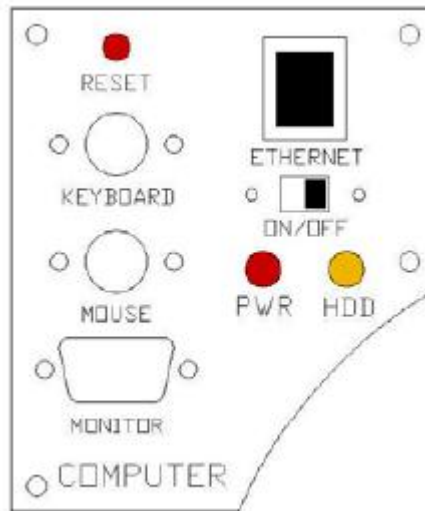
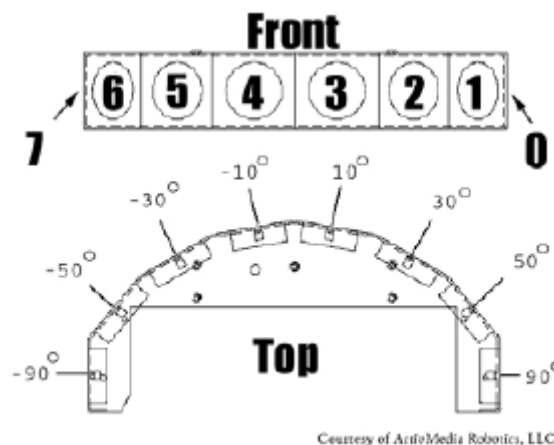


Figura 28 – Painel Computador *onboard*

7.2.5 Sonares

Os robôs baseados no ARCOS suportam até quatro conjunto de sonares, cada um com oito transdutores que permitem a detecção de objetos e informação para se evitar colisões, reconhecimento de padrões, localização e navegação. As posições dos sonares (Figura 29) em todos *Pioneer 3* são fixas: um em cada lado, e seis à frente com intervalo de 20 graus. O *DX* é fornecido com um semi-anel de sonares (oito ao total); o anel, totalizando 360 graus, é fornecido como item opcional.



Courtesy of ArtoMedia Robotics, LLC

Figura 29 – Semi-anel de Sonares

7.2.6 Motores, Rodas e Encoders

O sistema de tração do *Pioneer 3* utiliza motores DC reversíveis de alta velocidade e alto torque, cada um equipado com um encoder ótico de alta

resolução para definições precisas de posição e velocidade. As relações de engrenagens, pulsos por volta de encoder e tamanhos de pneus variam de acordo com o modelo do robô. Entretanto, o ARCOS pode corrigir estas diferenças promovendo uma odometria de alta precisão.

7.2.7 Bateria e Alimentação

Os robôs *Pioneer 3* contêm até três baterias, com troca a quente, 7 Ah, 12 VDC, seladas, e acessíveis via porta traseira. Contatos por mola da bateria ao sistema de potência do robô evitam a necessidade de se aparafusar conectores. Como as baterias operam em paralelo, o robô funciona com uma, duas ou três baterias, fato que vai indicar o tempo útil de operação do robô; quanto maior o número de baterias, maior o tempo útil de operação. Contudo, deve-se sempre lembrar de se balancear o peso das baterias no robô: no caso de uma bateria, ela deve ser colocada no centro, e caso se utilize duas, elas devem ser colocadas nas extremidades.

O *Pioneer 3-DX*, sem PC *onboard*, pode operar entre 8 a 10 horas com três baterias. Caso tenha o computador *onboard* o tempo cai para entre 3 a 4 horas.

– Indicadores de Bateria e Condições de Baixa Tensão

O Painel de Controle do Usuário tem um led bicolor denominado BATTERY que indica visualmente a tensão corrente da bateria. De aproximadamente 12,5 V e acima, o led apresenta cor verde brilhante. O led se torna progressivamente laranja e então vermelho se a tensão cair para aproximadamente 11,5 V.

O buzzer do Painel de Controle do Usuário irá soar um alarme repetitivo se a tensão da bateria cair consistentemente abaixo de 11,5 V. Se a tensão da bateria cair abaixo de 11 V o microcontrolador irá desligar automaticamente a conexão com o cliente e notificar o computador, via pino HOST RI, para que ele seja desligado evitando assim perda de dados ou corrupção do sistema.

– Recarregando as Baterias

O tempo típico de recarga de uma bateria utilizando o recarregador recomendado (800 mA) varia de acordo com o estado de descarga; equivale a três horas por Volt por bateria.

Com o recarregador de alta velocidade (4 A de máxima corrente), o tempo de recarga é drasticamente reduzido. Ele também fornece corrente suficiente para operar continuamente o robô e os acessórios *onboard*, tais como o PC *onboard* e rádios. Entretanto, com a utilização do recarregador de alta corrente, deve-se tomar a precaução de se recarregar pelo menos

duas baterias ao mesmo tempo. Com a carga de apenas uma bateria pode haver sobrecarga e danificar a si mesma e ao robô.

O tempo de carga para três baterias descarregadas (em torno de 11 V), utilizando o carregador de alta velocidade, é de 2,4 horas.

7.3 Acessórios

7.3.1 Conexão cliente-servidor

Todas as plataformas MOBILEROBOTS são servidores em uma arquitetura cliente-servidor. Deve-se fornecer o computador cliente para executar as aplicações inteligentes do robô móvel. O cliente pode ser um notebook ou *embedded* PC, ou um PC *offboard* conectado através de modems a radio ou Ethernet serial wireless. Em todos os casos, o PC cliente deve se conectar ao HOST interno ou a porta SERIAL no Painel de Controle para que o robô e software funcionem.

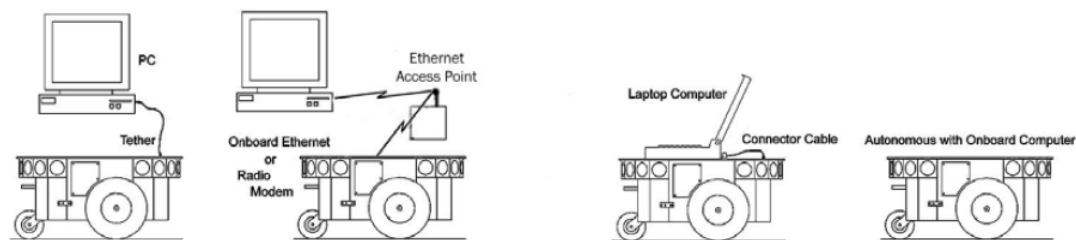


Figura 30 – Opções de conexão cliente-servidor

Para o notebook ou *embedded* PC, a conexão serial é feita via cabo serial. Modems a rádio podem substituir o cabo serial. Caso se tenha um modems a rádio, um deve estar dentro do robô e conectado à porta serial HOST, e outro modem deve estar ligado em algum computador *offboard* que estará rodando o software cliente.

Ethernet a rádio é um pouco mais complexa, mas é o método preferido porque permite que se utilizem diferentes computadores na rede para que se torne o cliente do robô. Caso se tenha um PC *onboard* (notebook ou *embedded*), ele pode fornecer a conexão Ethernet a rádio através de cartões PCMCIA. Atualmente, a grande maioria dos notebooks já tem integrado o controlador de Ethernet a rádio, o que facilita muito a comunicação sem fio.

7.3.2 PC Integrado

Montado logo atrás do nariz do robô, o PC integrado contém quatro portas seriais, 10/100Base-T Ethernet, portas para monitor, teclado e mouse, duas porta USB e suporte para disquete, bem como HD IDE. Para

funcionalidades adicionais, tais como som, compressão de vídeo, firewire ou barramento PCMCIA e Ethernet wireless, o PC *onboard* aceita os cartões de interface PC104 e PC104-plus, que são espetados na placa-mãe.

Os 5 VDC necessários vêm de um conversor dedicado DC:DC, montado próximo. O HD é montado no nariz do robô, entre um *cooler* e o *speaker* do computador.

O PC *onboard* comunica com o microcontrolador do robô através da porta serial HOST e porta serial dedicada COM1 no Windows ou `/dev/ttyS0` no Linux. O microcontrolador chaveia automaticamente a conexão HOST para o PC quando o software cliente abre a porta serial. Deve-se notar que o PC não interfere com clientes conectados externamente através da porta compartilhada SERIAL no Painel de Controle do Usuário.

– Operando o PC Onboard

Os softwares do robô e acessórios estão tipicamente instalados em `/usr/local` em sistemas Linux, ou `C:\Program Files\MobileRobots` no Windows.

No primeiro acesso ao PC *onboard*, é fortemente recomendado que o robô seja colocado sobre blocos para evitar possíveis movimentos indesejados. Deve-se conectar um teclado, monitor e mouse às suas respectivas portas no Painel de Controle do Computador, ligar o botão MAIN POWER e aguardar o início do sistema. Após o boot, deve-se logar no sistema. Já se encontra criado dois usuários: um com permissões de escrita e leitura de arquivos ('guest') e outra com acesso total ao PC e o OS ('root' no Linux e 'administrator' no Windows). Caso exista uma senha, ela é 'mobilerobots'. Deve-se notar que não é possível logar como 'root' no Linux quando conectado remotamente (via Telnet por exemplo). Para isto deve-se logar como 'guest', entrar o comando 'su -' e entrar com a senha de 'root'.

7.3.3 Rede

O conector RJ-45 no Painel de Controle do Computador fornece rede 10/100Base-T Ethernet diretamente com o PC *onboard*. É também instalado um cartão adaptador PCMCIA que fornece Ethernet sem fio. A antena da Ethernet wireless está localizada no topo do robô (mesa). Para completar a instalação wireless será necessário um *Access Point* para a LAN.

Tipicamente, o endereço IP da rede sem fio é 192.168.1.32, e que pode ser modificado de acordo com as necessidades. Os endereços IP podem ser modificados, no caso do Linux, nos arquivos `ifcfg-eth0` (Ethernet com fio) e `ifcfg-eth1` ou `ifcfg-wvlan0` (wireless) localizados na pasta

/etc/sysconfig/network-scripts. Feito isto a porta com endereço IP modificado deve ser reiniciada através do comando `ifdown <porta>;ifup <porta>`. Pode-se checar o status da porta via comando `ifconfig <porta>`. No caso do Windows, os endereços IPs podem ser modificados via Painel de Controle.

Caso se deseje mudar o endereço IP do robô no Linux, deve-se editar os arquivos `ifcfg-eth0` (Ethernet com fio) ou `ifcfg-eth1` (wireless). Tem-se como exemplo o seguinte arquivo `ifcfg-eth0`:

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.1.32
NETMASK=255.255.255.0
NETWORK=192.168.1.1
ONBOOT=yes
```

Os campos podem ser descritos da seguinte forma: `DEVICE`, indica o nome do dispositivo de rede; `BOOTPROTO`, indica como o endereço IP deve ser atribuído, pode ser do tipo `static` (endereço estático, atribuído no campo `IPADDR`) ou `dhcp` (endereço atribuído automaticamente quando a rede é inicializada); `IPADDR`, endereço IP da rede; `NETMASK`, máscara de rede; `NETWORK`, endereço do gateway; `ONBOOT`, indica se o dispositivo de rede deve ser inicializado no BOOT (`yes` para ser inicializado e `no` para não ser inicializado).

7.4 ARIA

O software *ARIA* (*Advanced Robotics Interface for Applications*), incluindo *ARNetworking*, vem com todas as plataformas MOBILEROBOTS. O *ARIA* é um ambiente de desenvolvimento *open-source* baseado em C++ que fornece uma interface robusta para uma variedade de sistemas inteligentes robóticos, incluindo o microcontrolador do robô e sistemas acessórios. O *ARNetworking* fornece a camada crítica para comunicações baseadas em TCP/IP com a plataforma MOBILEROBOTS sobre a rede.

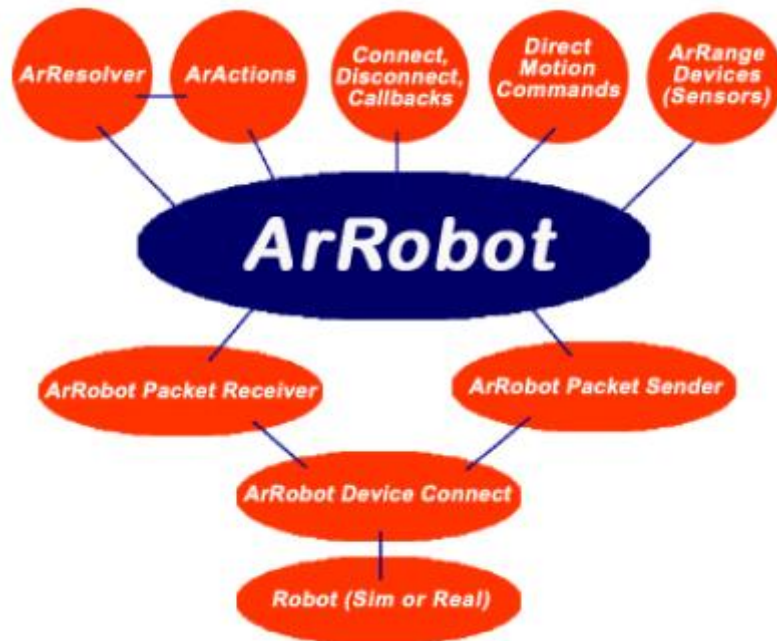


Figura 31 – Arquitetura do ARIA

A instalação do *ARIA* pode ser feita em qualquer computador com sistema operacional Linux instalado. Com o *ARIA* instalado em um computador, que não seja o do robô, é possível a realização de simulações com o auxílio do software *MobileSim* (a ser apresentado na próxima seção). Para a instalação deve-se executar o seguinte comando no terminal do Linux:

```
rpm -i Aria-2.4-1.i386.rpm
```

Após a instalação do *ARIA* tem-se o seguinte pacote localizado no diretório `/usr/local`

```

Aria/
  advanced  Demonstrações avançadas
  bin       Binário e dlls Win 32; binários Linux
  docs      Documentação extensa (deve-se executar o index.html em um
navegador para visualização)
  examples  Exemplos do ARIA
  include   Arquivos include do ARIA
  lib       Arquivos Win32 .lib e Win32 e Linux .so
  params    Arquivos de definição do robô (p3dx.p, por exemplo)
  src       Arquivos fonte ARIA (*.cpp)
  tests     Arquivos de teste
  utils     Comandos de utilidade
ArNetworking/  Uma biblioteca utilizada para facilitar a comunicação
na rede
  examples  Exemplos ArNetworking examples
  include   Arquivos include ArNetworking
  src       Arquivos fonte ArNetworking (*.cpp)
  
```

A documentação do ARIA (que se encontra no diretório *docs*) é uma importante fonte de consulta, pois descreve todas as suas classes, incluindo funções e variáveis. Esta documentação está no formato *html*, podendo ser visualizada em qualquer *browser*, tais como *Internet Explorer* ou *Mozilla Firefox*. A Figura 32 mostra a página inicial da documentação que apresenta um *overview* do ARIA. Neste *overview* o leitor iniciante irá encontrar uma introdução sobre o ARIA e também uma explicação geral sobre suas principais classes. As seções da documentação do ARIA podem ser acessadas via Menu Superior (Figura 33) ou Menu Esquerdo (Figura 34). As classes do ARIA são apresentadas na documentação através de hierarquia (Figura 33), lista alfabética (Figura 35) e lista explicativa (Figura 37). A classe escolhida é explicada com elevado grau de detalhamento, como o exemplo da classe *ArRobot* (Figura 36). Como o ARIA é *open-source*, os arquivos de classes são abertos e podem ser visualizados no menu “*File List*” (Figura 38); como exemplo tem-se o arquivo *ArRobot.cpp* (Figura 39) e *ArRobot.h* (Figura 40). Caso seja necessário detalhamento de alguma função ou variável do ARIA, sem que se saiba a que classe pertença, pode-se recorrer ao menu “*Class Members*” (Figura 41).

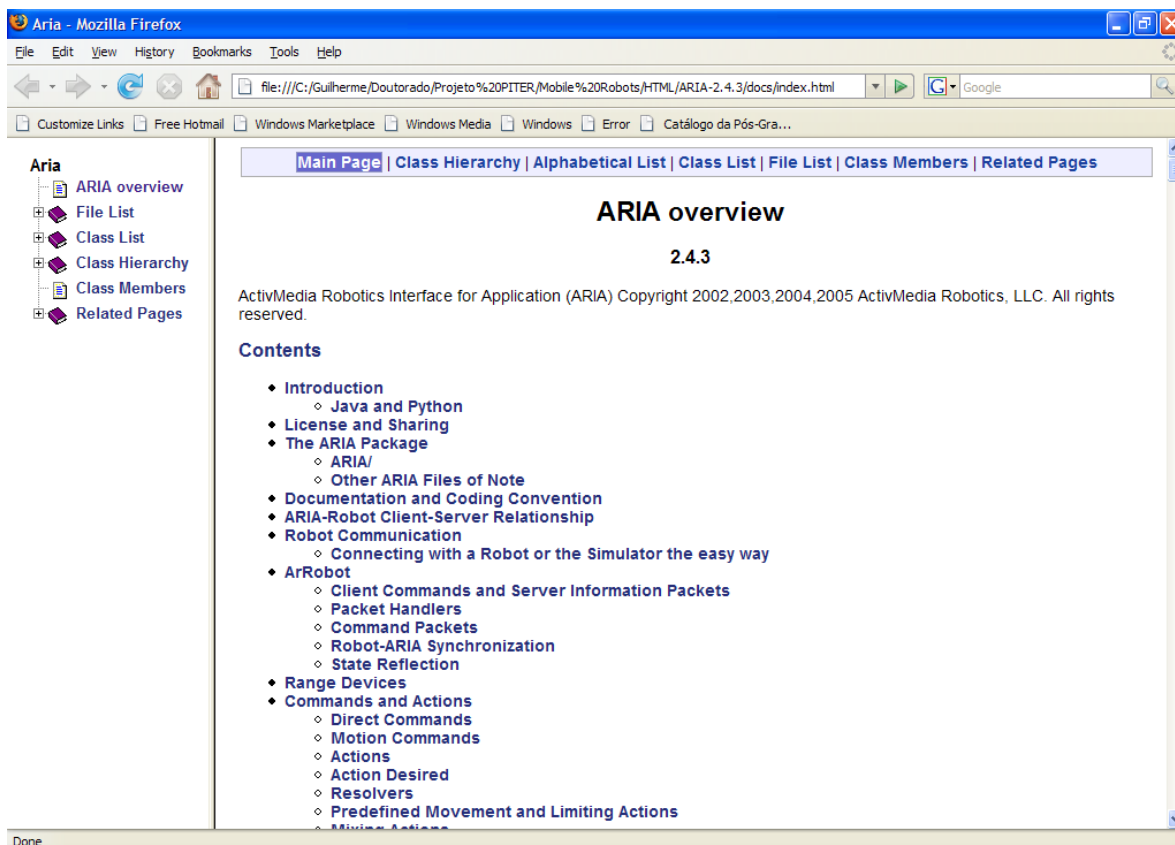


Figura 32 – ARIA overview

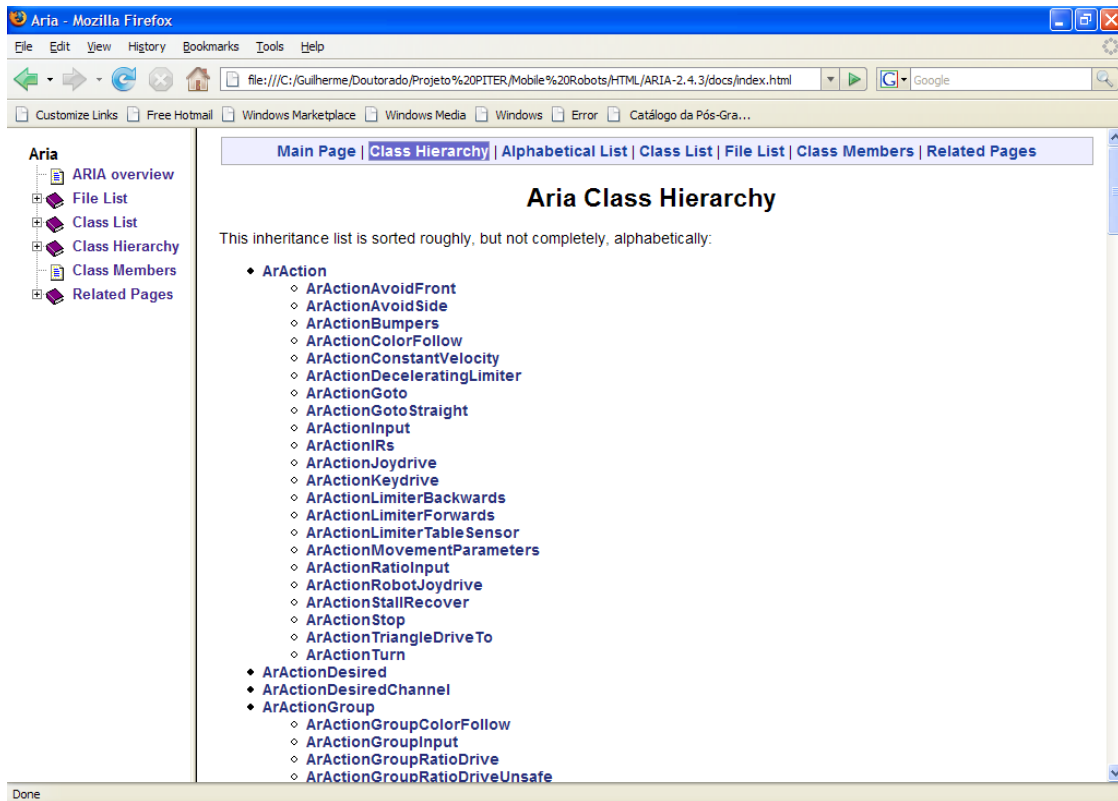


Figura 33 – Hierarquia de classes ARIA

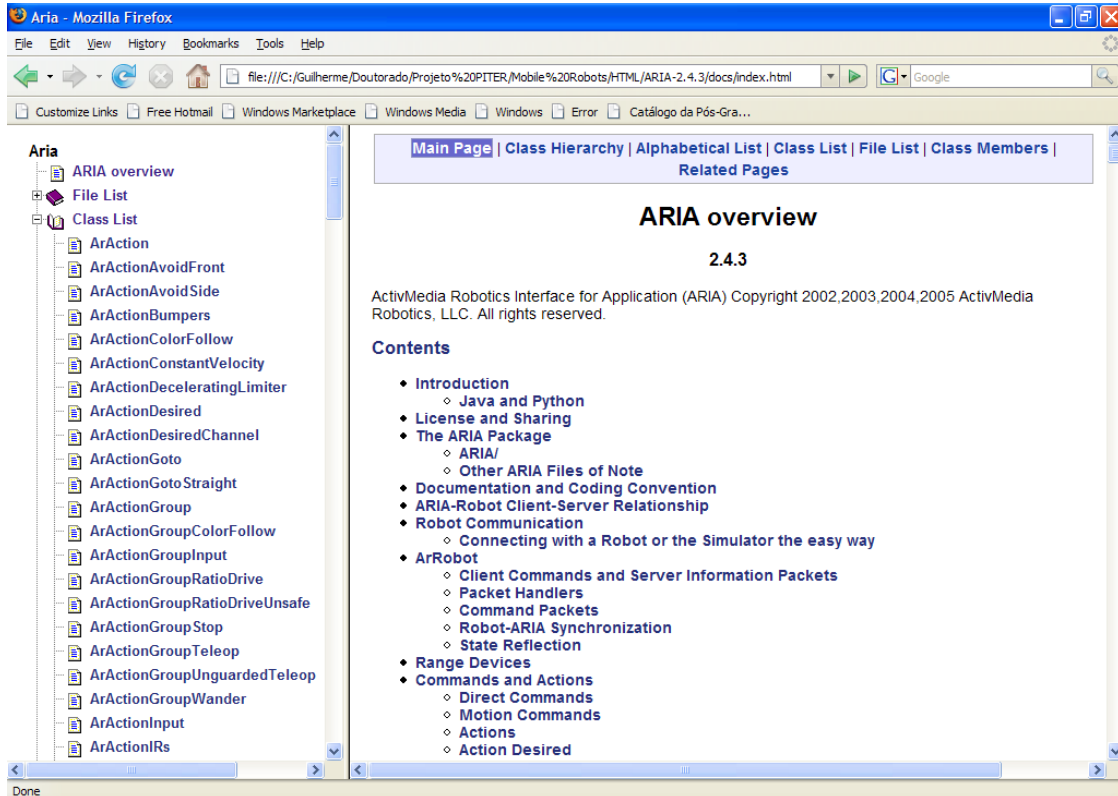


Figura 34 – Acesso via Menu Esquerdo

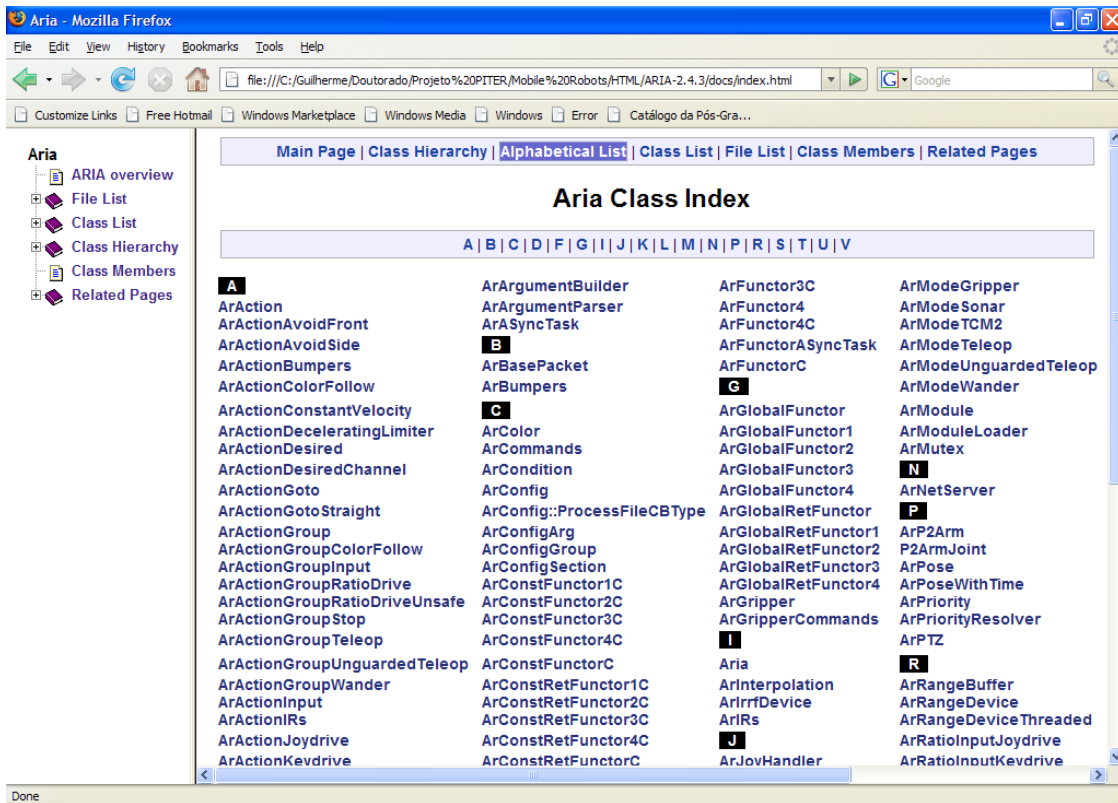


Figura 35 – Lista alfabética de classes ARIA

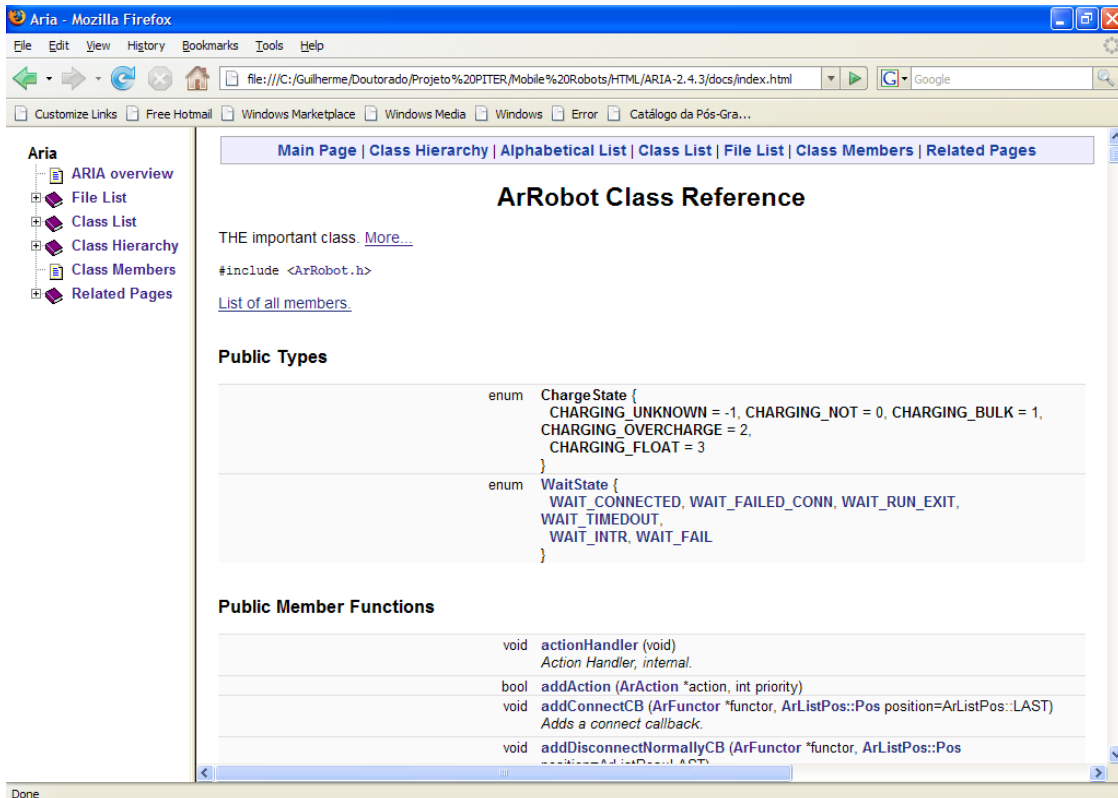


Figura 36 – Classe ArRobot

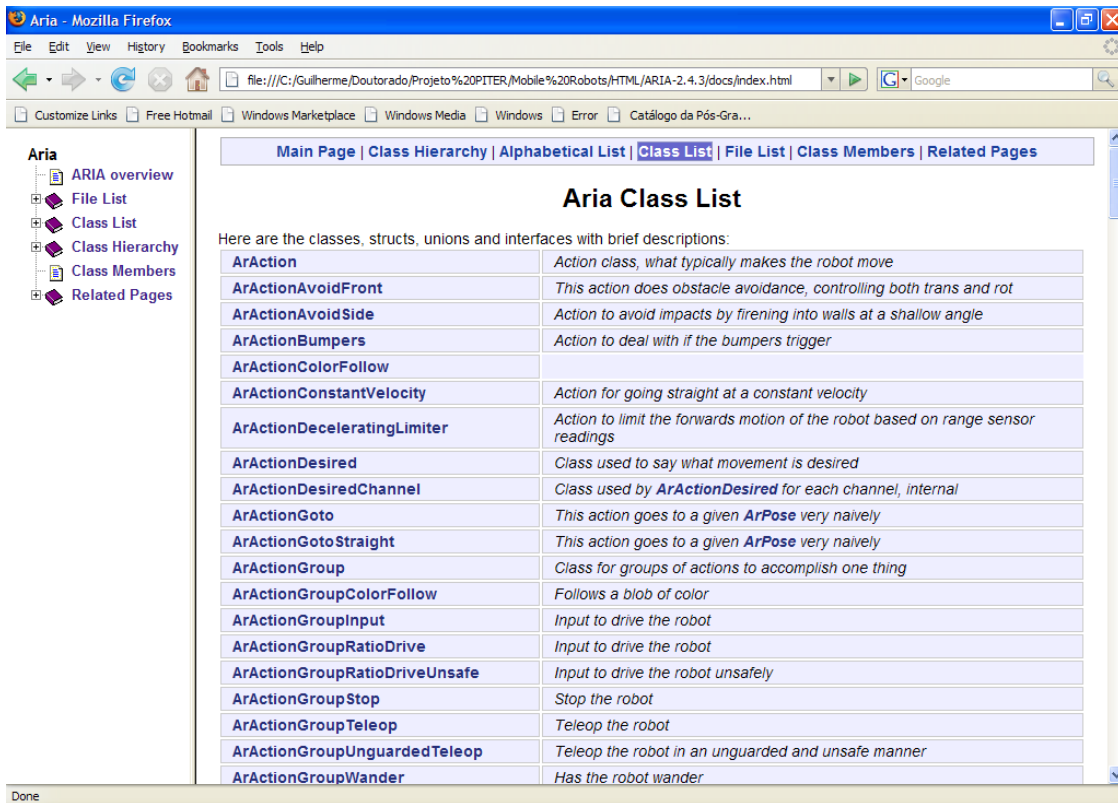


Figura 37 – Lista de classes ARIA

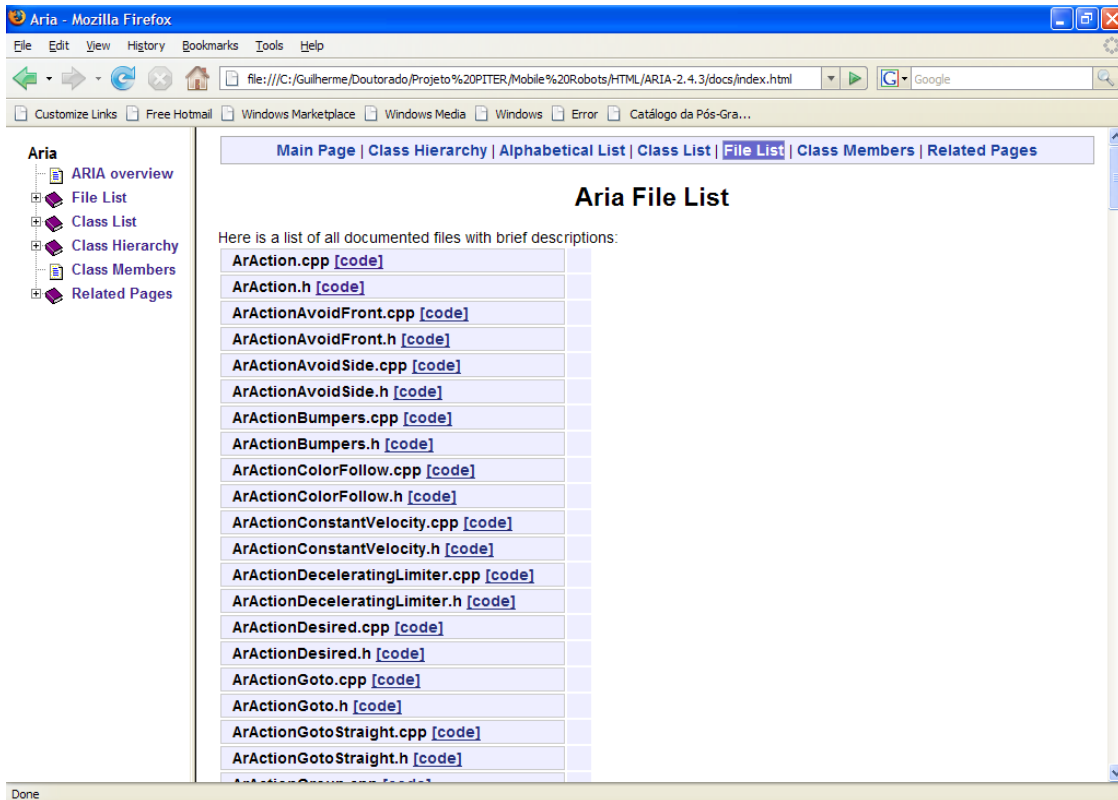


Figura 38 – Lista de arquivos ARIA

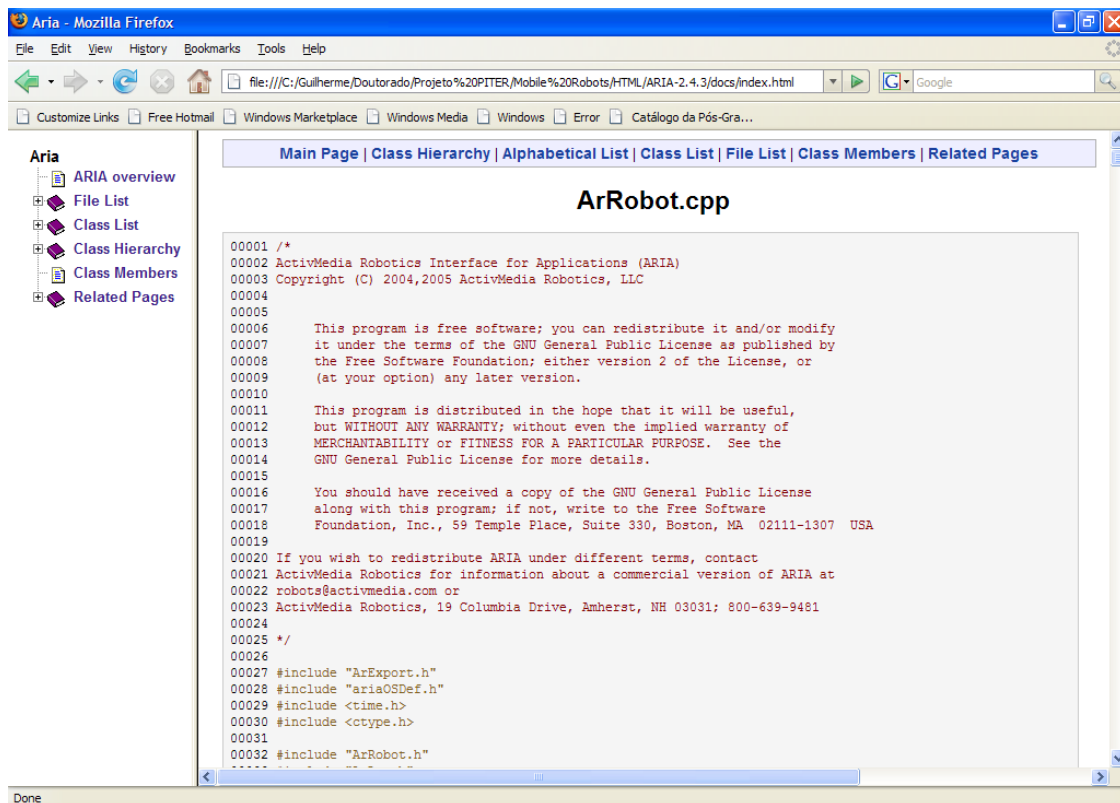


Figura 39 – Arquivo ArRobot.cpp

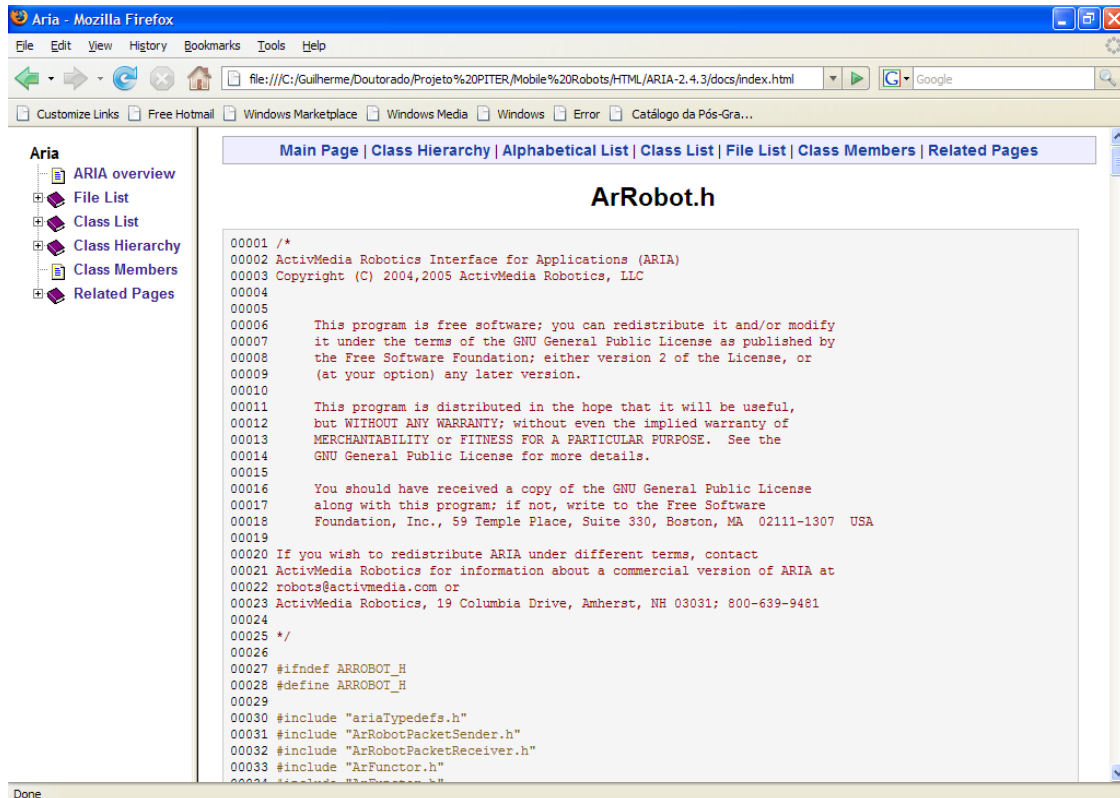


Figura 40 – Arquivo ArRobot.h

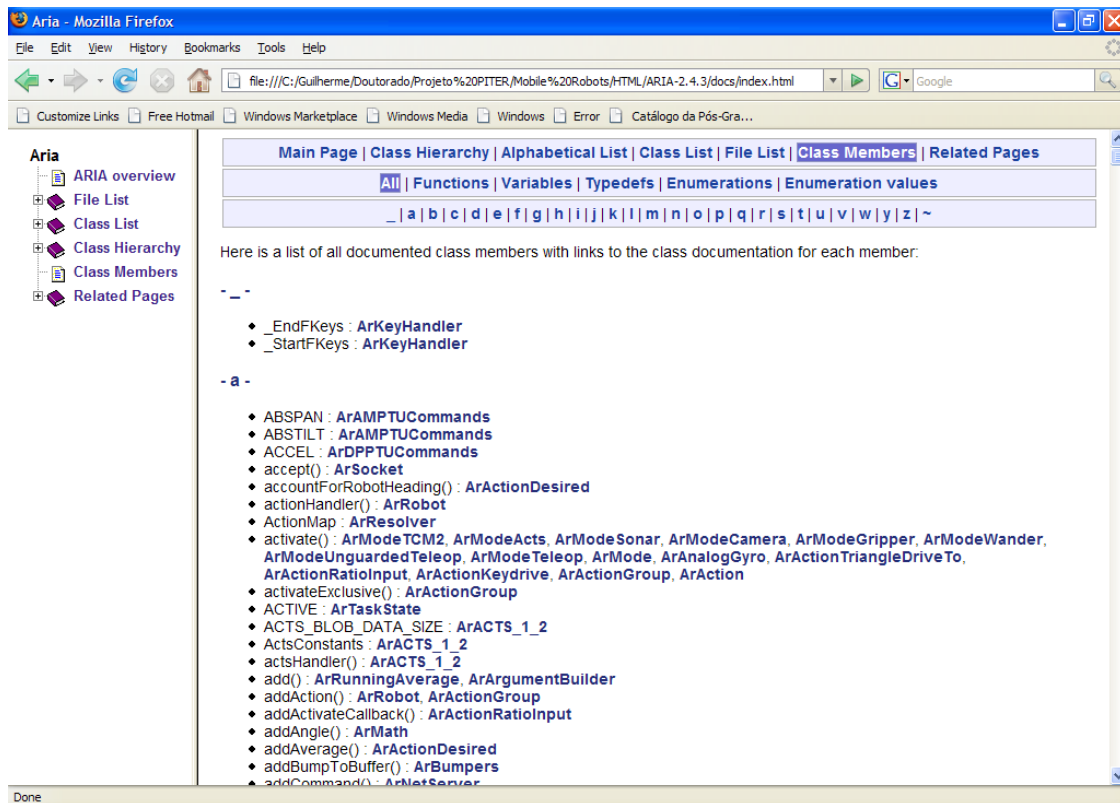


Figura 41 – Membros das classes ARIA

7.5 MobileSim

MobileSim é um software para simulação de robôs móveis e seus ambientes, realização de debug e experimentos em conjunto com programas baseados no *ARIA*. É baseado em biblioteca *Stage*, criada pelo projeto *Player/Stage/Gazebo* (<http://playerstage.sourceforge.net>).

O software *MobileSim* converte um mapa *ActivMedia* (um arquivo .map criado pelo software *Mapper3Basic*) para o ambiente *Stage*, colocando um modelo de robô simulado neste ambiente. Também provém uma conexão com o *Pioneer* simulado via porta TCP 8101 (similar à conexão via porta serial com o *Pioneer* real). A maioria dos programas baseados no *ARIA* conecta automaticamente na porta TCP caso ela esteja disponível.

A instalação do *MobileSim* pode ser feita em qualquer computador com o sistema operacional Linux instalado, e é feita via terminal executando-se o seguinte comando:

```
rpm -i MobileSim-0.2-0.i386.rpm
```

O *MobileSim* pode ser executado pelo seguinte comando via terminal do Linux:

```
cd /usr/local/MobileSim/bin  
./MobileSim
```

Na primeira tela do *MobileSim* deve-se escolher o mapa a ser utilizado e que se encontra no caminho `/usr/local/Aria/maps`. Caso não se queira trabalhar com nenhum mapa deve-se clicar em "No Map". As Figura 42 e Figura 43 mostram a seqüência de ações para se carregar o mapa.

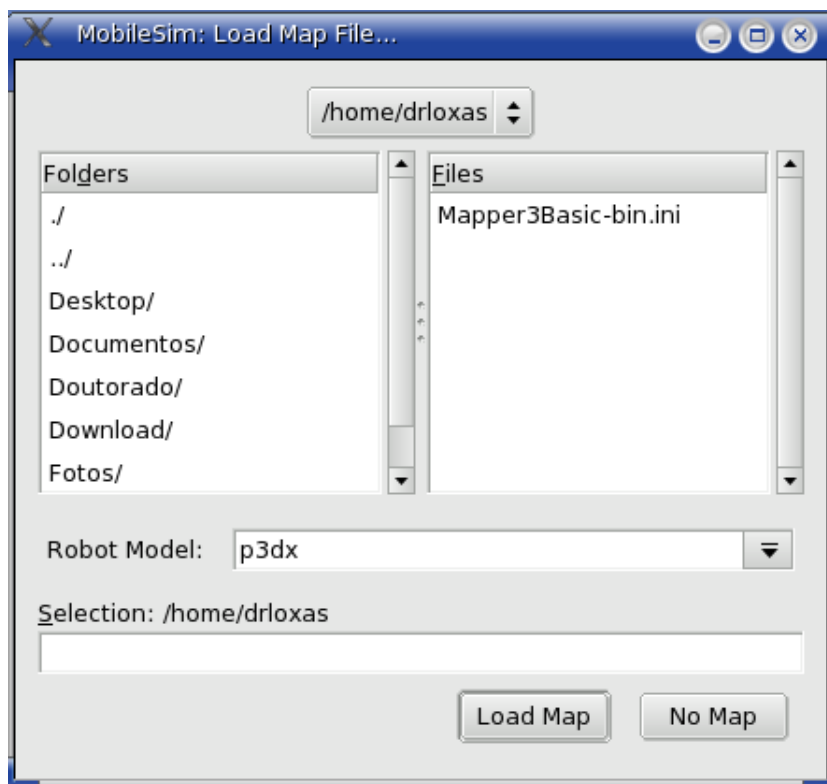


Figura 42 – Tela de inicialização do MobileSim

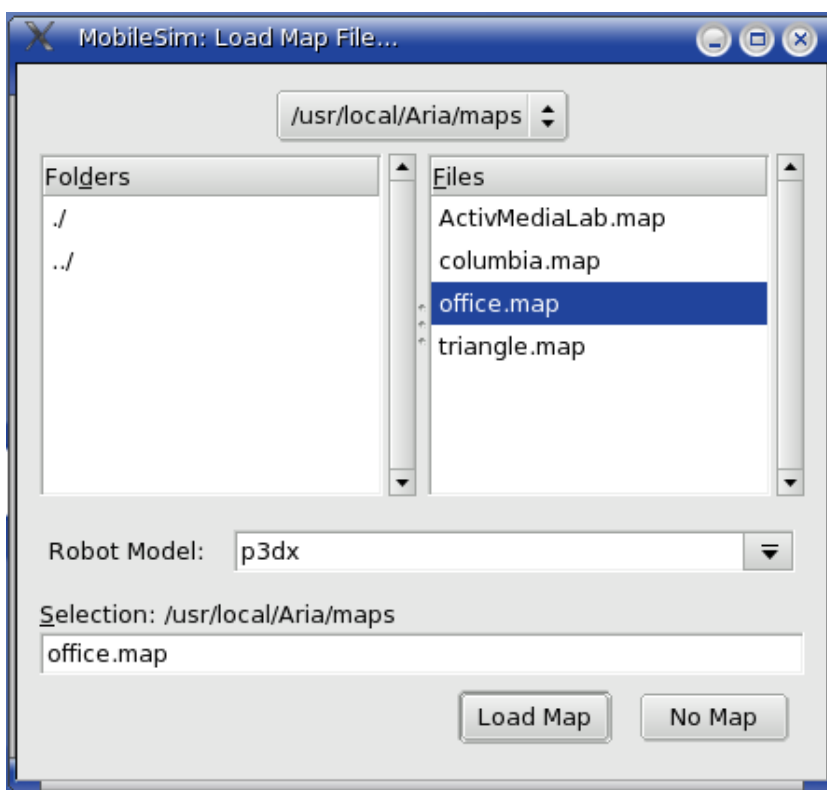


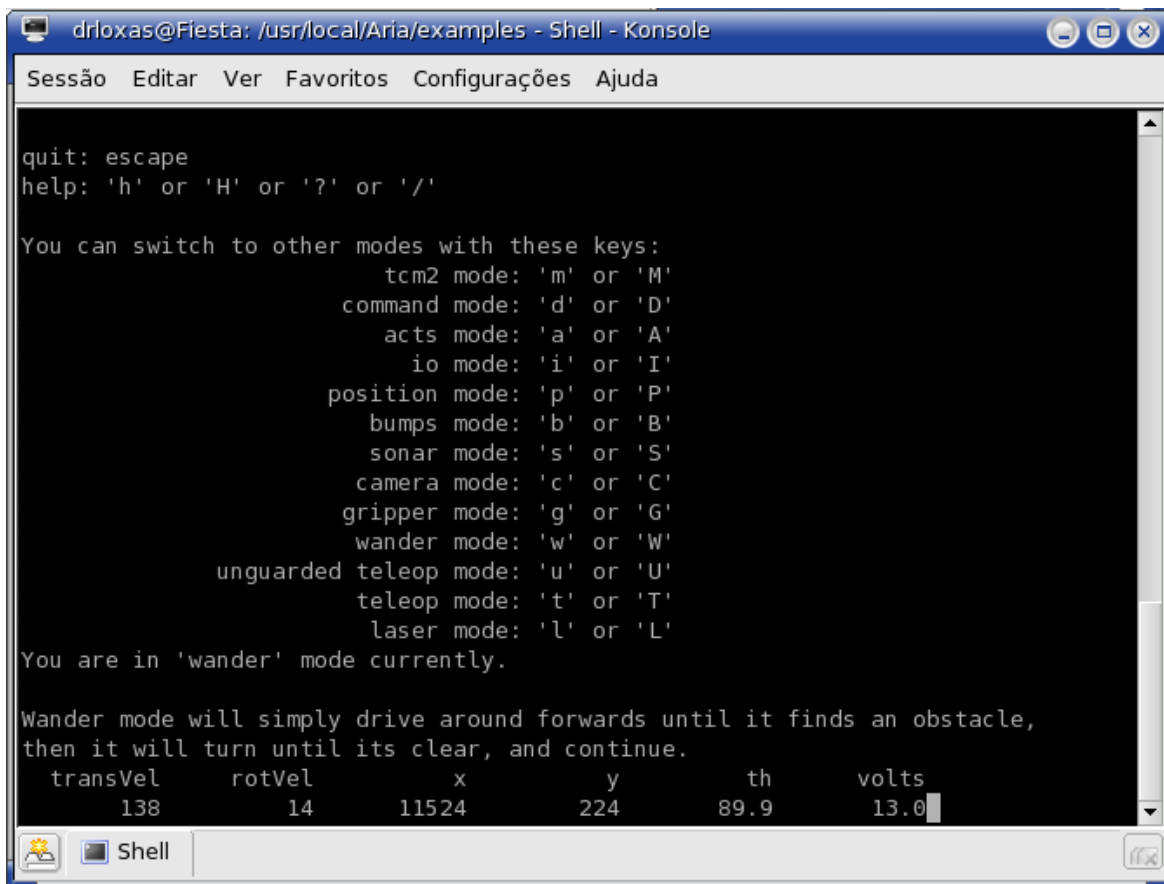
Figura 43 – Carregando o mapa no MobileSim

Após o carregamento do *MobileSim* deve-se abrir um programa desenvolvido com o *ARIA* para que seja executado no simulador. Como exemplo, o programa de demonstração será executado:

```
cd /usr/local/Aria/examples
./demo
```

Deve-se ter muito cuidado na execução de programas diretamente no robô, pois, geralmente, o programa tenta conectar com o simulador e caso o mesmo não esteja carregado, o programa é executado no robô, podendo causar acidentes. Apenas execute programas diretamente no robô caso tenha certeza de seu comportamento. Lembre-se, a execução direta envolve a conexão de teclado, mouse e monitor no robô, e colocação do mesmo sobre uma mesa e uma movimentação indevida neste momento seria desastrosa.

A Figura 44 mostra as opções disponíveis do programa *demo*. O robô pode ser colocado em modo de tele-operação via teclado pressionando-se a tecla "t". A Figura 45 mostra o robô em operação no *MobileSim*.



```
drloxas@Fiesta: /usr/local/Aria/examples - Shell - Konsole
Sessão  Editar  Ver  Favoritos  Configurações  Ajuda

quit: escape
help: 'h' or 'H' or '?' or '/'

You can switch to other modes with these keys:
      tcm2 mode: 'm' or 'M'
    command mode: 'd' or 'D'
      acts mode: 'a' or 'A'
        io mode: 'i' or 'I'
    position mode: 'p' or 'P'
      bumps mode: 'b' or 'B'
      sonar mode: 's' or 'S'
    camera mode: 'c' or 'C'
    gripper mode: 'g' or 'G'
    wander mode: 'w' or 'W'
unguarded teleop mode: 'u' or 'U'
    teleop mode: 't' or 'T'
    laser mode: 'l' or 'L'

You are in 'wander' mode currently.

Wander mode will simply drive around forwards until it finds an obstacle,
then it will turn until its clear, and continue.

transVel  rotVel      x      y      th      volts
    138      14      11524      224      89.9      13.0
```

Figura 44 – Execução do programa *demo*

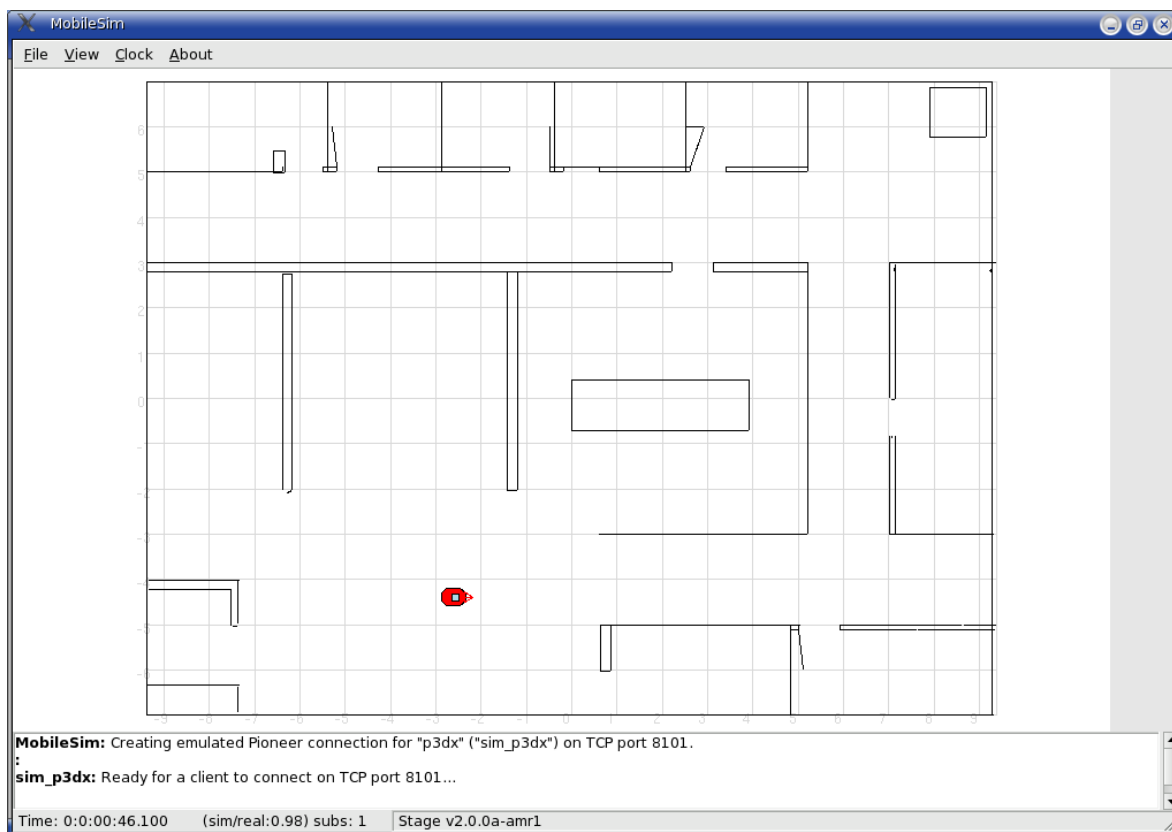


Figura 45 – Programa *demo* sendo executado no MobileSim

Pressionando-se a tecla "s" tem-se a visualização das medidas dos sonares no *Pioneer*, dadas em milímetros (Figura 46). Pode-se notar na Figura 47 os sonares ativos no simulador.


```
drlozas@Fiesta: /usr/local/Aria/examples - Shell - Konsole
Sessão Editar Ver Favoritos Configurações Ajuda

    acts mode: 'a' or 'A'
    io mode: 'i' or 'I'
    position mode: 'p' or 'P'
    bumps mode: 'b' or 'B'
    sonar mode: 's' or 'S'
    camera mode: 'c' or 'C'
    gripper mode: 'g' or 'G'
    wander mode: 'w' or 'W'
    unguarded teleop mode: 'u' or 'U'
    teleop mode: 't' or 'T'
    laser mode: 'l' or 'L'
You are in 'sonar' mode currently.

This mode displays different segments of sonar.
You can use these keys to switch what is displayed:
    '1': display all sonar
    '2': display sonar 0 - 7
    '3': display sonar 8 - 15
    '4': display sonar 16 - 23
    '5': display sonar 24 - 31
Sonar readings:
Displaying 0-7 sonar.
    0    1    2    3    4    5    6    7
  517  872 1923 4000 4000 3147 4000 4000
```

Figura 46 – Execução de opção *sonar* no programa *demo*

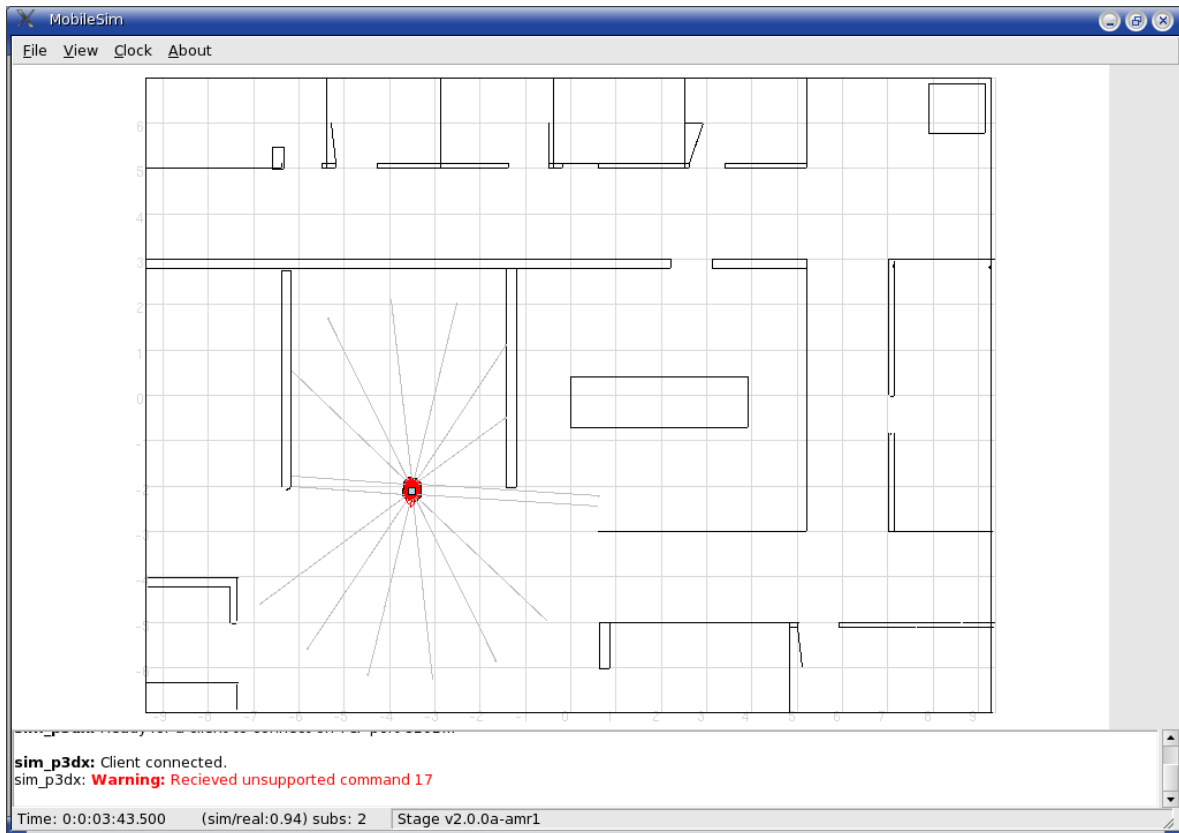


Figura 47 – Sonares ativos no MobileSim

7.6 Mapper3Basic

O *Mapper3Basic* é uma interface gráfica para mostrar e editar arquivos de mapa da *ActivMedia Robotics*. Com este software pode-se criar mapas que serão utilizados no *MobileSim*.

A instalação do *Mapper3Basic* é feita com a execução do seguinte comando no terminal do Linux:

```
rpm -i Mapper3Basic-1.2.1.rpm
```

Para a execução do *Mapper3Basic* deve-se executar o seguinte commando:

```
cd /usr/local/Mapper3Basic/bin  
./Mapper3Basic
```

A Figura 48 mostra o software *Mapper3Basic* em execução.

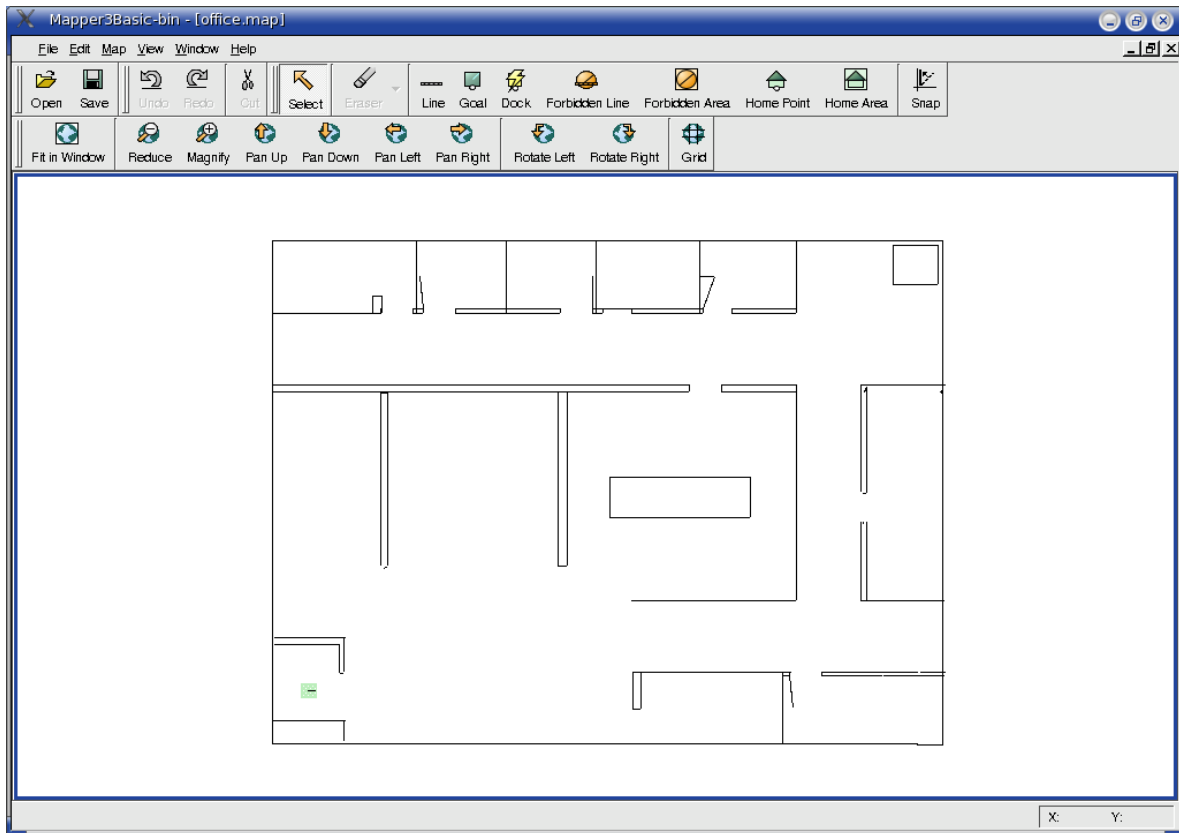


Figura 48 – Mapper3Basic

7.7 Exemplos de Programas utilizando o ARIA

Nesta seção serão apresentados dois exemplos introdutórios da utilização do *ARIA*. A programação é realizada linguagem C++ acrescida das classes e seus métodos contidos na biblioteca *Aria.h*. Maiores informações sobre as classes e seus métodos do *ARIA* podem ser obtidas no manual contido diretório *docs*.

O acesso ao computador *onboard* do robô deve ser feito via Telnet, através de um terminal Linux ou Windows. O comando para se conectar ao robô é:

```
telnet <endereço IP do robô>
```

Do terminal Telnet pode-se editar executar o programa desejado de forma segura. Entretanto, é fortemente recomendado que todos os programas sejam antes simulados no MobileSim, evitando possíveis acidentes, como por exemplo choques com paredes e objetos.

Caso o programa tenha sido editado em computador que não seja o *onboard* do robô, pode-se realizar o *download* via FTP. Os comandos para execução desta operação são:

```
ftp <endereço IP do robô>
user: guest
put <nomedoarquivo.cpp>
```

7.7.1 Exemplo 1

O programa “Exemplo 1” utiliza comandos de movimentação de translação e rotação do robô. O usuário deve entrar uma distância (em milímetros) de translação e uma rotação (em graus) para que o robô realize. Deve-se notar que a rotação somente será realizada após o término da translação.

A classe `ArSimpleConnector` é responsável pela conexão com o simulador *MobileSim* ou com o robô. Primeiramente o sistema tenta se conectar ao *MobileSim* via porta TCP, e caso o mesmo não esteja em execução o sistema tentar conexão com o robô.

O objeto `robot` deve ser criado com a classe principal do *ARIA*, `ArRobot`. Quando se utiliza o método `comInt` (da classe `ArRobot`) a comunicação é feita em baixo nível, diretamente com o ARCOS; também poderia ter sido utilizado `robot.enableMotors()` para habilitar os motores.

Como o programa é executado de modo assíncrono com a operação do robô, deve-se forçá-lo a aguardar o término do movimento antes de continuar, utilizando `while (!robot.isMoveDone())` para movimentos de translação e `while (!robot.isHeadingDone())` para movimentos de rotação. Caso se execute um comando de rotação logo após uma translação, os dois movimentos serão realizados de forma conjugada.

```
//Exemplo 1
//Realizacao de movimentos de translacao e rotacao

//Biblioteca Aria.h
#include "Aria.h"
int main(int argc, char** argv)
{
    //Inicio obrigatorio do Aria
    Aria::init();

    //Objeto conexão(simpleConnector) com simulador ou robô
    ArSimpleConnector simpleConnector(&argc,argv);

    //Objeto robo(robot)
    ArRobot robot;
```

```

//Testa argumentos
if(!simpleConnector.parseArgs() || argc>1)
{
    simpleConnector.logOptions();
    exit(1);
}

//Testa conexao com simulador ou robo
if(!simpleConnector.connectRobot(&robot))
{
    printf("Nao conseguiu conectar ao robo\n");
    Aria::exit(1);
}

//Inicializa variaveis de distancia(dist) e rotacao(rot)
int dist=1,rot;

//Inicializa robo
robot.runAsync(true);

//Habilita motores
robot.comInt(ArCommands::ENABLE,1);

while(dist!=0){ //Distancia igual a 0 termina o programa
    printf ("Entre com a distancia: ");
    scanf("%d",&dist);
    printf("Entre com a rotacao: ");
    scanf("%d",&rot);

    //Realiza a mesma funcao de robot.move(dist)
    //robot.comInt(ArCommands::MOVE,dist);

    //Desloca robô distancia dist
    robot.move(dist);

    //Aguarda termino do movimento
    while (!robot.isMoveDone());

    //Realiza a mesma funcao de robot.setHeading(rot)
    //robot.comInt(ArCommands::HEAD,rot);

    //Rotaciona robo rotacao rot
    robot.setDeltaHeading(rot);

    //Aguarda termino de rotacao
    while (!robot.isHeadingDone());
}

Aria::exit(0);
}

```

7.7.2 Exemplo 2

O programa "Exemplo 2" utiliza comandos de movimentação de translação e rotação do robô, velocidade variável de translação e leitura de valores dos sonares. O usuário deve entrar com a velocidade de translação e a distância mínima de um obstáculo frontal ou lateral. O robô inicia o movimento de translação com a velocidade fornecida e caso encontre um obstáculo à sua frente ou lado pára, realiza uma translação para trás, gira 180 graus e retorna o seu movimento de translação para frente até que encontre um novo obstáculo.

A classe `ArSonarDevice` é responsável pela criação do objeto `sonar` que deverá ser conectado ao objeto `robot` através de `robot.addRangeDevice(&sonar)`. Para que se tenha a leitura do menor valor informado pelos sonares frontais e laterais, deve-se utilizar o valor retornado pelo método `getClosestSonarRange(-90,90)`; considerando que o zero grau está no meio e a frente do robô, os valores -90 e 90 indicam a leitura dos seis sonares frontais e os dois laterais.

O controle de velocidade de translação é feito pelo método `setTransVelMax(vel)`, onde `vel` representa o valor da velocidade máxima de translação. A velocidade de translação executada pelo método `move(dist)` é determinada pelo método `setTransVelMax(vel)`. A utilização do método `setVel(vel)` indica que o robô irá realizar um movimento de translação, com a velocidade `vel`, até que o método `stop()` seja executado. Deve-se notar que a velocidade determinada por `setVel(vel)` está limitada à velocidade máxima determinada por `setTransVelMax(vel)`.

```
//Exemplo 2
//Movimentacao com velocidade variavel e utilizacao dos sonares
#include "Aria.h"
int main(int argc, char** argv)
{
    //Inicio obrigatorio do Aria
    Aria::init();

    //Objeto conexão(simpleConnector) com simulador ou robô
    ArSimpleConnector simpleConnector(&argc,argv);

    //Objeto robo(robot)
    ArRobot robot;

    //Testa argumentos
    if(!simpleConnector.parseArgs() || argc>1)
    {
        simpleConnector.logOptions();
        exit(1);
    }
}
```

```

//Testa conexao com simulador ou robo
if(!simpleConnector.connectRobot(&robot))
{
    printf("Nao conseguiu conectar ao robo\n");
    Aria::exit(1);
}

//Cria objeto sonar
ArSonarDevice sonar;

//Conecta o sonar ao robo
robot.addRangeDevice(&sonar)

//Inicializa variaveis de velocidade(vel), distancia(dist) e leitura
de sonar(valorsonar)
int vel,valorsonar,dist;

//Inicializa robo
robot.runAsync(true);

//Habilita motores
robot.comInt(ArCommands::ENABLE,1);

printf("Entre com a distancia minima da parede: ");
scanf("%d",&dist);
printf ("Entre com a velocidade: ");
scanf("%d",&vel);

//Determina a velocidade máxima de translacao do robo
robot.setTransVelMax(vel);

//Faz com que o robo se movimente com velocidade vel
robot.setVel(vel);

//Realiza medida do menor valor apresentado pelos sonares
frontais
valorsonar=robot.getClosestSonarRange(-90,90);

while(true){
    //Realiza medidas do sonar ate que se torne menor do que dist
    while(valorsonar>dist)
        valorsonar=robot.getClosestSonarRange(-90,90);

    //Pára o robo
    robot.stop();

    //Faz com que o robo retorne 500mm
    robot.move(-500);

    //Aguarda realizacao do movimento
    while(!robot.isMoveDone());

    //Faz com que o robô gire 180 graus
    robot.setDeltaHeading(180);

    //Aguarda termino de giro
    while(!robot.isHeadingDone());
}

```

```
        //Faz com que o robo se movimente com velocidade vel
        robot.setVel(vel);

        //Realiza menor medida dos sonares frontais
        valorsonar=robot.getClosestSonarRange(-90,90);

    }

    Aria::exit(0);

}
```