

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Trabalho de Conclusão de Curso:

**Plataforma I-cybie: Desenvolvimento e Estudo de
Modelos do Robô**

Aluna: Clarissa Fernanda Correia Lima Loureiro

Orientador: Bruno Barbosa Albert

Campina Grande
Fevereiro de 2008



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

Ofereço este trabalho ao meu pai,

Ricardo Loureiro (in memoriam)

Agradecimentos

Agradeço a minha mãe, professora Fernanda que me deu todo carinho, educação, apoio, exemplo e motivação desde os primeiros dias da minha vida.

Agradeço a minha tia “Tita” pela inspiração de personalidade.

Ao professor Antonio Marcus, pelo apoio e incentivo, por ser também um apaixonado pela robótica, e a todos que contribuíram para este trabalho.

Índice

1.	INTRODUÇÃO	5
2.	ETAPAS DA TRANSFORMAÇÃO EVOLUTIVA DE BRINQUEDO PARA ROBÔ	6
3.	PLATAFORMA DE DESENVOLVIMENTO	8
4.	SISTEMAS EMBARCADOS E COMPUTAÇÃO PERVASIVA	9
5.	BASE DE HARDWARE	10
6.	A ARQUITETURA DA PLATAFORMA	11
6.1.	Controle de movimento	11
6.2.	Kernel do Linux	12
6.3.	Câmera	12
6.4.	Processamento de imagem	13
6.5.	A API.....	13
7.	MELHORAMENTOS DA PLATAFORMA	15
8.	LOCOMOÇÃO DE QUADRÚPEDES	17
8.1.	Modelagem de Robôs Manipuladores.....	17
8.2.	O Modelo Geométrico.....	18
8.3.	O Modelo Cinemático	18
8.4.	O Robô I-cybie em Estudo	19
8.5.	O Modelo da Pata	20
8.6.	O Modelo do Robô.....	22
8.7.	Geração da Trajetória	22
8.8.	Otimização do Movimento.....	25
9.	CONCLUSÕES E CONSIDERAÇÕES FINAIS	26
	REFERÊNCIAS	27
	ANEXOS.....	28

1. INTRODUÇÃO

O desafio de se ter uma plataforma móvel real para testar algoritmos e software em vez de somente simular os dados, é um obstáculo para muitos pesquisadores e estudantes hoje em dia, pois soluções comercialmente disponíveis, são muito caras.

Um dos aspectos principais, que fizeram esta plataforma ser de baixo custo, se deve ao fato de que o i-cybie era um brinquedo comercial, e manufaturado em larga escala, o que fez com que a estrutura mecânica fosse acessível por um preço razoável.

Conseqüentemente, tomou-se possível adquirir um hardware barato e fazer as modificações necessárias para transformar o brinquedo em uma plataforma para desenvolver aplicações com robôs.

Portanto, esta servirá como um pilar estrutural para aplicações educacionais e de pesquisa.

A plataforma robótica utilizada é o brinquedo i-cybie, que possui comportamentos pré-programados e imita o comportamento de um cachorro. Possui 16 servos motores controlados por uma combinação de um controle remoto e uma CPU que utiliza um cartão de memória flash. O seu custo do ponto de vista de uma plataforma robótica era muito baixo, contudo, o preço para um brinquedo infantil era alto, por isso a Estrela parou de comercializar o i-cybie no Brasil, devido à falta de um grande mercado consumidor.

2. ETAPAS DA TRANSFORMAÇÃO EVOLUTIVA DE BRINQUEDO PARA ROBÔ

Etapa 1 : De Brinquedo a Robô

O padrão RS-232 foi proposto em 1962, e desde então tem sido usado intensivamente pela indústria. Esse padrão permite a comunicação entre um transmissor e um receptor a uma taxa de dados relativamente baixa (até aproximadamente 20Kb/s) e com baixas distâncias.

Para que o i-cybie fosse conectado a um pc, foi necessária a abertura do mesmo e uma modificação, para acrescentar um RS 232 ao seu circuito interno.

Desta forma, tornou-se possível conectar o brinquedo ao PC, o qual adquiriu uma capacidade muito maior de processamento, podendo ser chamado de robô,

Como a alimentação do i-cybie é de 3,6 volts, e a alimentação do pc é de 12V, inseriu-se um circuito conversor no cachorro, cujo esquema esta em anexo.

Etapa 2 : Comunicação bluetooth

Para que fosse possível existir uma comunicação bluetooth do robô com outros dispositivos externos, foi usado o modulo bluetooth Wintec BTV 42.

WBTV42 é uma linha de módulos Bluetooth embarcados que integra a faixa de controle para operações bluetooth, que incluem flash externo, cristal, e antena com um circuito RF eficiente.

Com uma pequena placa de circuito impresso, foi possível conectar o circuito do módulo a entrada Rx e a saída Tx do robô. Como a alimentação do módulo é 3.3V foi possível usar o mesmo nível de tensão do cachorro, que é 3.6V. Contudo, por precaução, foi também instalada uma alimentação extra com duas pilhas pequenas AAA, no caso de o módulo necessitar de uma corrente mais alta.

Desta forma, passa a existir uma porta serial, via bluetooth, tornando possível a conexão com outros dispositivos. O diagrama de pinos e a foto do módulo estão em anexo. Os detalhes da comunicação com o modulo pode ser encontrado no Simplified Starter Guide [1].

Etapa 3: Comunicação com a Internet

Depois da conexão do robô ao PC, através de uma pequena transformação no circuito interno do i-cybie, para a conexão de uma porta serial ao PC, já seria possível que o cachorro fosse conectado à internet, pois poderiam ser mandados comandos para o robô pela internet através do PC.

Mas ainda existia um cabo serial que limitava os movimentos do cachorro. Contudo, a conexão do módulo bluetooth tornou possível uma comunicação com outros dispositivos externos como o N800. Sendo assim, poderia haver uma conexão de internet sem fio, e o cachorro estaria ligado pela rede ao mundo inteiro.

Etapa 4: Conexão com o Roteador

Com a finalidade de deixar o robô mais independente, sem estar preso ao PC, pensou-se em conectar o robô a um roteador. O modelo escolhido foi o Asus WL-500g Deluxe, com Processador MIPS de 200MHz, 32MB de memória RAM, 4MB de memória Flash.

Foi inserida a distribuição Linux para roteadores sem fio, o firmware Open Wrt [2].

Foram feitas ainda, modificações nas conexões de uma porta serial e duas USB, de modo que no final tivesse duas portas seriais e quatro portas USB para viabilizar a conexão com o cachorro através da serial e com duas câmeras de baixa resolução através das portas USB.

3. PLATAFORMA DE DESENVOLVIMENTO

Uma plataforma de baixo custo foi concebida para desenvolver aplicações em robótica móvel que explorem recursos de conexão de dispositivos móveis programáveis sem fio.

A plataforma consiste em um brinquedo modificado, em um subsistema de visão de computador com uma câmera de baixa definição, e um subsistema de comunicação wireless: um roteador wireless, o módulo WBTV42 da Wintec e um cliente móvel, o Internet Tablet N800. Existem três CPU's interagindo na plataforma proposta, dentro do brinquedo cão modificado, dentro do roteador e dentro do cliente móvel.

O robô, quadrúpede, controlado via wireless, é um sistema complexo que traz alguns dos métodos e das técnicas de um campo emergente, desenvolvido para tratar dos problemas que impedem atualmente os robôs com patas de serem mais utilizados em situações reais, pelo seu alto custo no mercado.

O conhecimento de sistemas embarcados, rede de comunicação wireless, programação em tempo real, processamento de imagem, computação pervasiva e controle de movimentos é necessário para desenvolver uma plataforma que dê oportunidade de estudar um sistema complexo de uma maneira mais simples.

Em geral, uma plataforma de desenvolvimento é uma estrutura que possui dispositivos de software e de hardware, e que torna mais fácil as tarefas do projetista, para resolver eficientemente problemas complexos. É importante ter as ferramentas necessárias para permitir que o projetista focalize nos problemas centrais, abstraindo os detalhes que não são essenciais em determinados estágios do projeto.

Esta plataforma pode ser utilizada para varias aplicações domésticas reais como um sistema de segurança doméstica, para destacar os benefícios de usar as configurações propostas. O robô cão i-cybie usa processamento de imagem para detectar um invasor e uma vez que uma situação de risco é detectada, o relatório da invasão é emitido a uma companhia de segurança privada, ou informada ao proprietário da residência.

4. SISTEMAS EMBARCADOS E COMPUTAÇÃO PERVASIVA

Uma combinação de hardware e software, e de peças mecânicas dentre outras partes, projetadas para executar uma aplicação específica pode ser considerada como uma definição de um sistema embarcado.

Tem-se um sistema computadorizado especializado, o qual faz parte de um sistema maior, embarcado na plataforma, projetado para executar algumas funções específicas, o que configura nossa plataforma como um sistema embarcado.

Se a plataforma pode se comunicar com o ambiente através da Internet, é possível alcançar a informação de qualquer lugar, conduzindo o poder computacional a qualquer ponto do espaço. Conseqüentemente, o cão e seu proprietário podem estar conectados, tornando-se nós de uma rede, os quais podem fornecer dados e receber informações, o que constitui um sistema interativo e complexo.

Computação pervasiva é a idéia de que todo dispositivo pode ser conectado a uma rede infinita de outros dispositivos. O objetivo da computação pervasiva é de criar um ambiente onde a conexão dos dispositivos seja embarcada de tal maneira que a conexão não seja interrompida e sempre disponível. Mesmo sendo o sistema implementado com recursos complexos, para o usuário, o sistema tem um comportamento homogêneo e independente e as interfaces são ergonomicamente desenvolvidas de modo a facilitar a comunicação homem-máquina.

5. BASE DE HARDWARE

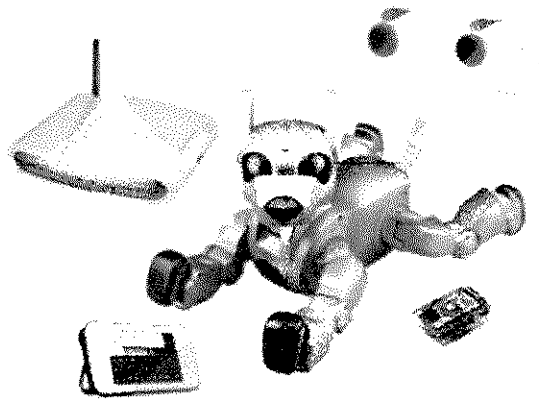


Fig1: Componentes do Hardware do Sistema

Faz parte do hardware do sistema: o próprio i-cybie, o roteador, o módulo bluetooth e duas câmeras de baixa resolução, para fornecer a visão do robô; todos os componentes ilustrados na figura acima.

Assim, com esta base de hardware, nosso robô quadrúpede incrementou o seu processamento, visão computacional, e uma comunicação sem fio, tudo isso com componentes de baixo custo.

A plataforma fornece uma maneira de pesquisar o problema do controle de trajetória, já que este necessita de um ambiente dinâmico e de uma estrutura física adequada. Além do que, como existe uma porta serial livre, sensores como o sonar, de toque, ou outros podem ser facilmente adicionados ao hardware, para melhorar a pesquisa sobre problemas como o mapeamento de um ambiente desconhecido.

A comunicação externa pode ser usada para transferência de informações, comunicação com um computador ou qualquer outro dispositivo móvel, para acessar os dados e poder fazer uma análise mais robusta dos mesmos.

6. A ARQUITETURA DA PLATAFORMA

A arquitetura proposta é constituída de componentes de hardware e software.

A figura abaixo representa um diagrama de blocos em que os detalhes da plataforma podem ser observados na descrição de cada bloco a seguir:

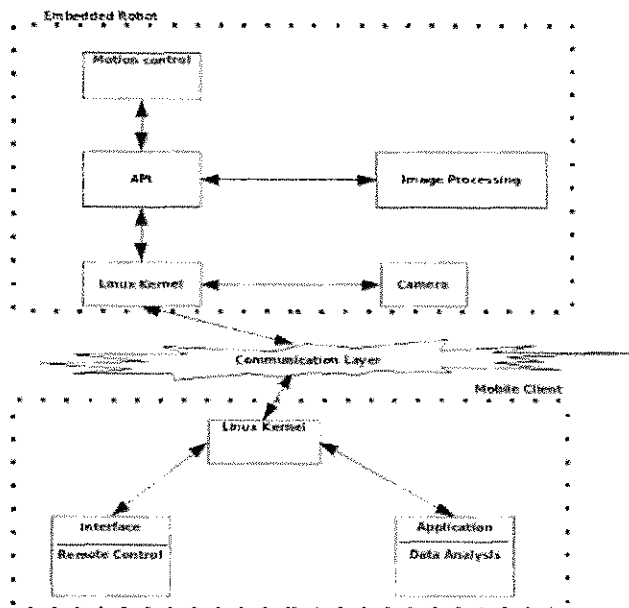


Fig2: Diagrama de blocos da arquitetura

6.1. Controle de movimento

O bloco de controle de movimento corresponde ao controle direto, de baixo nível, dos servos motores do robô. No exemplo específico do I-cybie, há 16 servos motores. Os servos podem ser controlados com o uso do SDK do I-cybie (da Aibopet). Foi feita a inclusão completa da SDK ao API, de uma maneira muito mais simples que evita que o projetista desperdice tempo para aprender a programar movimentos no SDK.

6.2. Kernel do Linux

O Kernel do Linux é a camada mais baixa da interface do sistema operacional no hardware, e é responsável por controlar os recursos computacionais do sistema.

Como a plataforma é open source, todas as características e potencialidades do kernel podem ser alcançadas e mudadas, de acordo com as necessidades do projetista.

As distribuições Linux, existentes na plataforma são: OpenWRT no caso do roteador ou o Internet Tablet OS 2007, no N800, que é usado como o cliente móvel nas aplicações. O primeiro é descrito, por seus autores, como uma distribuição de Linux para dispositivos embarcados que em vez de tentar criar um único firmware estático, fornece um sistema inteiramente livre para escrita, com gerenciamento de pacotes. Isto desobriga o projetista da seleção limitada de aplicação e configuração fornecidas pelo fabricante do dispositivo, permitindo uma customização do dispositivo com o uso dos pacotes adequados para várias aplicações.

Para projetistas, é uma estrutura (framework) para construir uma aplicação sem ter que construir um firmware completo em torno dela; para usuários isto significa a habilidade para uma customização inteira, usar o dispositivo de maneiras nunca pensadas.

O Internet Tablet OS 2007 da Nokia consiste em uma plataforma Maemo open source, a plataforma de desenvolvimento de aplicação para Internet Tablets, com a adição de componentes Closed Nokia.

Sendo a plataforma construída com padrões open source, um outro cliente móvel do robô poderia ser o OLPC (One Laptop Per Child) [3]. Consequentemente, o robô podia ser usado para adicionar mais interatividade e familiarizar as crianças no aprendizado com máquinas.

6.3. Câmera

Como existe o Linux embarcado na plataforma, é importante usar o Video4linux (a API original do kernel Linux para captação de vídeo), que é um dispositivo compatível, porque otimiza a imagem e simplifica a tarefa de adquirir a imagem da câmera, permitindo o acesso direto ao buffer da câmera.

Existem vários drivers para Video4Linux tal como Videodog [4] que fornece uma rápida e completa captura, ajuste de imagens, e uma grande variedade de opções para capturar e salvar o buffer para análise posterior, incluindo os formatos “raw” e JPEG.

6.4. Processamento de imagem

O bloco de processamento de imagem executa algumas análises no buffer da câmera, procurando por todos os aspectos que puderem ser considerados e usados nas aplicações desenvolvidas. Os aspectos principais são a cor, realizada pela análise de RGB que fornece o reconhecimento da cor pelo robô e por causa de sua simplicidade, a biblioteca Cimg [5] é usada; a detecção de movimento também pode ser implementada em muitas aplicações, mas precisa de um algoritmo mais elaborado.

6.5. A API

O programa do projetista age como um programa de controle principal, para onde todas as informações são destinadas. Para o desenvolvimento de tal programa, duas bibliotecas principais são fornecidas: `robot_actions`, para executar o controle da parte mecânica, `robot_vision`, com a subclasse `grab_image` para adquirir as imagens e processá-las.

A API abre uma série de opções para ganhar desempenho e qualidade nas tarefas comuns ao desenvolvimento.

Para demonstrar como a API funciona globalmente, é possível propor um problema simples a ser resolvido pelo desenvolvedor. Suponha que o robô precisa fazer um reconhecimento da cor, tal que poderia encontrar um cartão verde no ambiente e executar o movimento de andar, pois o cartão verde representaria um ambiente livre do perigo. O seguinte código representa a simplicidade da solução para este problema, quando o desenvolvedor tem a base do software fornecida por uma plataforma, uma função simples que procura por pontos verdes na foto:

```
/* Solução Simples
*/
#include "libs/grab_image.h"
#include "libs/robot_vision.h"
#include "libs/robot_actions.h"

while (true) {
grab_image(buffer);
if(is_green_card_present(buffer))
    robot_actions_send_command(WALK);
}
free(buffer);
}
```

7. MELHORAMENTOS DA PLATAFORMA

Mesmo tendo inúmeras possibilidades de aplicação para a plataforma desenvolvida, esta pode ser melhorada de várias formas. A aquisição de dispositivos, sensores e câmeras de alta resolução está sendo feita aos poucos, visto que o projeto da plataforma ainda não tem um patrocinador oficial. Mesmo assim, foi possível adquirir o Internet Tablet N800.

N800

Primeiramente, a aquisição de um N800 para fazer parte do sistema, eliminou o roteador e incrementou ainda mais o processamento, de modo que temos agora, um mini-computador inserido no I-cybie.

Desta forma não seria mais necessário o fio que prendia o robô ao roteador, e podemos ter o N800 nas “costas” do cachorro, tornando-o muito mais automatizado e flexível a uma gama ainda maior de aplicações.

A câmera utilizada pode ser também do próprio N800. Além do que a comunicação entre o robô e o N800 se dá por bluetooth, portanto, não será necessário nenhum fio entre os componentes do sistema. De forma que não mais limitará os movimentos do cachorro.

SDK

O SDK (Software Development Kit) é um conjunto de ferramentas que permite que o projetista possa criar aplicações variadas, de uma forma que não seja necessário perder muito tempo na aprendizagem de uma linguagem para a programação de movimentos no robô.

Embora já existam seqüências de movimentos no SDK do I-cybie, estes ainda são limitados e podem ser otimizados.

Por este motivo, faz-se necessário a aprendizagem e aperfeiçoamento da tradução do SDK desenvolvido para os movimentos do I-cybie.

Modelagem

Em paralelo ao aperfeiçoamento do SDK, faz-se mister um estudo específico do modelo do robô. É imprescindível conhecer os movimentos ótimos de um quadrúpede. A melhor forma para isso é a observação dos animais. Pois estes possuem a forma perfeita para a seqüência de movimentos para locomoção.

Seria injusta, porém, uma comparação da plataforma em desenvolvimento ao um animal quadrúpede, pois este possui milhares de sensores em suas patas e os seus algoritmos biológicos estão muito mais a frente do que qualquer sistema artificialmente desenvolvido atualmente no mundo.

O que se pretende neste trabalho, é efetuar um estudo de um modelo para o brinquedo I-cybie modificado.

8. LOCOMOÇÃO DE QUADRÚPEDES

O processo de locomoção de animais quadrúpedes é bastante complexo e requer a coordenação de vários músculos para a manutenção de uma postura estável enquanto o animal se desloca. Em se tratando de robôs com patas, a situação não é diferente. Para uma locomoção estável da plataforma robótica, é necessário a geração sistemática e periódica de seqüências de movimentos para as patas do robô, expressas como posições desejáveis para os pés a cada instante de tempo.

8.1. *Modelagem de Robôs Manipuladores*

Um manipulador robótico é composto por segmentos conectados por juntas e um efetuator final (*end-effector*), formando o que chamamos de cadeia cinemática. O efetuator corresponde à última peça da cadeia. As juntas podem ser prismáticas (movimento linear) ou rotacionais (movimento rotacional).

As variáveis das juntas prismáticas são distâncias denotadas por d e representam o deslocamento relativo de dois segmentos adjacentes. Já as variáveis das juntas rotacionais são ângulos denotados por θ e representam a rotação relativa entre dois segmentos adjacentes. O conjunto de variáveis das juntas de um manipulador é dado por um vetor q , sendo $q_i = \theta_i$ ou $q_i = d_i$, dependendo do tipo da junta.

Quando uma junta i atua, o segmento i move-se. Associado ao segmento i , existe um sistema de coordenadas $X_i \times Y_i \times Z_i$, o qual também se move com a atuação da junta i , conforme ilustrado na figura abaixo:

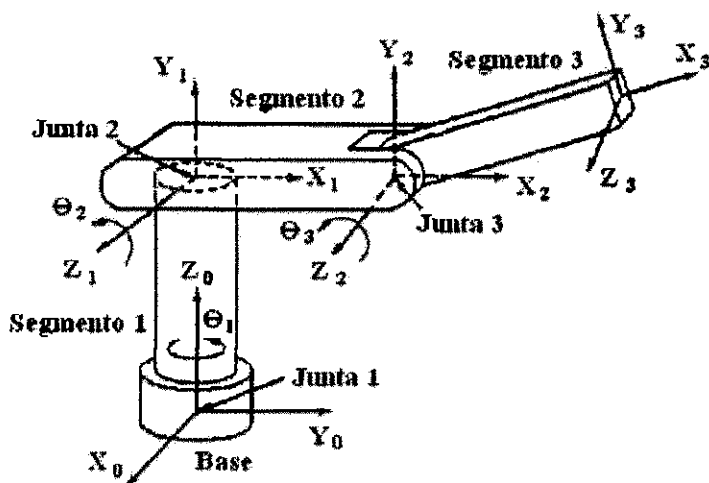


Fig3 Manipulador com três segmentos

Um objeto possui m graus de liberdade caso sua configuração (isto é, a localização de cada ponto do objeto) possa ser especificada por, no mínimo, m variáveis. Por conseguinte, para um manipulador robótico, cada junta determina um grau de liberdade.

8.2. O Modelo Geométrico

O modelo geométrico de um robô é dado por duas representações distintas: direta e inversa. O modelo geométrico direto (MGD) consiste em se obter a posição e a orientação do efetuador de um manipulador robótico, desde que se conheçam as variáveis de suas juntas. Já o modelo geométrico inverso (MGI) diz respeito à determinação das variáveis das juntas, dadas a posição e a orientação do efetuador.

8.3. O Modelo Cinemático

Tal como o modelo geométrico, o modelo cinemático de um robô também relaciona o efetuador às variáveis das juntas do manipulador. No entanto, considera-se a velocidade do efetuador, e não mais sua posição e orientação. É importante salientar que neste tipo de modelagem ainda não são levados em conta as forças e os torques que atuam no sistema.

Há também duas representações distintas: direta e inversa. O modelo cinemático direto (MCD) consiste na determinação da velocidade do efetuador, dada a velocidade das variáveis das juntas. Já o modelo cinemático inverso (MCI) consiste na determinação da velocidade das variáveis das juntas dada a velocidade do efetuador.

8.4. O Robô I-cybie em Estudo

O robô I-cybie em estudo, possui quatro patas com três graus de liberdade cada uma, totalizando doze graus de liberdade para o movimento. A metodologia proposta para estudo e obtenção dos modelos geométrico e cinemático consiste em modelar primeiro cada pata em separado e, em seguida, considerar o corpo como um todo.

Assim, cada pata foi considerada, inicialmente, como um manipulador robótico composto por três segmentos conectados por três juntas rotacionais e um efetuador, formando a cadeia cinemática.

O efetuador corresponde à peça final da cadeia, ou seja, o pé. A base do manipulador corresponde ao ombro e conecta a n -ésima pata ao corpo do robô ($n = 1, 2, 3, 4$). As patas 1 e 2 são dianteiras, enquanto que as patas 3 e 4 traseiras.

Raciocinando-se no espaço tridimensional, fica difícil trabalhar sem especificar um sistema de coordenadas em cada segmento do manipulador robótico. Existem diversas maneiras distintas de se alocar estes sistemas. No entanto, a determinação deles por meio da convenção de Denavit-Hartenberg, [7] proposta em 1955, tem-se tornado comum entre os engenheiros do mundo inteiro, favorecendo o surgimento de uma linguagem universal. Após essa alocação, a forma física do robô deixa de ser relevante, uma vez que cada eixo encontra-se completamente definido pelo sistema de coordenadas.

8.5. O Modelo da Pata

Por possuir três juntas e um efetuador, (figura abaixo) cada pata será descrita por quatro sistemas de coordenadas. A escolha da orientação dos eixos de cada sistema de coordenada foi realizada segundo a convenção de Denavit-Hartenberg que simplifica bastante a análise.

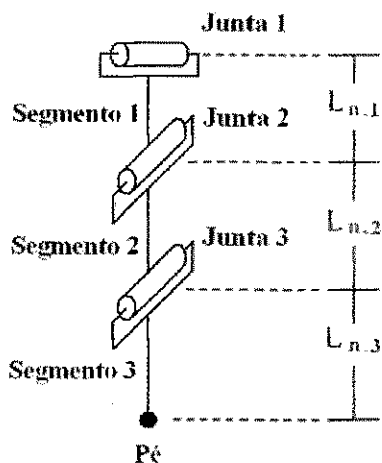


Fig4 Numeração dos seguimentos e das juntas

Primeiramente, os eixos $Z_{n,i}$ dos sistemas de coordenadas foram estabelecidos como os eixos de revolução de cada junta. No caso do efetuador, apenas se repetiu a orientação do eixo Z da junta 3. A base foi alocada sobre a junta 1, sendo seus eixos $X_{n,0}$ e $Y_{n,0}$ escolhidos arbitrariamente, mas respeitando a regra da mão direita. Uma vez fixada a base, vê-se que os eixos $Z_{n,0}$ e $Z_{n,1}$ não são coplanares. Então, o seguimento perpendicular a ambos os eixos, de $Z_{n,0}$ para $Z_{n,1}$, define o eixo $X_{n,1}$, e sua intersecção com o eixo $Z_{n,1}$ a origem $o_{n,1}$. Como os eixos $Z_{n,1}$ e $Z_{n,2}$ são paralelos, a origem $o_{n,2}$ pode ser colocada em qualquer ponto sobre $Z_{n,2}$, e o eixo $X_{n,2}$ estará na direção $o_{n,2}-Z_{n,1}$, não importando o sentido (de $o_{n,2}$ para $Z_{n,1}$ ou de $Z_{n,1}$ para $o_{n,2}$). Para efeito de simplificação, posicionou-se $o_{n,2}$ sobre a junta 3 e adotou-se $X_{n,2}$ no sentido de $Z_{n,1}$ para $o_{n,2}$. Finalmente, como os eixos $Z_{n,2}$ e $Z_{n,3}$ são paralelos, seguiu-se o mesmo procedimento para estabelecer o eixo $X_{n,3}$. Feito isso, os eixos $Y_{n,1}$, $Y_{n,2}$ e $Y_{n,3}$ foram arranjados de acordo com a regra da mão direita. O resultado final da alocação dos sistemas de coordenadas pode ser visto na figura:

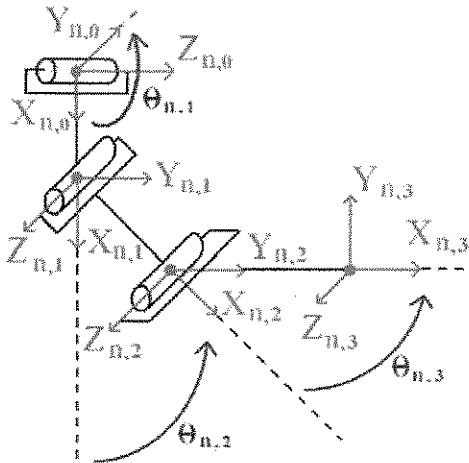


Fig5 Sentido positivo das variáveis das juntas

Cada pata possui todas as suas três juntas do tipo rotacional e consiste, portanto, em um mecanismo articulado. Sendo assim, o conjunto de variáveis das juntas da n-ésima pata é escrito como $q_n = [\theta_{n,1} \theta_{n,2} \theta_{n,3}]^T$ em que $\theta_{n,i}$ ($i = 1, 2, 3$) representa o ângulo de atuação na base da i-ésima junta para a n-ésima pata.

Os sentidos positivos de $\theta_{n,2}$ e $\theta_{n,3}$ corresponderão a rotações de forma a movimentar a pata para frente. No entanto, para as patas da lateral direita ($n = 1$ ou 4), o sentido positivo de $\theta_{n,1}$ corresponde a um movimento “para dentro” do robô, enquanto que, para as patas da lateral esquerda ($n = 2$ ou 3), o sentido positivo de $\theta_{n,1}$ corresponde a um movimento “para fora” do robô. A visualização desta convenção pode ser mais bem compreendida por meio da Figura abaixo:

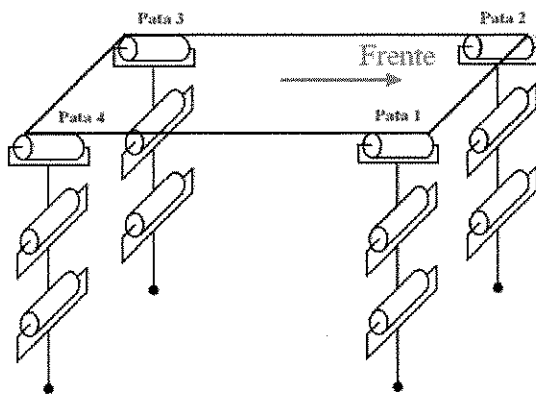


Fig 6 Numeração das Patas

8.6. O Modelo do Robô

Primeiramente, adotou-se um sistema de coordenadas no centro do corpo do robô, conforme ilustra a Figura abaixo. Este sistema será responsável por descrever a posição e a orientação do robô no espaço.

Sendo assim, observa-se que existe apenas uma rotação e uma translação de coordenadas entre os sistemas $X_{n,0} \times Y_{n,0} \times Z_{n,0}$ dos ombros e o sistema $X_c \times Y_c \times Z_c$ do corpo.

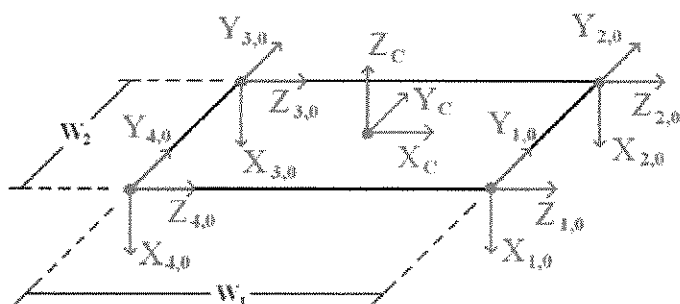


Fig7 Sistemas de coordenadas do corpo do robô

8.7. Geração da Trajetória

A locomoção de quadrúpedes é caracterizada por diversas seqüências periódicas de movimentos de suas patas. Dentre os principais modos de locomoção, pode-se citar a caminhada (ou passo), o trote, o galope, o meio-galope (ou galope curto) e a marcha. Essas diferentes formas de se locomover estão associadas à quantidade de patas em contato com o chão a cada intervalo de tempo e à velocidade de deslocamento do corpo, determinando dois tipos de estabilidade: estática e dinâmica.

À baixas velocidades, quando é levantada apenas uma pata de cada vez, lida-se com uma estabilidade estática na qual a projeção do centro de gravidade do robô no solo deve manter-se dentro do triângulo formado pelas três patas em contato com o chão.

À altas velocidades, quando as patas são levantadas duas a duas ou até mesmo quando todas elas perdem simultaneamente o contato com o solo por curtos períodos de tempo, lida-se com uma estabilidade dinâmica na qual forças inerciais ajudam na manutenção do equilíbrio.

Para fazer uma comparação, vai se tratar neste trabalho apenas da caminhada e da marcha. A caminhada é caracterizada pelo avanço de apenas uma pata por vez.

A pata que está no ar move-se livremente em um movimento chamado de balístico. As patas que estão em contato com o chão movem-se para trás em um movimento de recuo, dando o impulso necessário para a projeção frontal do corpo do robô.

Já a marcha é um tipo de “caminhada diagonal”, em que as quatro patas do quadrúpede se movem par a par, uma anterior com a respectiva posterior oposta, ou seja, a anterior direita avança simultaneamente com a posterior esquerda e vice-versa.

Gerador de trajetória

Um gerador de trajetória tem por objetivo definir, a cada instante de tempo discreto (k), um vetor que contém as coordenadas desejadas para os quatro pés. Portanto, o gerador é o responsável pelo planejamento da trajetória a ser seguida por eles, podendo usar qualquer lei de geração. Diante disso, enfatiza-se a necessidade de fazer uma distinção a cada instante de tempo entre os pés que estão em contato com o solo e os que estão em movimento balístico. Tendo em vista o modelamento dessa dinâmica de movimento, define-se a seguinte matriz :

$$\Gamma_{(k)} = \begin{pmatrix} \gamma_{1,(k)} & 0 & 0 & 0 \\ 0 & \gamma_{2,(k)} & 0 & 0 \\ 0 & 0 & \gamma_{3,(k)} & 0 \\ 0 & 0 & 0 & \gamma_{4,(k)} \end{pmatrix}$$

$\gamma_{n,(k)} = \{ 1$, se o pé da n -ésima pata estiver em contato com o solo no instante (k).

0 , caso contrário.

O propósito desta matriz $\Gamma(k)$ é que ela seja atualizada a cada instante (k) por meio de sensores de toque colocados nos pés do robô, caracterizando assim um sistema automático de atualização do modelo.

A trajetória de um pé é planejada da seguinte maneira, (ver figura abaixo):

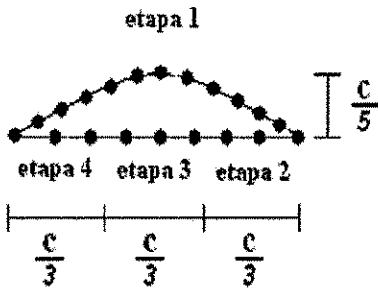


Fig8 Ciclo de caminhada

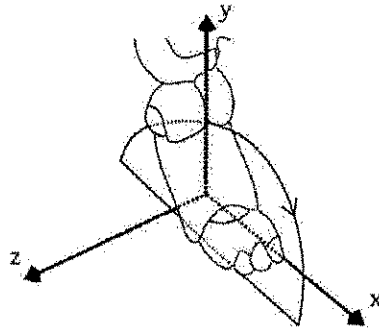


Fig9 Coordenadas da pata

*Para a pata que irá avançar, o pé descreve no ar uma trajetória senoidal discretizada:

$$\frac{c}{5} \text{sen} \left(\frac{k\pi}{N} \right), \quad k = 0, 1, \dots, N.$$

Na equação acima, c corresponde à distância entre a posição inicial do pé e o ponto final desejado, e N ao número de amostras utilizadas na discretização da trajetória.

Para as patas restantes, os pés descrevam uma trajetória retilínea em sentido contrário ao de locomoção do robô. Como os pés estão em contato com o chão, isto é, fixos, este movimento das patas fará com que o centro do robô se desloque para frente em relação a um referencial no solo. O comprimento desta trajetória retilínea pode ser c/3 (quando é levantada apenas uma pata de cada vez - caminhada) ou c (quando as patas são levantadas duas a duas - marcha).

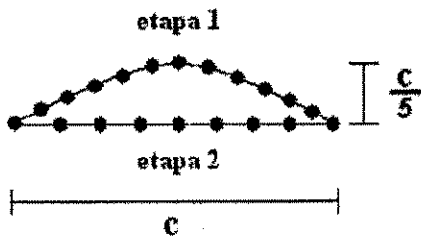


Fig10 Ciclo de marcha

8.8. Otimização do Movimento

O problema da otimização dos movimentos, resume-se à determinação das variações que os ângulos das juntas de cada pata deverão sofrer a cada instante de tempo, a fim de que os pés sigam a trajetória desejada.

Portanto, é necessário o desenvolvimento de um otimizador, que teria por objetivo encontrar o valor apropriado do ângulo de cada junta, de forma a minimizar os erros na trajetória, e conseqüentemente a energia gasta para cada atividade.

9. CONCLUSÕES E CONSIDERAÇÕES FINAIS

Devido à estrutura modular do hardware e do software podem-se adicionar sensores de vários tipos, como sensores de toque, velocidade, luminosidade, etc. de modo a se ter, a cada instante, informações do estado do sistema, ou seja, do robô, caracterizando assim um sistema automático de atualização do modelo.

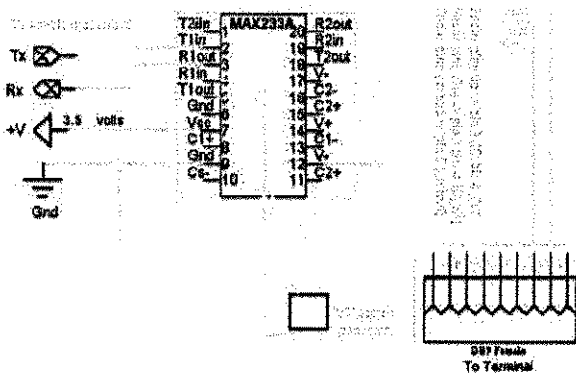
É necessário o desenvolvimento de um algoritmo de otimização para determinar o valor apropriado das variáveis do sistema-robô de forma a minimizar os erros na trajetória, no posicionamento, etc. e, enfim, no desempenho global do robô.

Espera-se com este estudo contribuir para a melhoria do sistema-robô i-cybie que vem se desenvolvendo aos poucos, porém, já mostrou amplas perspectivas e flexibilidade para inúmeras aplicações. Pode ser bastante útil como sistema educativo, assim como para finalidades diversas desde o entretenimento de crianças até como ferramenta na recuperação de doentes e idosos através da capacidade de comunicação e interatividade.

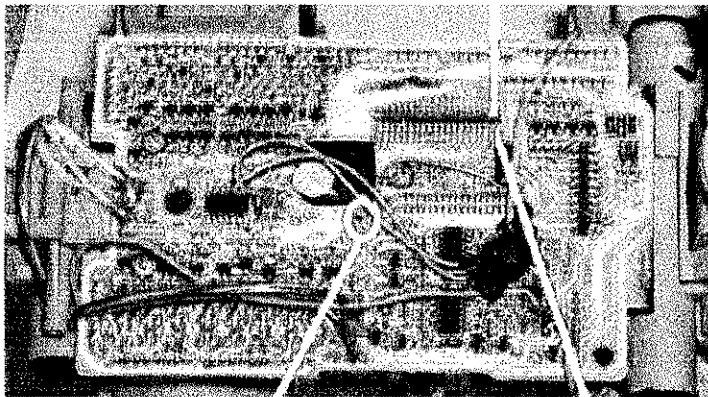
ANEXOS:

Modificações no i-cybie, Etapa 1:

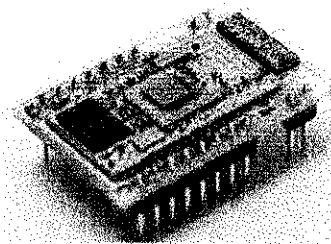
Esquema do circuito para a inserção do MAX 232, e comunicação do PC com o i-Cybie:



Detalhe de dentro do i-cybie, indicando onde soldar os componentes do circuito acima:



Modulo Bluetooth Wintec BTV42



A Pinagem do Wintec BTV42

