

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Trabalho Prático III

Introdução à Robótica

Alunos:
Dimas Dutra
Leandro Maia
Leandro Soriano
Luciano Borges

Belo Horizonte - MG

8 de novembro de 2007

Sumário

1	Introdução	2
2	Objetivos	2
3	Funcionamento	2
4	Tarefas	3
4.1	Tarefa 01	3
4.2	Tarefa 02	3
4.3	Tarefa 03	4
4.4	Tarefa 04	4
5	Dificuldades	5
6	Conclusão	5

1 Introdução

Esse TP visa a utilização de sistemas de controle para ajuste de posição e deslocamento do robô, que até o momento era realizado por temporizações obtidas empiricamente. Agora, o comportamento do robô quanto ao movimento se torna mais previsível.

2 Objetivos

Executar algoritmos de controle propostos no livro, incluindo posicionamento de roda e distância a obstáculo (seguir uma parede sem encostar)

Implementar o algoritmo de *WaveFront*, em que se dá uma pose objetivo para o robô, assim como localização de obstáculos, e ele deve encontrar um caminho e chegar lá.

3 Funcionamento

Os algoritmos estavam praticamente prontos, sendo só levemente adaptados. Como não temos bend-sensor, o robô "enxerga" a distância à parede com o sensor infravermelho, aproveitando de sua leitura variável com a distância ao objeto.

A rotina de leitura dos encoders (um em cada roda) é executada por instruções de hardware, através de um código em assembly, que nos retorna posição e velocidade de cada encoder. Estes foram anexados ao sistema de transmissão das rodas, através de uma engrenagem adicional e correntes, e oferecem uma relação de 40 pulsos por volta, que se mostrou uma boa velocidade.

O controle utilizado é proporcional-derivativo, isto é, dada uma distância, a potência entregue aos motores será proporcional tanto à diferença na posição quanto ao valor da velocidade.

Com o controle de posicionamento das rodas, além da translação passamos a controlar também a rotação. Por nosso robô ser um robô diferencial de duas rodas, seu giro sobre o próprio eixo (turn-in-place) é feito, como já se sabia anteriormente, girando as rodas igualmente em direções opostas. Como o eixo permanece fixo, as rodas percorrem uma distância igual ao arco correspondente ao ângulo de giro. Por exemplo, para girar 90 graus, basta comandar um deslocamento de $nR/2$, em que R é o raio de giro. As rotações, portanto também ficam bem precisas, o que auxilia muito o wavefront.

4 Tarefas

4.1 Tarefa 01

A primeira parte do exercício consiste simplesmente em adicionar um resistor variável em paralelo ao resistor de $47k\Omega$ existente na entrada do sensor na *Handboard*. A figura 3.51 representa o esquema de montagem do circuito. Foi utilizado um resistor que apresentou resistência máxima de $51,3k\Omega$.

Para evitar a criação de um caminho de baixa resistência, tivemos o cuidado de não ajustar o valor do resistor variável para menos que $1k\Omega$. Mesmo sendo pouco provável que o fototransistor deixasse passar toda a corrente, a precaução certamente evitaria que um curto circuito na *Handboard*.

Ao contrário do que esperávamos, a diminuição da resistência fez com que a leitura ficasse menos sensível, diminuindo cada vez mais a faixa de valores lidos.

A segunda parte do exercício consiste em calcular a corrente que circula no circuito supondo que um LED com queda de 2V, em série com um resistor de 330Ω estejam ligados à uma fonte de 5V.

Pelo princípio que não há perda no fio, se uma queda de 2V é devido ao LED, logo os outros 3V ficaram no resistor. Pela equação $V = Ri$, sendo $V = 3V$ e $R = 330\Omega$, temos que $i = 9,09 \times 10^{-3}A$.

Segundo o esquema da figura 6.27, e sabendo que o mesmo consegue fornecer até 100mA, podemos calcular o número de LED ligados em paralelo, seguindo o mesmo esquema, apenas dividindo a corrente máxima que pode ser fornecida pelo corrente consumida por cada conjunto LED-Resistor. Conforme calculado anteriormente, sabemos que cada conjunto consome 9,09mA, logo, $N = 100mA / 9,09mA = 11LEDs$.

4.2 Tarefa 02

O exercício consistia, basicamente, em implementar vários algoritmos para seguir parede. Como não possuíamos o *bend-sensor*, utilizamos o sensor de infravermelho para determinar a distância do robô à parede.

As versões do algoritmo *hard-turn* e *gentle-turn* permitem ao robô virar nas quinas da parede. A diferença entre o primeiro e o segundo é que como a virada é bem mais forte, ele consegue contornar as quinas de maneira mais correta, em compensação anda como "bebado" quando está em linha reta. Já o segundo não faz bem as curvas, mas anda em linha reta bem melhor. O problema no segundo algoritmo é que como a curva é feita de maneira bem suave, o raio fica muito grande, e conseqüentemente a curva não é realizada

da maneira adequada.

A terceira versão do algoritmo utiliza controle para determina a "força" com que o robô irá virar, de forma a melhorar a qualidade na reta e na quina. O resultado foi que o robô se comportou bem melhor, mas dá um tranco no início porque o *set point* está muito distante, conseqüentemente o erro também o será.

Já a quarta versão utilizou uma variação do anterior, porém o *set point* não está muito distante, acompanhando o robô a uma distância determinada. O resultado foi que o tranco inicial diminuiu, porém não houve muita variação no andar do robô. Outra característica importante é que se o robô passar do ponto determinado, o mesmo não volta.

Os gráficos resultantes das medidas realizadas rodando cada um dos quatro algoritmos é apresentada a seguir.

4.3 Tarefa 03

Esse exercício determinou a utilização de controle proporcional-derivativo de forma a melhorar a qualidade da realização das tarefas. Apesar de tentativa e erro não ser a melhor das opções, os valores de ganho e erro utilizados foram determinados desta maneira. Um problema encontrado, é que conforme os valores utilizados, o robô entrava em estado estacionário, isso porque a potência fornecida aos motores era insuficiente para que o mesmo diminuísse o erro, e como o erro não era integrado no tempo, o robô permanecia parado.

Segue o gráfico que representa o controle PD da roda:

O pico de velocidade observado no gráfico do controle da roda ocorre devido a termos atribuímos um valor na variável `encoderX_counts`. Isso representa uma alteração muito rápida na posição da roda que é interpretada como um pico de velocidade. Como em nosso experimento não utilizamos ação derivativa, isso não representou nenhum problema.

4.4 Tarefa 04

O algoritmo do *WaveFront* foi implementado e testado. As posições dos obstáculos e pontos de partida e chegada podem ser configurados em tempo de execução através do menu disponível na *HandBoard*.

5 Dificuldades

primeira grande dificuldade enfrentada nesse trabalho foi a aquisição dos arquivos binários .icb, que continham as rotinas de monitoração dos encoders.

Quanto aos sensores, enfrentamos algo inesperado: o robô passa a ter comportamento seguindo uma parede diferente baseado na sua textura, já que agora é um reflexo de luz que o informa a posição.

O controle de rotação também enfrentou certos problemas pois o raio de giro calculado (metade da distância entre as rodas) se mostrou bem distinto do que realmente foi usado para se surtir o efeito esperado.

Os problemas não pararam por aqui! Os algoritmos de controle de posicionamento da roda foram projetados para encoders de quadratura, entretanto, nós só dispomos de um encoder para cada roda. O problema com isso é que quando a posição ultrapassa a referência, e a contagem dos pulsos não volta, passamos a ter realimentação positiva na entrada! (em outras palavras, se o robô passar do ponto não pára e volta, mas tenta chegar pelo outro lado do planeta - a bateria não duraria) A solução para isso foi impedir o overshoot, basicamente parando o robô se ele ultrapassar a referência. Para o controle continuar satisfatório, ajustamos os ganhos de modo que o robô não chegue muito rápido na referência, não havendo um grande erro ao ser desligado.

6 Conclusão

Felizmente, apesar de muita dificuldade, o robô parece realizar as tarefas propostas. Os algoritmos, apesar de grande parte prontos, requiseram atenção e entendimento, de forma que pudessem ser modificados para realizar os exercícios propostos.

Um resultado importante deste trabalho é como a utilização de métodos de controle auxiliam o comportamento mais adequado do robô, principalmente se comparado à técnicas de contagem de tempo