

OTIMIZAÇÃO DE TRAJETÓRIAS ATRAVÉS DE CAMINHOS MÍNIMOS PARA A LOCOMOÇÃO DE CADEIRA DE RODAS ROBÓTICA

HENRIQUE DO CARMO CAMPANELI¹, MÁRIO MESTRIA².

1. *Graduando no Curso Superior de Engenharia Elétrica, IFES (Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo), Campus Vitória*

E-mails: henriq.car.cam@hotmail.com, hcampaneli@gmail.com

2. *Professor da Coordenadoria de Engenharia Elétrica, IFES (Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo), Campus Vitória*

Av. Vitória, 1729 – Jucutuquara – Vitória – ES, CEP: 29040-780

E-mails: mmestria@ifes.edu.br, mmestria@uol.com.br

Abstract— This paper consists in finding the possible shortest paths, to be travelled by a robotic wheelchair, between any two accessible points of the physical environment, using an optimization algorithm. After setting the path, the algorithm reports the features such as angles and distances performed during route. These paths allow finding trajectories for the robotic wheelchairs that will be activated through the user's interface and communicating its interface with an embedded system. Thus, the wheelchair user can choose from the points set by the user's interface and move around in this environment. The results of this paper contribute to the improvement in the quality of life of people who have disabilities, the wheelchair users, and to the advances in assistive robotics.

Keywords— Assistive robotics, Embedded intelligent systems, Robotic wheelchair, Minimal paths.

Resumo— Este trabalho consiste em encontrar os caminhos menores possíveis, a serem percorridos por uma cadeira de roda robótica, entre quaisquer dois pontos acessíveis do ambiente físico, utilizando um algoritmo de otimização. Após definir o caminho, o algoritmo informa as características como ângulos e distâncias realizadas durante percurso. Estes caminhos permitem encontrar trajetórias para as cadeiras de rodas robótica, que serão acionados através da interface de usuário e comunicando sua interface com um sistema embarcado. Assim, o usuário de cadeira de roda poderá escolher dentre os pontos definidos pela interface de usuário e locomover-se neste ambiente. Os resultados deste artigo contribuem para a melhoria na qualidade de vida das pessoas portadoras de necessidades especiais, os usuários de cadeira de rodas, e para os avanços na robótica assistiva.

Palavras-chave— Robótica assistiva, Sistemas inteligentes embarcados, Cadeira de rodas robótica, Caminhos mínimos.

1 Introdução

As pesquisas em robótica assistiva têm contribuído para a melhoria na qualidade de vida de pessoas portadoras de necessidades especiais, principalmente os usuários de cadeira de rodas.

Em (Bemelmans; Gelderblom; Jonker; Witte, 2012) é sublinhado a importância dos robôs na área de assistência social, destacando a contribuição da robótica na racionalização, manutenção, ou até mesmo na melhoria da qualidade de vida de pessoas. Robôs podem contribuir para cuidados de saúde em termos de capacidade e de qualidade realizando de forma muito precisa determinada tarefa. Os robôs podem auxiliar nas tarefas das pessoas aumentando a sensação de autonomia e autogestão.

O projeto MOBI (Mobilidade Inteligente) (Lima; Andreão, 2010) está inserido na linha de robótica assistiva ((Allin; Eckel; Markham; Brewer, 2010), (Meng; Lee, 2006)) e visa à locomoção autônoma de um usuário de cadeira de rodas (cadeirante) dentro de um ambiente acessível e previamente mapeado, sem que haja colisão com

obstáculos. Trata-se de um trabalho de desenvolvimento de pesquisa interdisciplinar que busca impulsionar os avanços tecnológicos na área e promover a inserção dos portadores de necessidades especiais na sociedade.

Independentemente da idade, a mobilidade é uma característica fundamental para todo ser humano (Faria; Vasconcelos; Reis; Lau, 2012).

Soluções de navegação autônoma de robôs móveis em ambientes abertos e fechados já foram apresentadas em (Matsumoto; Komoriya; Hatase; Nishimura, 2006) e em (Hsiao; Lee; Shen; Lai, 2004), assim como, as soluções aplicadas às cadeiras de rodas ((Amaral; Garcia; Bastos; Mazzo, 2009), (Bastos; Ferreira; Celeste; Cavalieri; Sarcinelli; Casaño; Perez; Soria; Kumar, 2007)).

As soluções apresentadas pela literatura consideram em princípio a definição prévia das condições de contorno, tais como origem, destino e tempo de execução, para as trajetórias a serem percorridas e estabelecidas pelo usuário ou por sistemas de mapeamento e localização externos ao sistema. Nota-se na literatura que o tratamento dado à robótica móvel, principalmente a cadeira de rodas, não contempla o controle de sistema embarcado embutido ao robô móvel.

Ao incorporar a robótica móvel, é necessário o uso de sensores externos eficientes, modelos dinâmicos realísticos e técnicas de controle sofisticadas para permitir que o robô possa interagir com eficácia (Coelho; Nunes, 2003). A cadeira de rodas estudada neste trabalho será dotada de controle embarcado com capacidade de tratamento de sinais extraídos do usuário, por exemplo, a voz ou entrada via teclado, possibilitando se deslocar sem a necessidade de condução de pessoas externa.

Para economizar tempo e custos de energia com a locomoção da cadeira robótica, é necessário o uso de um sistema de navegação que consegue obter o menor caminho com segurança entre duas localizações (Cherubini; Oriolo; Macrì; Aloise; Babiloni; Cincotti, 2007). Este percurso deverá ser incorporado no sistema de controle da cadeira robótica. Um método utilizado para este fim é a aplicação do Algoritmo de Dijkstra (Dijkstra, 1959), que através de um grafo ponderado, consegue encontrar a distância mínima entre um ponto de origem e um de destino, escolhidos pelo cadeirante.

Neste trabalho foi desenvolvido um programa que permite ao usuário identificar um destino e o sistema se encarrega de gerar uma trajetória para alcançá-lo. Para isto são definidos diversos pontos mapeados no ambiente por onde o usuário de cadeira de roda transitará. A trajetória traçada será aquela que permite a otimização de caminhos entre dois pontos e neste caso será a de menor distância. A partir disto, um movimento se iniciará na cadeira de roda robótica através de um motor comandado por um sistema embarcado.

Dada as características do espaço físico, também foi possível calcular, com o uso de um algoritmo, os ângulos de giro e as devidas distâncias entre dois pontos. Com essas informações, o sistema de controle, levando em consideração o perfil de velocidade linear e a orientação da cadeira, pode aplicar a geração da trajetória desejada. Neste sentido, não há necessidade de auxílio de pessoas no ambiente por onde será transitada a cadeira de roda robótica.

O restante do artigo é organizado da seguinte forma. A seção 2 descreve a cadeira de rodas robótica. A seção 3 apresenta a metodologia utilizada para realizar as trajetórias do usuário de cadeira de roda robótica, que necessita se deslocar entre quaisquer dois pontos. A seção 4 contém alguns exemplos mostrando a execução do algoritmo. A seção 5 mostra as discussões sobre o artigo. Finalmente as conclusões são descritas na última seção.

2 Cadeira de Rodas Robótica

O programa desenvolvido neste trabalho, através do algoritmo, fornecerá ao sistema embarcado da cadeira de rodas robótica as informações necessárias para definir as trajetórias entre quaisquer dois pontos escolhidos. O circuito digital mostrado na Figura 1 é responsável por todo processamento de sinais da cadeira.



Figura 1 - Circuito digital acoplado na cadeira de rodas

Para a detecção de obstáculos, próximos a cadeira de rodas, foi utilizado um sensor de ultrassom localizado na parte de frente da cadeira, no braço direito, como mostra a Figura 2.



Figura 2 - Sensor de ultrassom para detecção de obstáculos

Ao detectar o obstáculo, o sensor de ultrassom envia o sinal para o circuito digital, Figura 1, que comanda a cadeira de rodas, e então a cadeira para de movimentar-se. Para a execução dos movimentos foram utilizados dois motores, (veja Figura 3), um para cada roda da cadeira, que são acionados pelo circuito digital mostrado anteriormente.



Figura 3 - Motores instalados na cadeira de rodas

As Figuras 4 a 6 mostram diversas vistas da cadeira de rodas robótica. A Figura 4 mostra a parte frontal da cadeira de rodas, a Figura 5 a parte traseira e a Figura 6 a vista lateral.



Figura 4 - Vista frontal da cadeira de rodas

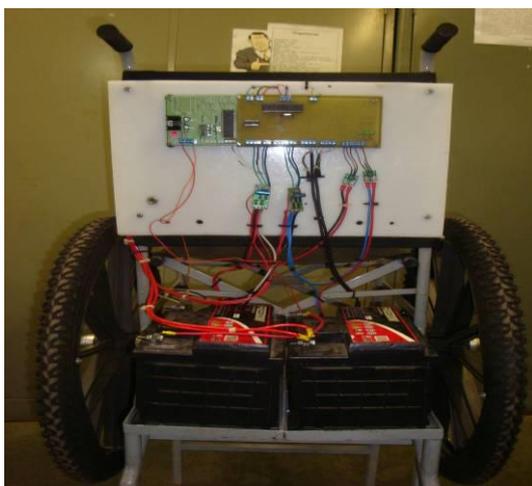


Figura 5 - Vista traseira da cadeira de rodas



Figura 6 - Vista lateral da cadeira de rodas

3 Metodologia

Como ambiente de trabalho para este projeto adotou-se o Campus Vitória do Instituto Federal de Espírito Santo (IFES). Com a obtenção das plantas baixas de escala 1:250 (pavimento inferior e pavimento superior) deste câmpus, foram mapeados diversos pontos como laboratórios, salas, elevador, cantina, biblioteca, dentre outros, com suas respectivas coordenadas cartesianas para a utilização do Algoritmo de Dijkstra e desenvolvimento do programa.

Além disto, alguns lugares mostrados nas plantas não apresentavam de maneira correta os espaços físicos como pilastras, paredes e salas desativadas. Nestes casos realizamos os mapeamentos dos pontos coletando em campo as informações destes espaços físicos através de medições e uso de trena.

Usando o *software* Labwindows/CVI (National Instruments, 2012), foi implementado um programa que coleta todas as informações dos pontos (número, local e coordenadas x e y) em arquivo texto, e gera as devidas adjacências entre esses pontos. Dessa forma, sabendo as coordenadas x e y de cada ponto calcula-se o ângulo obtido com a equação 1 entre um ponto e seu adjacente, sucessivamente, desde a origem até o destino. Estas informações junto aos valores das respectivas distâncias obtidas com a equação 2 e ângulos de giro devem ser executadas pela cadeira de rodas robótica.

$$\theta = \arctg\left(\frac{y_1 - y_2}{x_1 - x_2}\right) \quad (1)$$

$$D_{1,2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

Explicação do Algoritmo Dijkstra:

Dado um grafo ponderado $G(V, A)$, com valores não negativos, onde V é um conjunto finito e não vazio de vértices e A é um conjunto finito de

arestas, ou seja, segmento de retas, que interligam dois vértices, escolhe-se um vértice contido neste grafo como raiz da busca (a origem). Este algoritmo calcula o caminho mínimo deste vértice até o vértice destino, passando pelos demais vértices intermediários de G .

O algoritmo parte de uma estimativa inicial para o menor caminho, e vai sucessivamente ajustando esta medida, até que todos os vértices do grafo tenham sido analisados. Dado que o sistema de coordenadas cartesianas utilizado neste trabalho não introduziu arcos ou ramos com pesos negativos, o algoritmo garante de forma exata a solução de menor caminho e tem desempenho melhor do que realizar buscas exaustivas ou aleatórias, dentre todas as possíveis soluções, devido aos vértices intermediários.

A complexidade do Algoritmo de Dijkstra é da ordem de $O((m+n) \cdot \log n)$, onde m é o número de arestas e n é o número de vértices, que comparado ao Algoritmo de Ford-Moore-Bellman (Bellman, 1958), no qual calcula também caminhos mínimos, possui complexidade igual a $O(n \cdot m)$.

Procedimentos do algoritmo:

Dado o grafo G dirigido ou não, com pesos não negativos em cada arco ou ramo, o algoritmo encontra a distância mínima $dist(v_j)$ entre um vértice inicial (v_i) e um vértice final (v_j). Seja v_i um vértice de V e W um conjunto que contém os vértices cuja distância a v_i é mínima e conhecida no momento da execução do algoritmo.

1. Considera-se a distância ao próprio v_i como zero, $dist(v_i)=0$.
2. Utiliza-se $dist(v_k)$ como uma estrutura que armazena e referencia a distância de v_i ao vértice v_k .
3. Para todos os vértices (v_k), considera-se a sua distância a v_i como valendo LIMITE (valor alto que descarta a possibilidade de caminho mínimo), $dist(v_k)=LIMITE$, para todo v_k em $V \setminus \{v_i\}$.
4. Qualquer que seja v_k não pertencente a W , tal que v_j foi o último vértice a entrar em W e v_k pertencente ao conjunto V , se $dist(v_k) > (dist(v_j) + \text{distância de } v_j \text{ a } v_k)$, então $dist(v_k) = (dist(v_j) + \text{distância de } v_j \text{ a } v_k)$.
5. Determina-se, qualquer que seja v_k não pertencente a W , o vértice que representa o menor valor de distância dentre $dist(v_k)$, sendo v_m esse vértice.
6. Faça $v_k = v_m$, $dist(v_k) = dist(v_m)$ e v_k passa a ser o novo elemento de W , sendo $v_j = v_k$.
7. Se o vértice v_k coincidir com o vértice final (v_j), então $dist(v_j)$ é a menor distância entre v_i e v_j . Então, a execução do algoritmo termina. Senão, volta-se ao passo 4.

4 Resultados

Na execução do programa, o usuário consegue ler uma lista dos locais de interesse enumerados de 0 a 156 do Campus Vitória, apresentada no seguinte formato: contém o número e o nome do local associado ao número. A Figura 7 ilustra a interface de escolha dos locais de interesse para o usuário.

Numero	Local
0	- Auditorio
1	- Laboratorio de Informatica 2
2	- Laboratorio de Informatica 4
3	- Laboratorio de Informatica 5
4	- Laboratorio de Informatica 6
5	- Escada Superior
6	- Laboratorio de Informatica 7
7	- Elevador
8	- Laboratorio de Informatica 8
	.
	.
	.

Figura 7 - Lista dos locais de interesse do usuário

O usuário, então, escolhe os pontos de origem e de destino, com a opção de escrever o nome do local sem diferenciação de maiúsculas e minúsculas, porém não aceitando acentos e “ç”. Exemplo, “Biblioteca” ou escolher o número referente ao local (“145”) que é o número da lista referente à Biblioteca. A seguir, serão mostrados alguns exemplos da execução do programa.

Exemplo 1:

O usuário digitou “Coordenadoria de Eletrotecnica” (número “137”) como origem e a “Cantina” (número “151”) como destino, conforme mostrado na Figura 8.

```

Digite o Numero ou Local de origem: Coordenadoria de Eletrotecnica
Digite o Numero ou Local de destino: Cantina

*****
Menor caminho entre 137 e 151:
137 -> 136 -> 154 -> 152 -> 150 -> 151

Distancias:
137 -> 136 = 1.938 metros (Girou: 0.000 graus) Direcao:270.000 graus
136 -> 154 = 9.160 metros (Girou: 90.000 graus) Direcao: 0.000 graus
154 -> 152 = 1.600 metros (Girou: -90.000 graus) Direcao:270.000 graus
152 -> 150 = 9.443 metros (Girou: 7.010 graus) Direcao:277.010 graus
150 -> 151 = 34.062 metros (Girou: 82.990 graus) Direcao: 0.000 graus

Distancia Total: 56.203 metros

*****

```

Figura 8 - Programa executado para o Exemplo 1

Exemplo 2:

Neste exemplo, o usuário escolheu “Caixa Economica Federal” (número 150) como origem e “Diretoria” (número 87) como destino, (veja Figura 9).

```
*****
Menor caminho entre 150 e 87:
150 -> 146 -> 142 -> 140 -> 99 -> 97 -> 82 -> 83 -> 84 -> 85 -> 86 -> 87

Distancias:
150 -> 146 = 23.215 metros (Girou: 0.000 graus) Direcao:270.000 graus
146 -> 142 = 1.500 metros (Girou: 0.000 graus) Direcao:270.000 graus
142 -> 140 = 25.500 metros (Girou: 0.000 graus) Direcao:270.000 graus
140 -> 99 = 8.274 metros (Girou: 4.332 graus) Direcao:274.333 graus
99 -> 97 = 11.292 metros (Girou: -7.506 graus) Direcao:266.827 graus
97 -> 82 = 6.360 metros (Girou: 3.173 graus) Direcao:270.000 graus
82 -> 83 = 7.938 metros (Girou: 90.000 graus) Direcao: 0.000 graus
83 -> 84 = 9.375 metros (Girou: 0.000 graus) Direcao: 0.000 graus
84 -> 85 = 9.250 metros (Girou: 0.000 graus) Direcao: 0.000 graus
85 -> 86 = 9.375 metros (Girou: 0.000 graus) Direcao: 0.000 graus
86 -> 87 = 7.750 metros (Girou: 0.000 graus) Direcao: 0.000 graus

Distancia Total: 119.828 metros
*****
```

Figura 9 - Programa executado para o Exemplo 2

Exemplo 3:

Neste outro exemplo, a escolha do usuário foi: “Salao de Convencoes” (número 13) como origem e “Gremio Estudantil” (número 15) como destino. A Figura 10 mostra este caso escolhido pelo usuário.

```
*****
Menor caminho entre 13 e 15:
13 -> 12 -> 11 -> 10 -> 9 -> 8 -> 7 -> 6 -> 5 -> 14 -> 15

Distancias:
13 -> 12 = 12.250 metros (Girou: 0.000 graus) Direcao:180.000 graus
12 -> 11 = 4.000 metros (Girou: 90.000 graus) Direcao:270.000 graus
11 -> 10 = 10.003 metros (Girou: -88.568 graus) Direcao:181.432 graus
10 -> 9 = 4.000 metros (Girou: -1.432 graus) Direcao:180.000 graus
9 -> 8 = 5.500 metros (Girou: 0.000 graus) Direcao:180.000 graus
8 -> 7 = 6.625 metros (Girou: 0.000 graus) Direcao:180.000 graus
7 -> 6 = 3.000 metros (Girou: 0.000 graus) Direcao:180.000 graus
6 -> 5 = 8.000 metros (Girou: 0.000 graus) Direcao:180.000 graus
5 -> 14 = 10.625 metros (Girou: -90.000 graus) Direcao: 90.000 graus
14 -> 15 = 9.008 metros (Girou: 2.386 graus) Direcao: 92.386 graus

Distancia Total: 73.011 metros
*****
```

Figura 10 - Programa executado para o Exemplo 3

Exemplo 4:

Por fim, neste último exemplo a escolha do usuário foi: “Diretoria” (número 67) como origem e “Salao de Convencoes” (número 13) como destino, como mostra a Figura 11.

Neste último exemplo, o caminho traçado envolve os dois andares do Campus Vitória com a utilização do elevador, que são os pontos 73 (pavimento) e 7 (segundo andar). Assim, no momento em que passa pelo vértice 73 e depois vai para o vértice 7, na verdade, a cadeira de rodas está

apenas subindo o elevador. Esse elevador, neste trabalho, é a única conexão entre os dois andares.

```
*****
Menor caminho entre 67 e 13:
67 -> 68 -> 69 -> 71 -> 72 -> 73 -> 7 -> 8 -> 9 -> 10 -> 11 -> 12 -> 13

Distancias:
67 -> 68 = 9.000 metros (Girou: 0.000 graus) Direcao: 0.000 graus
68 -> 69 = 13.250 metros (Girou: 0.000 graus) Direcao: 0.000 graus
69 -> 71 = 11.750 metros (Girou: 0.000 graus) Direcao: 0.000 graus
71 -> 72 = 5.250 metros (Girou: 0.000 graus) Direcao: 0.000 graus
72 -> 73 = 5.750 metros (Girou: 0.000 graus) Direcao: 0.000 graus
73 -> 7 = 0.146 metros (Girou: -149.036 graus) Direcao:210.964 graus
7 -> 8 = 6.625 metros (Girou: 149.036 graus) Direcao: 0.000 graus
8 -> 9 = 5.500 metros (Girou: 0.000 graus) Direcao: 0.000 graus
9 -> 10 = 4.000 metros (Girou: 0.000 graus) Direcao: 0.000 graus
10 -> 11 = 10.003 metros (Girou: 1.432 graus) Direcao: 1.432 graus
11 -> 12 = 4.000 metros (Girou: 88.568 graus) Direcao: 90.000 graus
12 -> 13 = 12.250 metros (Girou: -90.000 graus) Direcao: 0.000 graus

Distancia Total: 87.524 metros
*****
```

Figura 11 - Programa executado para o Exemplo 4

5 Discussões

Através dos pontos mapeados na planta do Campus Vitória no IFES e o uso do Algoritmo de Dijkstra foram possíveis traçar as trajetórias entre quaisquer dois pontos acessíveis escolhidos pelo usuário de cadeira de rodas, validando o método utilizado.

As localidades envolvidas no trabalho foram escolhidas de tal forma que a cadeira de rodas não apresente dificuldades na locomoção, ou seja, foram considerados pavimentos planos e rampas com baixa inclinação, sem irregularidades.

Como foi mostrado nos exemplos das Figuras 8 a 11, o programa informa os pontos que a cadeira de rodas robótica deve percorrer permitindo calcular a distância de um ponto a outro, os graus que a cadeira deve girar ao passar de um ponto para outro, a direção do percurso e por fim a distância total percorrida. Essas informações adquiridas com a execução do programa servirão então de apoio para a realização das tarefas com o sistema de controle embarcado.

Deve ser levada em consideração a possibilidade de obstrução do trajeto por obstáculos, tais como, estandes, bancos, pessoas, dentre outros. Assim, foi exigido da cadeira de rodas robótica um tratamento sensorial sofisticado para a detecção de obstáculos nessas ocasiões, garantindo a segurança na locomoção.

Vale ressaltar também que a imprecisão nos valores de distâncias e ângulos adquiridos, devido às medições nas plantas baixas com régua milimétrica, pode gerar erros, mesmo que pequenos na execução do deslocamento da cadeira. No entanto, isto não prejudica as trajetórias da cadeira de rodas robótica.

6 Conclusões

Este artigo contribui para a área de robótica assistiva na locomoção autônoma de usuários de cadeira de rodas. A pesquisa está relacionada ao planejamento e geração de trajetórias em um ambiente previamente definido e devidamente mapeado, podendo ser aplicado em outros ambientes abertos ou fechados (uso domiciliar, por exemplo), onde seja possível a movimentação da cadeira de rodas robótica em questão.

Nosso artigo está alinhado com diversas pesquisas em diferentes áreas de atuação (como por exemplo, processamento de sinais) que, em conjunto, buscam intensificar o desenvolvimento da tecnologia assistiva, proporcionando a melhoria da qualidade de vida das pessoas portadoras de deficiência física.

Agradecimentos

Este trabalho de iniciação científica teve apoio parcial da FAPES (Fundação de Amparo à Pesquisa do Espírito Santo), proposta de número 0112130709554374-137, Edital PIBIC/PIVIC/PIBITI/PIVITI 01/2012 - CNPQ/IFES/FACITEC/FAPES do IFES (Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo). Os autores agradecem ao professor Rodrigo Varejão Andreão pelas sugestões e correções neste artigo.

Referências Bibliográficas

- Allin, S.; Eckel, E.; Markham, H.; Brewer, B. R. (2010). Recent Trends in the Development and Evaluation of Assistive Robotic Manipulation Devices. *Phys. Med. Rehabil. Clin. N. Am.* n. 21, p. 59-77.
- Amaral, P.F.S.; Garcia, J.C.G.; Bastos Filho, T.F.; Mazzo, M. (2009). Sistema de Navegação em Ambientes Fechados para uma Cadeira de Rodas Usando um Servidor de Arquivos XML com Informações de Acessibilidade. In *SBAI 2009 – IX Simpósio Brasileiro de Automação Inteligente, 2009, Brasília. Anais do IX Simpósio Brasileiro de Automação Inteligente.*
- Bastos Filho, T. F.; Ferreira, A.; Celeste, W. C.; Cavalieri, D. C.; Sarcinelli Filho, M.; Casaño, C. de L. C.; Perez, E.; Soria, C.; Kumar, D. K. (2009). An Intelligent Multimodal Robotic Wheelchair with Communication Capabilities. In *III International Congress on Digital Homes, Robotics and Telecare for All. Barcelona.*
- Bellman, R. (1958). On a Routing Problem. *Quarterly of Applied Mathematics*, v. 16, p. 87-90.
- Bemelmans R.; Gelderblom G. J.; Jonker P.; de Witte, L. (2012). Socially Assistive Robots in Elderly Care: A Systematic Review into Effects and Effectiveness. *JAMDA*, n. 13, p. 114-120.
- Cherubini A.; Oriolo, G.; Macrì, F.; Aloise, F.; Babiloni, F.; Cincotti, F.; Mattia, D. (2007). Development of a multimode navigation system for an assistive robotics project. *IEEE International Conference on Robotics and Automation*, Roma, p. 2336-2342.
- Coelho, P. M. M.; Nunes, U. J. C. (2003). Dinâmica de Robôs Móveis: Cadeira de Rodas Robotizada. *Ingenium*, v. 2, n. 77, p. 70-73.
- Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, vol. 1, n. 1, p. 269-271.
- Faria, B. M.; Vasconcelos, S.; Reis, L. P.; Lau, N. (2012). A Methodology for Creating Intelligent Wheelchair Users' Profiles. In J. Filipe e A. L. N. Fred (eds.), *ICAART 1, SciTePress*. p. 171-179.
- Hsiao, C.H.; Lee, M.L.; Shen, Y.C.; Lai, F. (2004). A design of small-area automatic wheelchair, *IEEE International Conference on Networking, Sensing and Control*, v.2, p. 1341- 1345.
- Lima, L. E. M.; Andreão, R. V. (2010). MOBI - Mobilidade Inteligente. Projeto de Pesquisa e Desenvolvimento da FAPES. Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, Vitória.
- Matsumoto, O.; Komoriya, K.; Hatase, T.; Nishimura, H. (2006). Autonomous Traveling Control of the "TAO Aicle" Intelligent Wheelchair. *International Conference on Intelligent Robots and Systems, IEEE/RS*, p. 4322-4327.
- Meng, Q.; Lee, M.H. (2006). Design issues for assistive robotics for the elderly. *Advanced Engineering Informatics*, n. 20, p. 171-186.
- National Instruments. *LabWindows/CVI*. (2012). Disponível em: <<http://sine.ni.com/np/app/main/p/docid/nav-110/lang/pt/>>. Acesso em: 01 set. 2012.