



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Autorizada pelo Decreto Federal nº 77.496 de 27/04/76

Recredenciamento pelo Decreto nº 17.228 de 25/11/2016

PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2021

Implementação de um Sistema de Navegação para um Robô Bípede Utilizando ROS

Gustavo Mendes Bastos¹; Anfranserai Morais Dias²

1. Estagiário PEVIC, Graduando em Engenharia da Computação, Universidade Estadual de Feira de Santana, e-mail: gustavomendes.fsa@gmail.com
2. Orientador, Departamento de nome, Universidade Estadual de Feira de Santana, e-mail: beltrano@provedor.br

PALAVRAS-CHAVE: ROS; ZMP

INTRODUÇÃO

Diferente de um robô de esteira ou que utiliza rodas, um robô bípede possui uma habilidade para andar em terrenos irregulares. Entretanto a locomoção de um robô bípede requer uma atenção maior já que eles tendem a tombar e cair. Na tentativa de solucionar esse problema alguns estudos adotam o método do Ponto de Zero Momento (ZMP - *Zero Moment Point*).

No artigo publicado “*Biped Walking Pattern Generator allowing Auxiliary ZMP Control*” é proposto um modelo de controle baseado na utilização de um Ponto de Zero Momento (ZMP) [1]. De acordo com os autores um modelo de controle ZMP simples possui uma perda de flexibilidade comparado a outros padrões como “*passive dynamic walk*”, “*hybrid zero dynamics*” e o “*central pattern generator*”. Neste trabalho foi proposto o uso de um ZMP auxiliar, permitindo assim a absorção de forças externas, adaptação às mudanças de elevação no terreno e controle de postura. O caminhar é derivado de um modelo de pêndulo invertido. Nessa estratégia a dinâmica no plano sagital (movimento na direção X) pode ser aproximada por um robô bípede possuindo carrinhos em vez de pés, já que o movimento do carrinho pode representar tanto a mudança de peso de um pé para o outro, quanto o torque nos pés.

No estudo realizado por Qiang Huang [2], é descrito o caminhar de um robô bípede, que possui apenas pernas e um tronco, como um movimento periódico que pode ser dividido em dois momentos. Um deles é descrito por uma fase de suporte duplo, o outro é uma fase de suporte único. A fase de suporte duplo começa quando o calcanhar da perna que está a frente toca o chão, e termina quando os dedos da perna de trás deixam o chão. Este artigo também descreve o deslocamento a partir do movimento dos pés e da cintura. Uma vez que esses valores são encontrados, as posições das juntas são obtidas através do cálculo da cinemática inversa.

Ill-Woo Park diz que devido a defeitos como o desnível do piso, erros de leitura e um modelo do robô imperfeito, o ZMP real de um bípede é diferente do ZMP desejado, sendo necessário o controle de movimento e balanço com base no feedback de sensores. Então ele propõe uma estratégia de geração de um ciclo de caminhada com base em um sensor F/T (Force/Torque) [3]. Este trabalho propõe a divisão do corpo do robô em duas partes. A primeira composta por perna e tornozelo, e a segunda definida pela pélvis. Além disso, são determinados três pontos de movimento: o tornozelo esquerdo, o tornozelo direito e a pélvis. Esses pontos são as partes que irão se movimentar. Para realizar os cálculos de movimento foram adotados dois sistemas de coordenadas. O primeiro tem origem na pélvis. O segundo é posicionado no espaço com a origem na posição inicial do movimento. Com esses sistemas de coordenadas são montadas equações para o movimento no plano sagital, e equações para o movimento no plano coronal (movimento no eixo y).

Esse resumo tenta descrever brevemente como foi realizado a tentativa de implementar um sistema de navegação para um robô bípede.

MATERIAL E MÉTODOS OU METODOLOGIA

Durante a realização das atividades foram utilizados os seguintes recursos:

- PIC 32 – um microcontrolador utilizado para processar todos os dados de sensores do robô e manter a comunicação entre o computador de bordo e os servos;
- Arduíno – Uma placa que possui um microcontrolador que foi projetada para ser facilmente programável;
- ROS – Sistema operacional utilizado para o desenvolvimento de robôs, utilizado para fazer o controle da troca de informações dos sensores e a central.

O robô bípede disponível para a realização desse trabalho é um robô bípede com 6 graus de liberdade em cada perna, semelhante ao ilustrado na figura abaixo. Para controlar os servo-motores e realizar a leitura de parte dos sensores foi usado um microcontrolador PIC32MX270F256B.

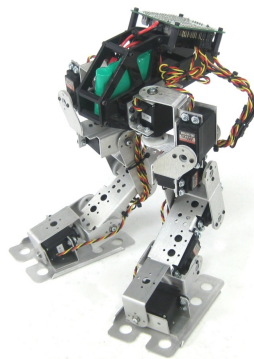


Figura 1: Robô bípede (Fonte: Lynxmotion)

Para programar o microcontrolador, inicialmente foi proposto o uso da plataforma disponibilizada pela fabricante, a MPLAB IDE. Porém devido a problemas que serão discutidos mais à frente essa ideia foi descartada.

Para realizar a implementação de um sistema de navegação de um bípede é necessário que ele possua um padrão para o seu caminhar. Para elaborar esse padrão foi feito um levantamento bibliográfico sobre implementações de rotinas de caminhada.

Além do padrão é necessário realizar um controle de sensores para detectar a direção que o robô está indo ou se ele está tombando ou não, para realizar esse controle dos sensores é utilizado o *Robotic Operatinal System* (ROS) que é uma plataforma que possui ferramentas para facilitar a criação e controle de um robô. O ROS conta com diversas ferramentas como o software de simulação Gazebo que foi utilizado para realizar as simulações do robô durante esse trabalho. Para realizar a simulação do robô bípede foi necessário encontrar um modelo 3D para a utilização no simulador. Inicialmente foi procurado um modelo similar ao modelo utilizado mas, dos encontrados, não foi possível o uso de nenhum deles. No fim, foi decidido usar o modelo do robô NAO. Esse modelo foi desenvolvido pela Aldebaran, e possui suporte para a utilização tanto do sistema ROS quanto para o Gazebo.



Figura 2: Robô NAO. (Fonte: ROS WIKI)

RESULTADOS E/OU DISCUSSÃO

O requisito básico para o desenvolvimento do plano de trabalho é a programação do microcontrolador, mas como dito anteriormente, foram encontrados alguns problemas. No início das atividades foi observada uma modificação nas ferramentas disponibilizadas pela fabricante. Isso dificultou a busca da documentação necessária para realizar a programação. A maior parte da documentação encontrada se referia a versão anterior da biblioteca, que foi descontinuada. Além disso, não há compatibilidade entre as novas ferramentas e as antigas. Este fato resultou em uma mudança no plano de trabalho. Foi decidido tornar o microcontrolador compatível com a plataforma Arduino, amplamente difundida nos dias atuais. Essa modificação necessita o uso de um *bootloader*.

O *bootloader* é um programa utilizado para permitir a programação do PIC32 como se fosse um outro dispositivo. Inicialmente foi pesquisado por um *bootloader* feito para o PIC32MX270F256B, foi encontrado um desenvolvido para uma placa chamada chipkitPI. No primeiro momento o carregamento do *bootloader* foi feito, mas a IDE do Arduíno não conseguia detectar o microcontrolador. Pensou-se que a causa eram as configurações da IDE, o que levou a uma nova pesquisa para encontrar um arquivo que permitisse às alterações das configurações. Essa ideia foi descartada devido à complexidade e a falta de documentação para a criação desse arquivo. O próximo passo consistia em adaptar o *bootloader* para funcionar com a placa disponível. Para isso foi pesquisada a existência de uma versão compilável do *bootloader*. Mesmo com as alterações feitas, não foi possível carregar o programa no microcontrolador. O orientador então decidiu modificar a placa que estava sendo utilizada. Uma vez feitas todas as modificações foi possível a programação do PIC32.

CONSIDERAÇÕES FINAIS

O microcontrolador é a peça central do desenvolvimento do plano de trabalho, pois é responsável pelo controle de cada motor existente no robô e pela aquisição dos dados medidos através dos sensores. Cada um dos contratempos encontrados acarretaram em mudanças no plano e atrasos.

Os objetivos descritos no plano de trabalho não foram alcançados. Este problema poderia ter sido resolvido com a substituição do microcontrolador, que resultaria no não aproveitamento, em trabalhos posteriores, de tudo que seria desenvolvido. Foi muito mais proveitoso realizar as tentativas de solução, pois este conhecimento será usado em trabalhos associados ao projeto da colônia de robôs. Atualmente, o microcontrolador pode ser programado utilizando a IDE do Arduíno. Foram realizados testes com os sensores isoladamente e com o acionamento dos motores. Por fim, foram implementadas algumas rotinas de navegação em ambiente simulado, mas não foram realizados testes suficientes para validação do modelo pesquisado.

REFERÊNCIAS

- [1] S. Kajita et al., "Biped Walking Pattern Generator allowing Auxiliary ZMP Control," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 2993-2999, doi: 10.1109/IROS.2006.282233.
- [2] Qiang Huang et al., "Planning walking patterns for a biped robot," in IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 280-289, June 2001, doi: 10.1109/70.938385.
- [3] Ill-Woo Park, Jung-Yup Kim, Jungho Lee and Jun-Ho Oh, "Online free walking trajectory generation for biped humanoid robot KHR-3(HUBO)," Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., 2006, pp. 1231-1236, doi: 10.1109/ROBOT.2006.1641877.