

Movimentação de Robô Modular Apodal em Degraus

Vitor Hugo Prado Gomes

Universidade de Brasília

vitorhpgomes@gmail.com

Carla Cavalcante Koike

Departamento de Ciência da Computação

Universidade de Brasília

ckoike@unb.br

Dianne Magalhães Viana

Departamento de Engenharia Mecânica

Universidade de Brasília

diannemv@unb.br

Abstract

1. Introdução

Robôs modulares são robôs compostos pela repetição de diversas unidades autônomas iguais, e essas unidades autônomas são chamadas de módulo. A morfologia do robô modular varia de acordo com a distribuição dos módulos, que pode ser desde uma forma humanoide a uma serpente. A grande vantagem desse tipo de robô é a capacidade de alterar entre as diferentes configurações para realizar determinada tarefa da melhor maneira possível.

O estudo de robôs modulares apodais, robôs modulares que possuem formato de serpente e não dependem de pés ou rodas para se locomoverem, vem crescendo nos últimos anos devido a capacidade de se locomoverem em uma grande variedade de terrenos (Lipkin et al., 2007). Devido a natureza dos terrenos em que a configuração apodal é vantajosa, é necessário que esse tipo de robô seja capaz de desviar de obstáculos, como por exemplo atravessar um buraco no terreno ou subir um degrau.

O trabalho consiste no desenvolvimento de um algoritmo para um robô modular apodal subir um degrau. O simulador V-REP PRO EDU (E. Rohmer, 2013) foi utilizado para auxiliar no desenvolvimento e para validar o funcionamento do algoritmo.

2. Metodologia

2.1. Módulo e robô

O módulo do robô possui uma base com geometria de octaedro, e desse octaedro saem duas formas que não tem uma geometria bem definida. Dentro do octaedro ficam os componentes como arduino, bateria, placa de alimentação e

Xbee. O módulo possui dois servomotores, um em cada extremidade, e eles que são responsáveis pela movimentação do robô. O eixo de rotação desses servomotores possuem uma diferença de 90 graus, de maneira a possibilitar tanto o movimento pitch como o movimento yaw.

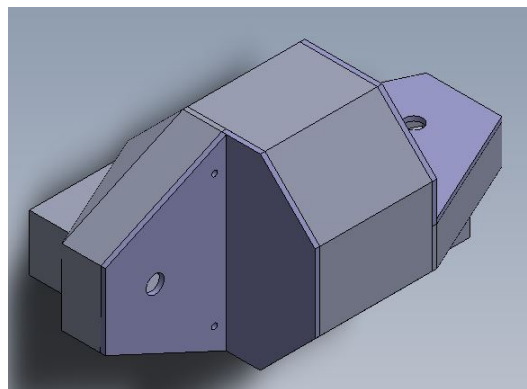


Figura 1. Vista isométrica do módulo

As Figuras 2 e 3 mostram as cotas importantes do módulo. Am é a altura do módulo que vale 80mm, Lm é a largura do módulo que vale 80mm e Cm é o comprimento do módulo que vale 155,8mm. A cota Cc , que vale 21,25mm, é importante para determinar o comprimento total de um robô com N módulos, uma vez que esse comprimento vai ser sobreposto na união dos módulos.

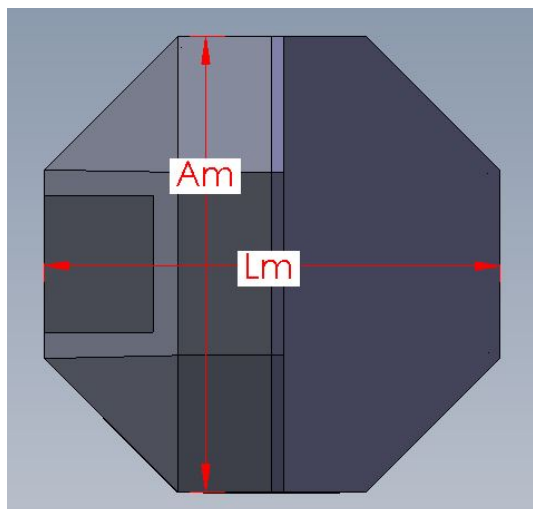


Figura 2. Largura e altura do módulo

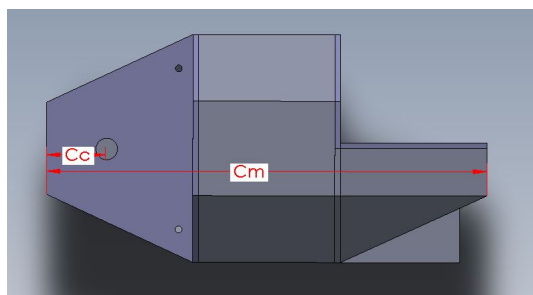


Figura 3. Comprimento do módulo e distância até o eixo do motor

Para fazer o robô, os módulos são conectados de maneira encadeada por hastes. Essas hastes ligam o servomotor responsável pelo movimento pitch de um módulo no servomotor responsável pelo movimento yaw do próximo módulo. Dessa forma, os servomotores pitch e yaw estão revezados no robô.

2.2. Preparando a simulação

O primeiro passo para fazer a simulação é replicar o módulo e o robô cobra e todas as suas características dentro do ambiente da simulação. Para isso, utilizou as ferramentas de desenho do simulador para fazer uma simplificação do módulo, de uma forma que toda a parte externa que entra em contato com o ambiente fosse replicada.

Com a carga do módulo pronto, modificou-se o centro de massa, momentos de inércia e peso de acordo com o desenho que foi feito no SolidWorks. Como apenas o servomotor é essencial para a locomoção do robô, ele foi representado como uma junta rotacional. O robô cobra que foi feito possui 10 módulos, que são conectados por hastes.

O controle das juntas do robô é feito por um script em python, que se conecta com a simulação através de um API do VREP *Remote API Functions (Python)* (n.d.).

2.3. Primeiro algoritmo

No primeiro algoritmo, o robô vai estar posicionado de maneira perpendicular ao degrau e, utilizando apenas o movimento pitch, ir colocando os módulos em cima do degrau. O movimento pitch é responsável em levantar o módulo na altura do espelho do degrau e de colocar o módulo em cima do degrau.

Uma vez que o robô de N módulos esteja posicionado, esse algoritmo pode ser representado pela máquina de estados finitos da Figura 4

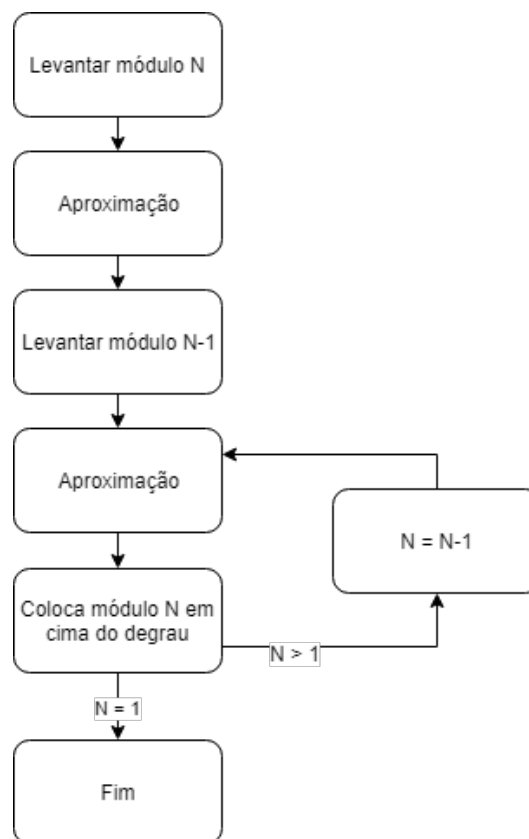


Figura 4. Máquina de estados para o algoritmo pitch-pitch

Devido ao tamanho de cada módulo e do robô, vão existir restrições nas dimensões do degrau para que o algoritmo funcione. Como o robô está posicionado de maneira perpendicular ao degrau no início do algoritmo, a largura do degrau vai estar relacionado a largura do módulo, o piso do degrau vai estar relacionado ao comprimento total do robô e o espelho do módulo vai estar relacionado a altura que consegue levantar o módulo. As Figuras 5 e 6 mostram essas restrições.

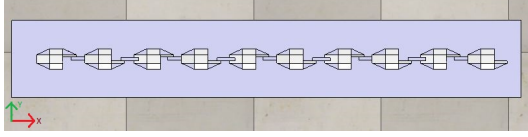


Figura 5. Tamanho mínimo para a piso (Eixo X) e para a largura (Eixo Y) do degrau no pitch-pitch

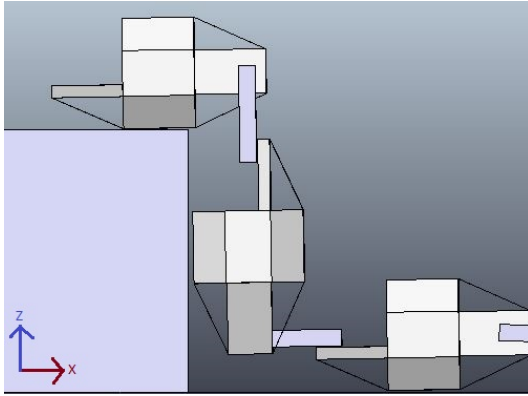


Figura 6. Tamanho máximo para o espelho (Eixo Z) do degrau no pitch-pitch

Sendo N o número de módulos do robô e Ch o comprimento da haste que ligam os módulos, podemos relacionar os tamanhos de piso, espelho e largura do degrau com os tamanhos do módulo e da haste que liga cada módulo através das equações (1), (2) e (3)

$$Espelho \leq Lm/2 + Cm - Cc \quad (1)$$

$$Piso \geq N * Cm + N * (Ch - 2 * Cc) \quad (2)$$

$$Largura \geq Lm \quad (3)$$

2.4. Segundo algoritmo

No segundo algoritmo, o robô vai estar posicionado de maneira paralela ao degrau e, utilizando uma combinação dos movimentos pitch e yaw, ir colocando os módulos em cima do degrau. O movimento pitch é utilizado para levantar os módulos na altura do espelho do degrau e o movimento yaw é responsável em colocar o módulo em cima.

Uma vez que o robô de N módulos esteja posicionado, esse algoritmo pode ser representado pela máquina de estados finitos da Figura 7

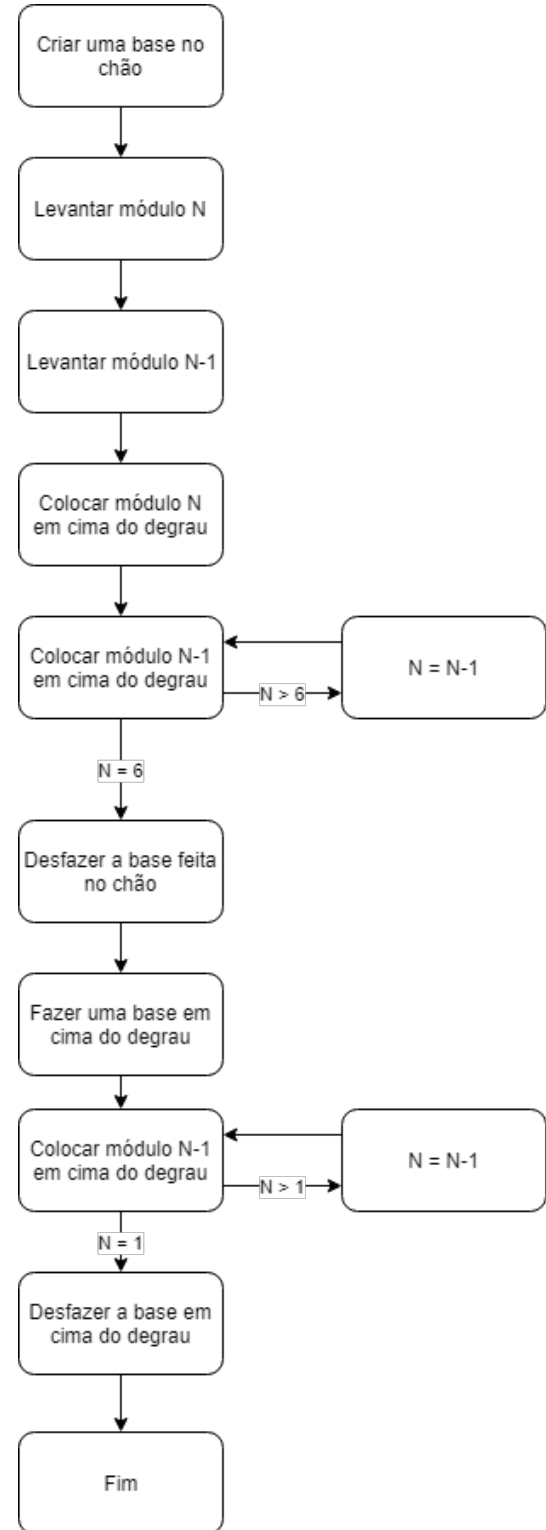


Figura 7. Máquina de estados para o algoritmo pitch-yaw

Como o robô está posicionado paralelamente ao degrau no início do algoritmo, as restrições vão ser diferentes do primeiro algoritmo. O espelho vai estar relacionado com a

altura máxima que consegue levantar o módulo, o piso vai estar relacionado com o tamanho da base que o robô vai fazer e a largura vai estar relacionada com o tamanho total do robô. As Figuras 8, 9 e 10 mostram as restrições.

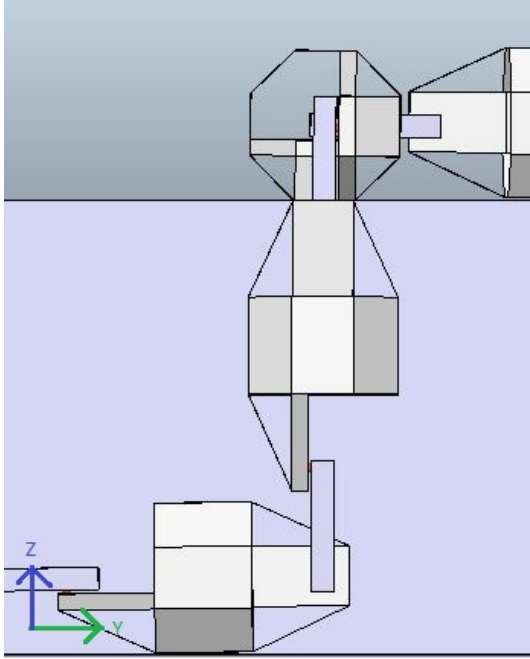


Figura 8. Tamanho máximo para o espelho (Eixo Z) do degrau no pitch-yaw

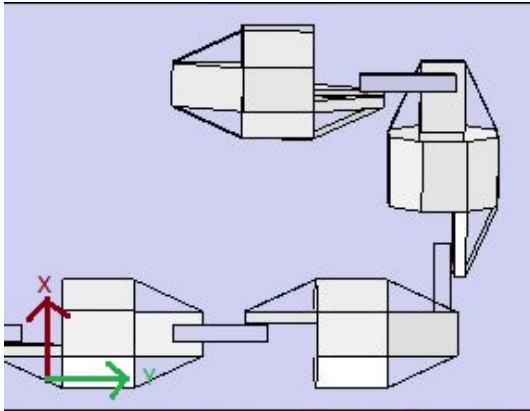


Figura 9. Tamanho mínimo para o piso (Eixo X) do degrau no pitch-yaw

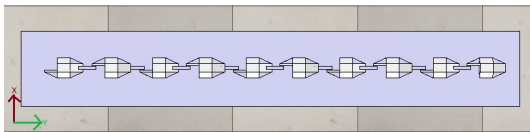


Figura 10. Tamanho mínimo para a largura (Eixo Y) do degrau no pitch-yaw

Sendo N o número de módulos do robô e Ch o comprimento da haste que ligam os módulos, pode-se relacionar os tamanhos da largura, piso e espelho do degrau com os tamanhos do módulo e da barra que ligam cada módulo através das equações (4), (5) e (6).

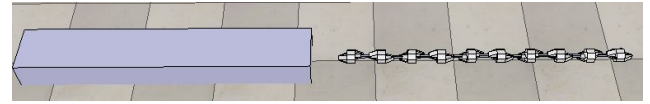
$$Espelho \leq Am/2 + Ch + Cm - Cc \quad (4)$$

$$Piso \geq Lm/2 + Ch + Cm - 2 * Cc + Lm/2 \quad (5)$$

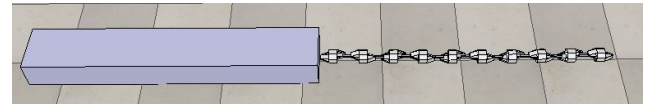
$$Largura \geq N * Cm + N * (Ch - 2 * Cc) \quad (6)$$

3. Resultados

3.1. Primeiro Algoritmo



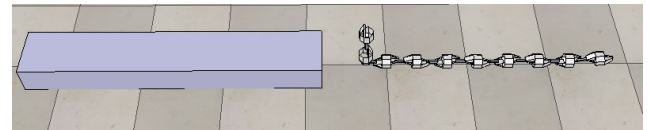
(a) Posição inicial da simulação para o primeiro algoritmo



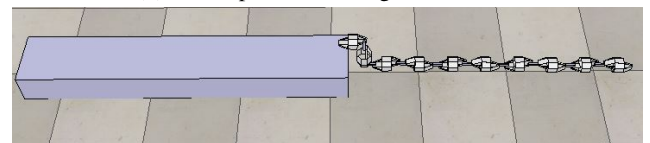
(b) Posição após a primeira aproximação



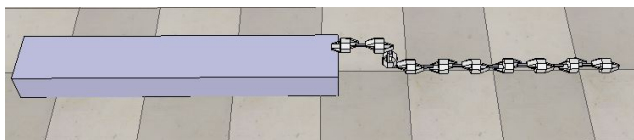
(c) Robô após levantar o primeiro módulo



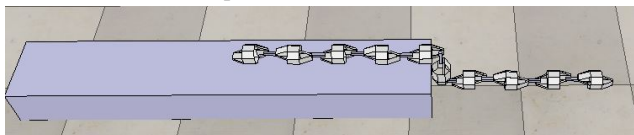
(d) Robô após levantar segundo módulo



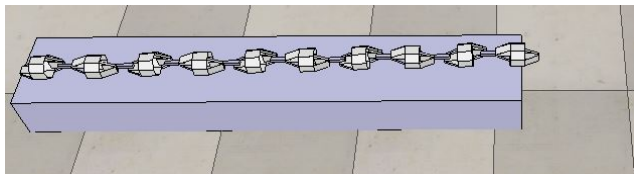
(e) Robô após colocar o primeiro módulo em cima do degrau



(f) Robô após levantar o terceiro módulo



(g) Posição do robô na metade da simulação



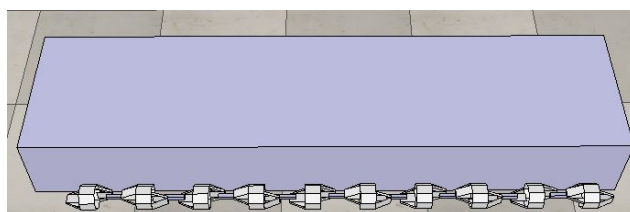
(h) Posição do robô no fim da simulação

Figura 11. Simulação do primeiro algoritmo

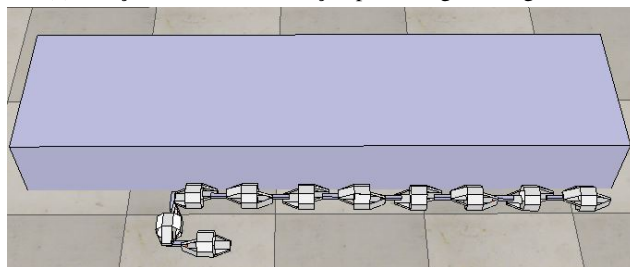
O algoritmo começa com o robô posicionado perpendicularmente ao degrau, conforme mostra a Fig. 11a. Depois, o robô faz uma aproximação ao degrau utilizando a equação 3.32 da página 86 do artigo (Gómez, 2008) até chegar o mais perto possível, conforme mostra a Fig. 11b. O robô levanta o módulo mais perto do degrau, Fig. 11c, e, por causa disso, ele fica mais longe do degrau, necessitando novamente de fazer uma aproximação. Após a aproximação, ele levanta o módulo no chão mais perto do degrau, Fig 11d, e aproxima novamente do degrau. Com os dois primeiros módulos levantados, ele coloca o primeiro módulo em cima do degrau, Fig. 11e, e logo depois levanta o próximo módulo, Fig. 11f. A partir desse ponto, o robô fica em um ciclo de aproximar do degrau e levantar o módulo do chão mais perto do degrau, até o robô inteiro estiver em cima dele. A aproximação do robô é feita utilizando os módulos em cima ou embaixo do degrau, dependendo de onde tem o maior número deles. As Figuras 11g e 11h mostram o robô na metade e no fim da simulação de subir o degrau, respectivamente.

A simulação inteira tem duração de 5 minutos e 26 segundos. A maior parte desse tempo deve-se ao movimento de aproximação que o robô faz para cada módulo que ele coloca em cima do degrau.

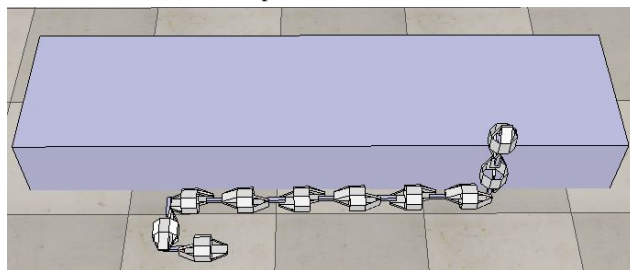
3.2. Segundo Algoritmo



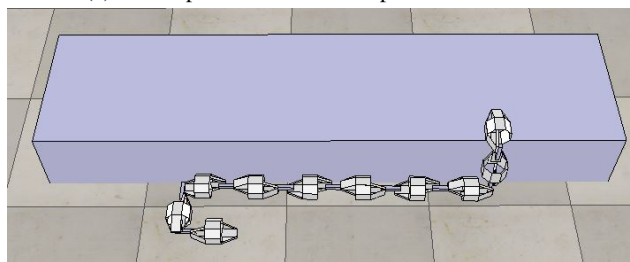
(a) Posição inicial da simulação para o segundo algoritmo



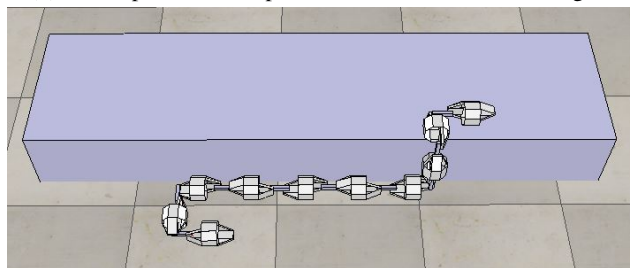
(b) Robô após fazer a base estável



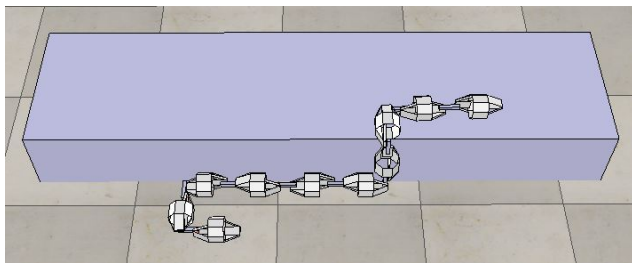
(c) Robô após levantar os dois primeiros módulos



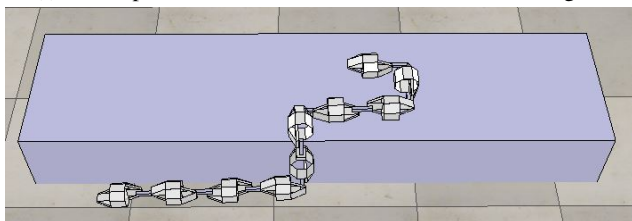
(d) Robô após colocar o primeiro módulo em cima do degrau



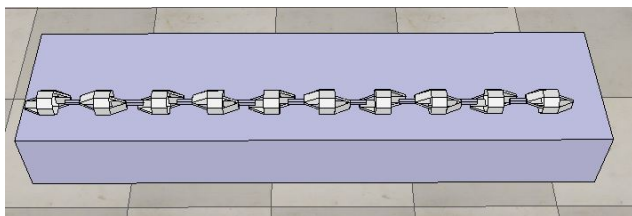
(e) Robô após colocar o segundo módulo em cima do degrau



(f) Robô após colocar o terceiro módulo em cima do degrau



(g) Robô após fazer a base em cima do degrau e desfazer a base no chão



(h) Posição final do robô para o segundo algoritmo

Figura 12. Simulação do segundo algoritmo

O algoritmo começa com o robô posicionado paralelamente ao espelho do degrau, Fig. 12a. Inicialmente é feita uma base com dois módulos para aumentar a estabilidade durante a subida do degrau, Fig. 12b. Com a base feita, os dois primeiros módulos são levantados, Fig. 12c, e o primeiro módulo é colocado em cima do degrau, Fig. 12d. Com o primeiro módulo em cima do degrau, os próximos módulos são colocados em cima um por um, Figs. 12e e 12f, até que metade do robô esteja em cima do degrau. Quando metade do robô já está na parte de cima do degrau, ele faz uma nova base com os módulos em cima do degrau e desfaz a base no chão, Fig. 12g. O mesmo movimento de colocar os módulos no degraus é repetido até o robô inteiro estar em cima dele e a base em cima do degrau é desfeita, Fig. 12h.

A simulação inteira tem duração 3 minutos e 9 segundos, 2 minutos e 17 segundos a menos que o primeiro algoritmo.

4. Conclusões

Ambos os algoritmos funcionam para degraus de escada com as dimensões de piso, espelho e largura ideais para cada um deles. O primeiro algoritmo possui restrições muito grandes para degraus de escadas de acordo com a *Norma Brasileira ABNT NBR 9050* (2015), sendo a pior de-

las o comprimento do robô estar relacionado com o tamanho do piso. Além disso, esse algoritmo utiliza apenas os servomotores que fazem o movimento pitch, subutilizando a capacidade do módulo desenvolvido, mas podendo ser utilizado por um robô mais simples.

As restrições do segundo algoritmo são mais fáceis de serem atendidas para degraus de escada, uma vez que o comprimento do robô está relacionado com a largura do degrau. Esse algoritmo utiliza todos os motores do robô para realizar o movimento, sendo necessário um robô mais complexo para sua realização. Além disso, no fim do movimento de subida de um degrau, o robô já está posicionado para subir o próximo degrau de uma escada.

Referências

- E. Rohmer, S. P. N. Singh, M. F. (2013), V-rep: a versatile and scalable robot simulation framework, in 'Proc. of The International Conference on Intelligent Robots and Systems (IROS)'.
- Gómez, J. G. (2008), 'Modular robotics and locomotion: application to limbless robots', *Pdd. Universidad Autonoma de Madrid. Madrid*.
- Lipkin, K., Brown, I., Peck, A., Choset, H., Rembisz, J., Gianfortoni, P. and Naaktgeboren, A. (2007), Differentiable and piecewise differentiable gaits for snake robots, in 'Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on', IEEE, pp. 1864–1869.
- Norma Brasileira ABNT NBR 9050* (2015), <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsPython.htm>. Acessado: 2018-08-26.
- Remote API Functions (Python)* (n.d.), <http://www.ufpb.br/cia/contents/manuais/abnt-nbr9050-edicao-2015.pdf>. Acessado: 2018-08-26.