We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Design, Simulation, and Control of a Hexapod Robot in Simscape Multibody

Claudio Urrea, Luis Valenzuela and John Kern

Additional information is available at the end of the chapter

Abstract

In this chapter, we present the design, simulation, and control of a hexapod robot using tools available in MATLAB software. In addition, we design and implement a dynamic model (using the Simscape Multibody™ toolbox) as well as a three-dimensional model of the robot, using Virtual Reality Modeling Language (VRML), that help to visualize the robot's walking sequence. This three-dimensional model is interconnected with the Simscape Multibody™ blocks using MATLAB's virtual reality blocks. Apart from this, and following specific requirements, we design and implement a Proportional–Integral–Derivative controller in order to obtain a pre-established displacement for the robot that, thanks to the developed computer simulations, proved to be satisfactory. Special emphasis is put in obtaining a modular representation of the dynamic model of the studied robot because it will permit to design more sophisticated nonlinear controllers in future works, allowing a good dynamic behavior of the robot in front of environmental perturbations, an issue that will become evident through computer simulations of its displacement.

Keywords: Mobile Hexapod Robot, Robot control, Modeling, Simscape Multibody, Virtual reality

1. Introduction

Research on multi-legged walking robots, which are created to mimic the structure of limbs and movement control in insects and arthropods, has been carried out for decades. Among many multi-legged robots, the hexapod robot is one of the most employed robots for a wide range of tasks [1, 2].

Hexapod robots have many advantages over other kinds of multi-legged walking robots: they can easily get and keep their equilibrium while moving (they are statically stable); they have the ability to adapt to irregular surfaces of different nature; they have redundancy of legs (it allow them to continue their task even if they lose a limb); they are omnidirectional and are less affected by environmental conditions than robots with wheels [3, 4].

Their advantages make them suitable for tasks requiring some degree of autonomy and high levels of reliability. Among the possible fields of application for hexapod robots, we have volcanic exploration, rescue procedures, detection of antipersonal landmines, undersea operations (marine floor), as well as sample collection, search for life, recognition missions in extraterrestrial exploration. The most of those tasks are hazardous and are usually accompanied by harsh environments, not compatible with human operation [5–7].

2. Hexapod robot

Success in designing a hexapod robot lies fundamentally in the structure of chosen legs. The main aspects of a hexapod robot's displacement are ruled by physical limitations of their legs. It is of paramount importance to choose a leg whose design provides the maximum possible range of movements and that does not pose unnecessary constraints that can affect the movement of the robot [5, 6].

2.1. Direct kinematics of a hexapod robot

In order to obtain the kinematics of the studied robot, it is necessary to use the Denavit-Hartenberg algorithm, applying it to a leg of the hexapod robot. This robot is formed by a symmetrical structure composed of six identical legs, having three degrees of freedom (DOF)
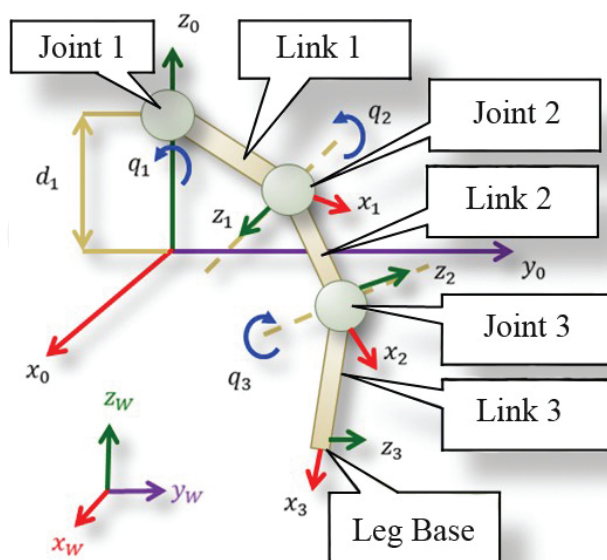

Figure 1. Model of the hexapod robot leg.

of rotational type in each leg. Thanks to its symmetry, this analysis can be done in one single leg [4, 7], as shown in **Figure 1**.

According to **Figure 1**, joint 1 is the point where the leg unites with the body of the robot (which is called "thorax"), link 1 is called "coxa," link 2 is called "femur," and link 3 is "tibia." Parameters obtained through the application of the Denavit-Hartenberg method are displayed in **Table 1**. Those parameters were obtained from the study of Olaru [7].

Link	a_i	α_i	d_i	θ_i
1	L1	90°	d1	θ1
2	L2	0°	0	θ2
3	L3	180°	0	θ3

Table 1. Denavit-Hartenberg parameters.

Those parameters originate the homogeneous transformation matrices that relate link i-1 with link i. It is possible to find the homogeneous transformation matrix for each link by means of equation (1), the which is presented in the study of Pullteap [4] and Olaru [7].

$$
{}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \cdot sen\theta_i & sen\alpha_i \cdot sen\theta_i & a_i \cdot \cos\theta_i \\ sen\theta_i & \cos\alpha_i \cdot \cos\theta_i & -sen\alpha_i \cdot \cos\theta_i & a_i \cdot sen\theta_i \\ 0 & sen\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

The matrix that relates the first link with the reference system defined in **Figure 1** is shown in equation (2); the matrix that relates the second link with the first one is shown in equation (3); and finally, the matrix that relates the second link with the base of the robot's leg is given by equation (4).

$$
{}^{0}A_1 = \begin{bmatrix} \cos\theta_1 & 0 & sen\theta_1 & L_1 \cdot \cos\theta_1 \\ sen\theta_1 & 0 & -\cos\theta_1 & L_1 \cdot sen\theta_1 \\ 0 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

$$
{}^{1}A_2 = \begin{bmatrix} \cos\theta_2 & sen\theta_2 & 0 & L_2 \cdot \cos\theta_2 \\ sen\theta_2 & -\cos\theta_2 & 0 & L_2 \cdot sen\theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}
$$

$$
{}^{2}A_{3} = \begin{bmatrix} \cos\theta_{3} & -sen\theta_{3} & 0 & L_{3}\cdot\cos\theta_{2} \\ sen\theta_{3} & \cos\theta_{3} & 0 & L_{3}\cdot sen\theta_{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{4}
$$

The total transformation is obtained by multiplying expressions (2)–(4), as pointed in equation (5), and the resulting expression represents the relationship between the system of coordinates of the robot's base with the base of the leg (6)–(9) [4, 7].

$$
{}^{0}A_{3} = {}^{0}A_{1}\cdot{}^{1}A_{2}\cdot{}^{2}A_{3}
\tag{5}
$$

$$
x = \cos\theta_{1}\cdot\left(L_{1} + L_{2}\cdot\cos\theta_{2} + L_{3}\cdot\cos\left(\theta_{2}-\theta_{3}\right)\right)
\tag{6}
$$

$$
y = sen\theta_{1}\cdot\left(L_{1} + L_{2}\cdot\cos\theta_{2} + L_{3}\cdot\cos\left(\theta_{2}-\theta_{3}\right)\right)
\tag{7}
$$

$$
z = d_{1} + L_{2}\cdot sen\theta_{2} + L_{3}\cdot sen\left(\theta_{2}-\theta_{3}\right)
\tag{8}
$$

2.2. Inverse kinematics

Inverse kinematics is the process of determination of the angles in terms of the coordinates for the leg's desired position in the Cartesian system. Unlike the problem posed by direct kinematics, the procedure for getting the equations is strongly dependent on the robot's configuration, making it a complex procedure because it is very difficult to obtain systematically those equations, even plainly impossible. Inverse kinematics, in this case, is obtained through geometrical considerations based on the leg's shape. Considering **Figure 2**, the obtained equations are shown in (8)–(11):

$$
\theta_{1} = \arctan\left(\frac{y_{1}}{x_{1}}\right)
\tag{9}
$$

$$
\theta_{2} = \arccos\left(\frac{L_{2}^{2} + x_{3}^{2} + y_{3}^{2} - L_{3}^{2}}{2\cdot L_{2}\cdot\sqrt{x_{3}^{2}+y_{3}^{2}}}\right) + \arctan\left(\frac{y_{3}}{x_{3}}\right)
\tag{10}
$$

$$
\theta_{3} = \pi - \arccos\left(\frac{L_{2}^{2} + L_{3}^{2} - \left(x_{3}^{2}+y_{3}^{2}\right)}{2\cdot L_{2}\cdot L_{3}}\right)
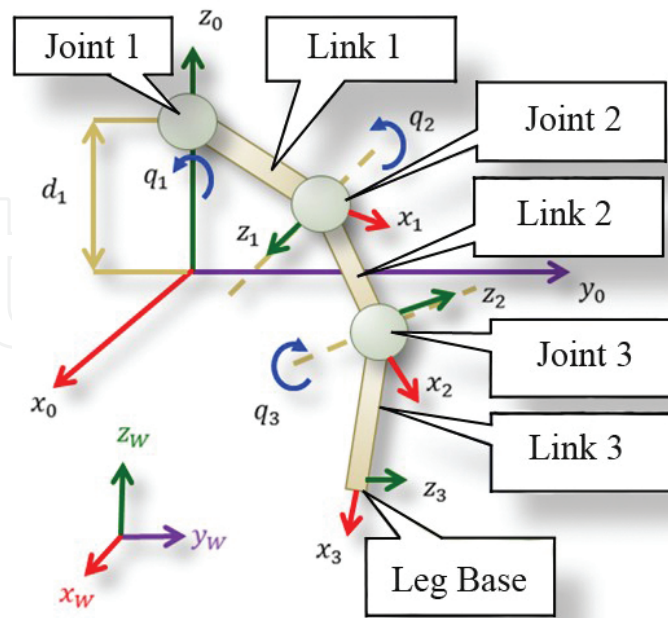\tag{11}
$$

Figure 2. Model of the leg for the hexapod robot.

3. Dynamics of the hexapod robot

The problem of obtaining a robot's dynamic model is one of the more complex aspects in the field of robotics, and it is necessary for achieving the following objectives: design and evaluation of robot's dynamic control, sizing of actuators, evaluation of the robot's mechanical structure, and motion simulation of the robot design [4].

The dynamic model of the studied hexapod robot can be obtained through the application of Lagrange-Euler or Newton-Euler algorithms. Even if the Newton-Euler method is more efficient from the point of view of computer processing, we can also employ Lagrange-Euler method, because this robot has few DOF per leg. The obtained equations are shown in (12)–(14), and they also include masses of actuators—in this case, servo motors of mass M. Besides, the analysis is simplified by assuming that the masses of the first and the last link are identical [7].

$$\tau_1 = \ddot{\theta}_1 \cdot \left(I_1' + I_1'' + M \cdot \left(l_1^2 + R_3^2 \right) + m_2 \cdot \left(r_2^2 + r_3^2 \right) \right) \tag{12}$$

$$\tau_2 = \ddot{\theta}_2 \left(I_2' + I_2'' + M l_2^2 + m_3 r_4^2 \right) - g \left(l_3 \cos(\theta_2 + \theta_3) \left(3M + m_1 + \frac{3m_2}{2} \right) + l_2 \cos\theta_2 \left(2M + m_1 + \frac{m_2}{2} \right) \right) \tag{13}$$

$$\tau_3 = \ddot{\theta}_3 \cdot \left(I_3' + I_3'' \right) - g \cdot l_3 \cdot \cos \left(\theta_2 + \theta_3 \right) \cdot \left(3M + m_1 + \frac{3m_2}{2} \right) \qquad (14)$$

where:

I_1', I_2', I_3': moment of inertia associated to servo motors 1, 2, and 3, respectively.

I_1'', I_2'', I_3'': moment of inertia associated to links 1, 2, and 3, respectively.

r_2, r_3, r_4: radius of rotation of the mass center associated to links 1, 2, and 3, respectively.

R_3: radius of the rotation circle of servo motor 3.

4. Hexapod robot model and simulations

4.1. Hexapod robot model

In order to carry out a modeling and further simulation of a hexapod robot, it is necessary to take into account the robot's physical characteristics (mass, dimensions of thorax, measures of links, and the inertia matrix). In this case, we will employ the model developed in Ref. [9], which details the robot's size, as well as the other parameters. However, some parameter modifications will be made, trying to get as close as possible to real conditions. Basically, we will employ the robot geometry shown in **Figure 3** and **4**.
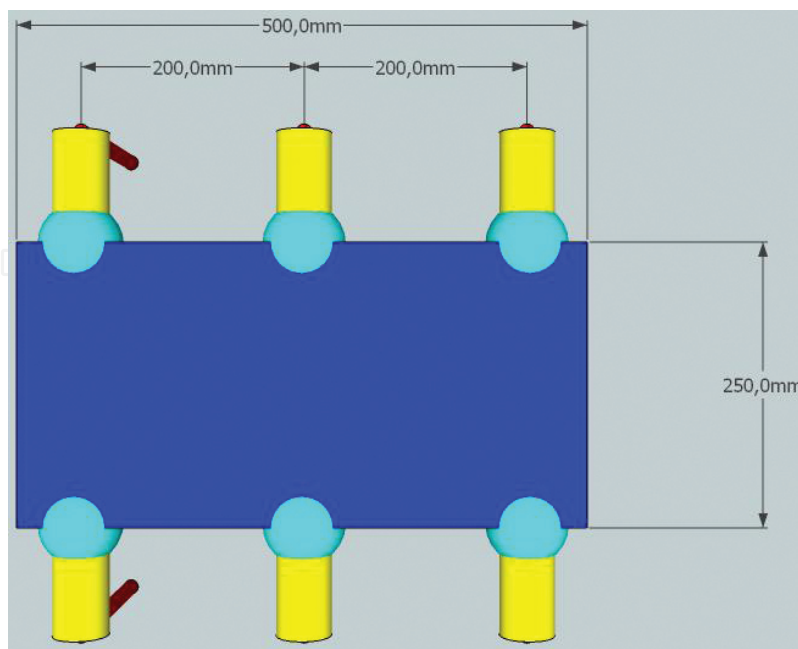

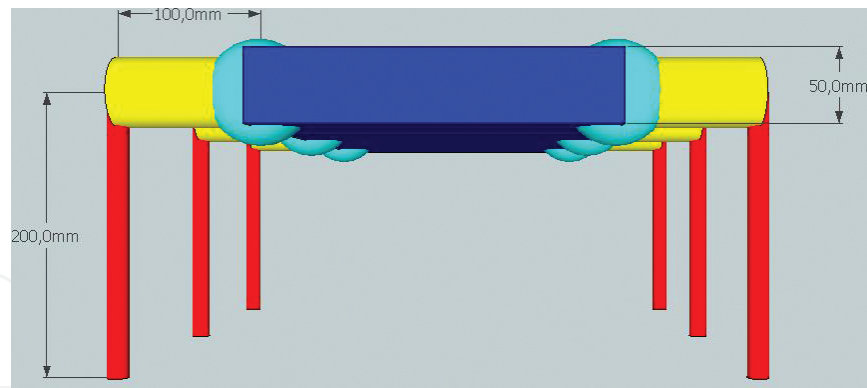**Figure 3.** Top view of the hexapod robot.

Figure 4. Front view of the hexapod robot.

To obtain the three-dimensional (3D) model of the hexapod robot, we will employ the VRML language, developing a robot model using simple geometrical figures [8]. This language allows to obtain a complex model, simply using a group of basic 3D bodies (cubes, spheres, cones, etc.). There are many alternatives to develop a 3D model using VRML language: one of them consists in programming it directly using its commands or employing a graphical editor that eases the design. In this last case, a *script* is developed containing the instructions that model the robot (**Figure 5**).
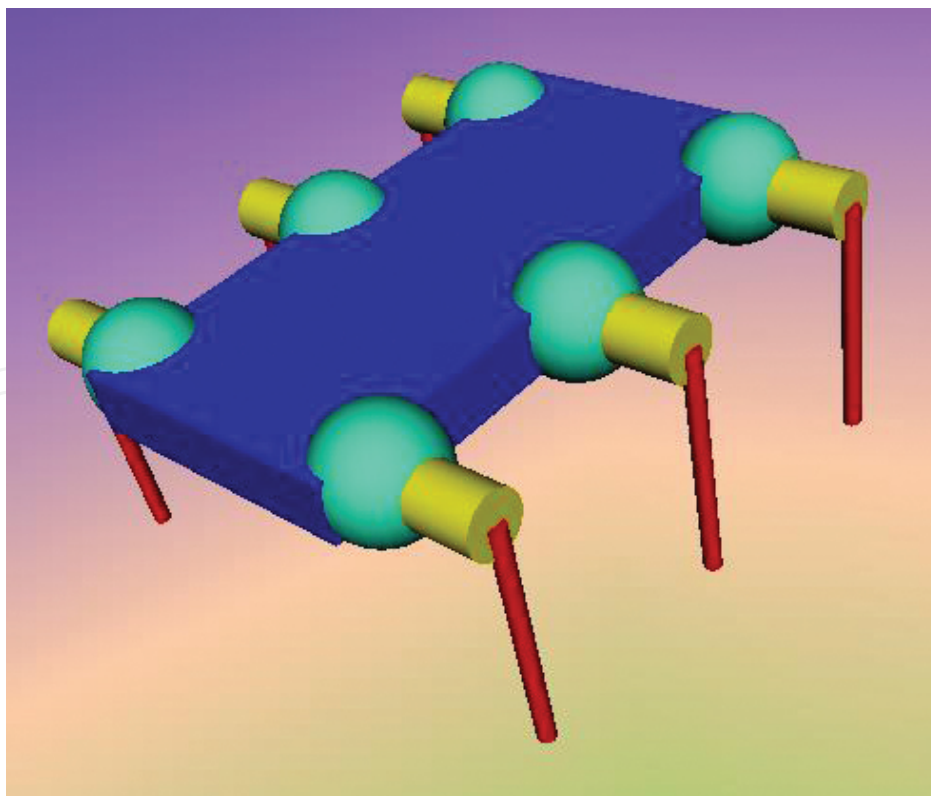

Figure 5. Perspective view of the three-dimensional model of the hexapod robot.

To obtain mass (m) of each robot's link, we take into account the geometrical considerations of the link and the kind of material that would be employed in the construction of this type of robot; for example, we consider (for simplicity's sake) that the robot legs are composed of two cylinders with different radius (R) and height (H); the thorax (the central body of the hexapod robot) has a length (l), height (h), and width (w), and it is supposed that the robot is built in transparent acrylic, whose density (p) is 1190 kg/m³. We calculate the cylinder's volume by means of equation (15), the robot's thorax volume by means of equation (16), and the mass by means of formula (17). The robot's inertia matrix is obtained using the expression (18) for the cylinder and equation (19) for the body.

$$V_C = \iiint_S dV = \pi \cdot R^2 \cdot H \tag{15}$$

$$V_P = \iiint_S dV = h \cdot w \cdot l \tag{16}$$

$$m = V \cdot \rho \tag{17}$$

$$I_C = \begin{bmatrix} \frac{1}{12} \cdot M_C \cdot \left(3R^2 + H^2\right) & 0 & 0 \\ 0 & \frac{1}{2} \cdot M_C \cdot R^2 & 0 \\ 0 & 0 & \frac{1}{12} \cdot M_C \cdot \left(3R^2 + H^2\right) \end{bmatrix} \tag{18}$$

$$I_C = \begin{bmatrix} \frac{1}{12} \cdot M_C \cdot \left(3R^2 + H^2\right) & 0 & 0 \\ 0 & \frac{1}{2} \cdot M_C \cdot R^2 & 0 \\ 0 & 0 & \frac{1}{12} \cdot M_C \cdot \left(3R^2 + H^2\right) \end{bmatrix} \tag{19}$$

It is important to calculate those parameters, because they will be employed in the simulation section. In Simscape Multibody™, this information must be entered in each corresponding block. We must notice that the calculation of masses does not includes additional loads which, in real terms, represent the robot's electronics, sensors, batteries, actuators, and cables, therefore simplifying the number of variables to be considered. The calculation of volumes and masses is shown in **Table 2**, and the corresponding inertia matrices for each link are obtained by means of expressions (20)–(22).

Parameter	Description	Value
HM	Thigh length	10 (cm)
HP	Leg length	20 (cm)
RM	Thigh radius	2.5 (cm)
RP	Leg radius	0.75 (cm)
h	Thorax height	5 (cm)
w	Thorax width	25 (cm)
l	Thorax length	50 (cm)
VT	Thorax volume	6250 (cm³)
VM	Thigh volume	196.35 (cm³)
VP	Leg volume	35.34 (cm³)
MT	Thorax mass	7437.5 (g)
MM	Thigh mass	233.66 (g)
MP	Leg mass	42.06 (g)

Table 2. Hexapod robot parameters.

$$I_{THIGH} = \begin{bmatrix} 2,312.26\,(g \cdot cm^2) & 0 & 0 \\ 0 & 730.19\,(g \cdot cm^2) & 0 \\ 0 & 0 & 2,312.6\,(g \cdot cm^2) \end{bmatrix} \tag{20}$$

$$I_{LEG} = \begin{bmatrix} 1,407.92\,(g \cdot cm^2) & 0 & 0 \\ 0 & 11.83\,(g \cdot cm^2) & 0 \\ 0 & 0 & 1,407.92\,(g \cdot cm^2) \end{bmatrix} \tag{21}$$

$$I_{THORAX} = \begin{bmatrix} 1,564,973.96\,(g \cdot cm^2) & 0 & 0 \\ 0 & 1,936,848.96\,(g \cdot cm^2) & 0 \\ 0 & 0 & 402,864.58\,(g \cdot cm^2) \end{bmatrix} \tag{22}$$

The center of mass for each link can be obtained making a simple analysis on each link. We can observe that they are symmetrical bodies, and their mass is homogeneous all over their structure, so the center of mass coincides with the geometric center of each element.

4.2. Simulations

When observing the complexity of obtained expressions, it becomes evident that the greater the number of DOF a robot has, the more difficult is to find the equations, more computer

resources are consumed, and longer time and greater effort are spent trying to obtain them. As previously mentioned, an expression for the individual dynamics of a single leg of the hexapod robot is relatively easy to obtain, nevertheless the hexapod robot has six legs, that is a total of 18 DOF, therefore making the simulation more complex. That is why we employed a MATLAB tool called Simscape Multibody™, which has the advantage to perform simulations using blocks that represent links and joints (rotational or prismatic), as if the robot was being assembled. The advantage of this is that we do not need to obtain previously the dynamic model, because those blocks simulate it by configuring parameters on each block separately. Those parameters are inertia matrix, masses, center of mass, and geometry of the robot. In order to do that, we proceed as follows: we start by implementing a leg, and next, we convert it into a block, easing handling and replication for the robot's simulation. Each leg has three associated PID control blocks, provided by MATLAB. **Figure 6** shows the blocks composing the hexapod: the blocks representing the legs, the block of virtual reality, and the block for robot trajectory.
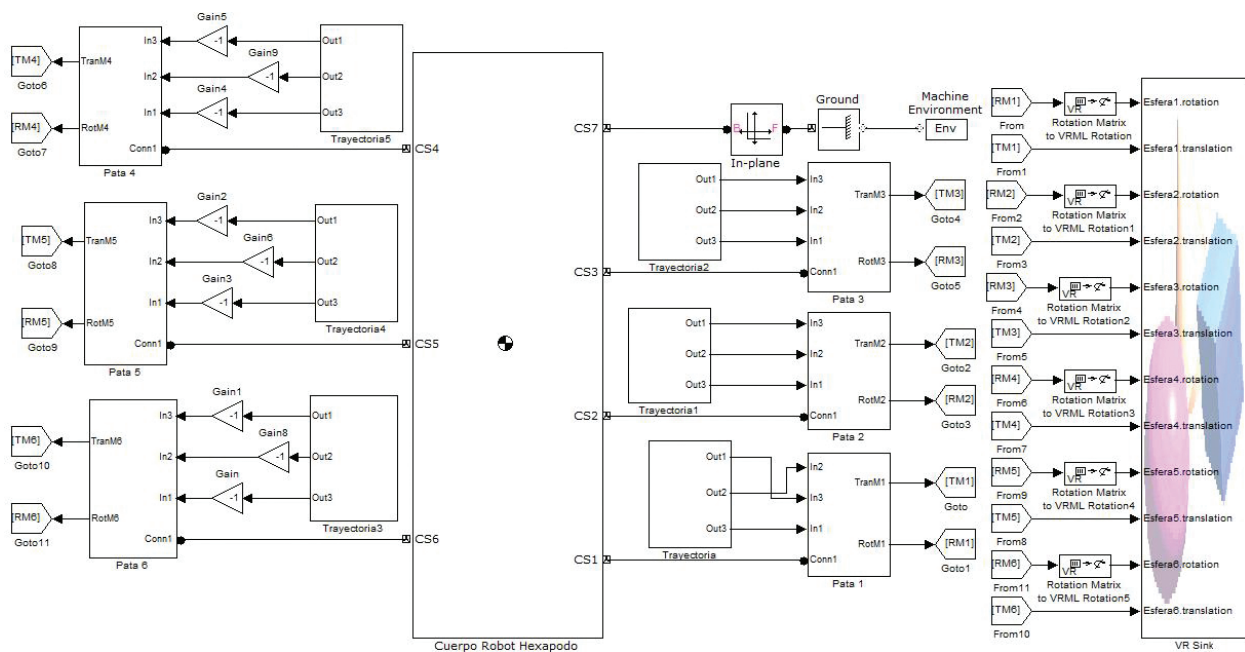

Figure 6. Model of the hexapod robot.

Despite the fact that the hexapod robot is clearly modeled through a nonlinear system, a PID control was implemented in each DOF of the robot. We started by tuning one leg (it has three PID controllers) and then replicated those parameters in the rest of the legs. This procedure was possible because in this work we did not intend to design a controller that able to perform a precise control for this robot but rather a controller that responded in a quick and acceptable manner to input references. In order to implement the controller, we used the PID block available in Simulink, which eases the process of designing and tuning the controller.

Trajectory planning is made through sinusoidal functions, according to the robot's sequence for straight-line displacement, and the work space. We developed a block containing the

positions that the hexapod robot must follow in time (see **Figure 7**). For displacements, we considered the joint that unites femur and tibia is fixed at 0° (it is not necessary that this link moves in rectilinear displacements), also considered that the joint uniting coxa and thorax will cover between −50° and 50°, and finally, that the joint uniting coxa and femur will vary from −30° to 30°. The displacement speed can be modified by manipulating the angular frequency (fixed at 1 (rad/s) in this case). The displacement sequence is detailed in **Figure 8**.
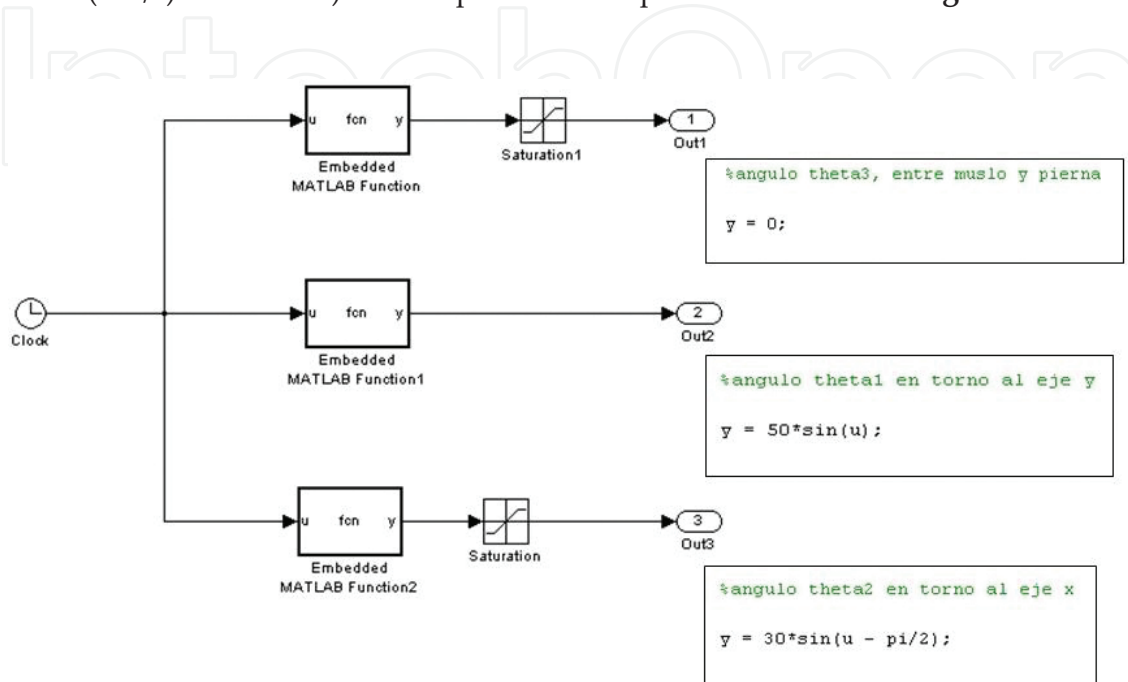

Figure 7. Trajectory planning block for each joint.

For tuning the PID controllers, we do not use a conventional method, but we do this process by varying parameters in the following way (trial and error): we vary the proportional gain without intervening the rest of the gains, and when reaching an adequate value, we continue varying the integrative gain until the stationary state error becomes zero (or near to zero), and finally we modify the derivative gain until it reaches a proper value. The proportional action delivers power enough to arrive with speed to the setpoint, the integral action eliminates the stationary state error, and the derivative action responds to the error's change speed and produces a significant correction before the magnitude of the error becomes too great. Due to the nonlinearity of the hexapod robot, the PID control is not suitable.
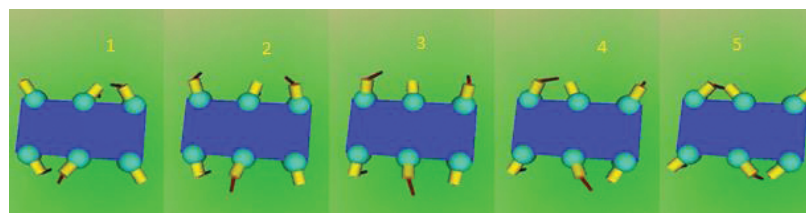

Figure 8. Simulation of the hexapod's walking sequence.

The main file that contains the folders with the programming codes of the computational simulations presented in this chapter can be downloaded directly from: http://www.mathworks.com/matlabcentral/fileexchange/56184-hexapod-robot

5. Conclusions and future development

Obtaining a dynamic model for a hexapod robot can be a laborious and complex task (especially when the robot has several DOF), which makes Simscape Multibody™ a powerful and easy-to-use tool for this purpose. We do not need to obtain an explicit dynamic model when using Simscape Multibody™ because such model is elaborated by means of blocks that represent links and joints, therefore consuming little time in the implementation of computer simulations. The development of a model in three dimensions and its further simulation help to visualize the design and possible problems that a real robot could confront (if the robot already exists or if it is going to be built).

PID controllers are not the most suitable devices to perform position control for this kind of robots; however, it is necessary to remark that the aim of our work was not to design a controller that could allow precise control, but getting the robot to respond in a quick and acceptable manner to input references. Special emphasis has been put in obtaining a modular representation of the dynamic model of the studied robot because it will permit to design more sophisticated nonlinear controllers in future works, allowing a good dynamic behavior of the robot in front of environmental perturbations, an issue that will become evident through computer simulations of its displacement.

In the near future, a robot of this kind will be designed and built, which will permit to implement algorithms for intelligent control, such as neural nets, fuzzy logic, and/or adaptive control. Additionally, not only the locomotion of the hexapod robot will be developed but also its artificial vision systems.

Acknowledgements

Author details

Claudio Urrea[*], Luis Valenzuela and John Kern

*Address all correspondence to: claudio.urrea@usach.cl

Grupo de Automática, Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile (USACH), Santiago, Chile

References

[1] Mohiuddin A. A novel navigation algorithm for hexagonal hexapod robot. American Journal of Engineering and Applied Sciences. 2010;3(2):320–327.

[2] Bo J. Design and Configuration of a Hexapod Walking Robot. In: Third International Conference on Measuring Technology and Mechatronics Automation, IEEE Computer Society; 2011. pp. 863–866.

[3] Duan X. Tripod Gaits Planning and Kinematics Analysis of a Hexapod Robot. In: IEEE International Conference on Control and Automation; Christchurch, New Zealand. 2009. p. 1850–1855.

[4] Pullteap S. Development of a hexapod robot controlling by fuzzy logic controller. International Journal of Artificial, Intelligence, and Mechatronics. 2013;1(6):141–146.

[5] Nonami K., Kumar R., Irawan A., and Razali M. Hydraulically Actuated Hexapod Robots. Kelly Bennett, In Design, Implementation and Control. Japan: Springer; 2014.

[6] Fedák V., Ďurovský F., and Üveges R. Analysis of Robotic System Motion in SimMechanics and MATLAB GUI Environment. In: MATLAB Applications for the Practical Engineer, Chapter 20; New York, USA., Kelly Bennett InTech; 2014. [Internet]. Available from: http://www.intechopen.com/books/matlab-applications-for-the-practical-engineer/analysis-of-robotic-system-motion-in-simmechanics-and-matlab-gui-environment [Accessed: March 14, 2016]

[7] Olaru M. Hexapod robot. Mathematical support for modeling and control. In: System Theory, Control, and Computing (ICSTCC), 2011 15th International Conference on, 14–16 October, Sinaia. IEEE; 2011. p. 1–6.

[8] Barai R. K., Saha P., and Mandal A. SMART-HexBot: a Simulation, Modeling, Analysis and Research Tool for Hexapod Robot in Virtual Reality and Simulin. In: AIR '13 Proceedings of Conference on Advances in Robotics, July 4th–6th, Pune, India. New York, NY, USA: ACM; 2013.

[9] MathWorks. MathWorks - MATLAB and Simulink for Technical Computing [Internet]. Available from: http://www.mathworks.com/?requestedDomain=www.mathworks.com [Accessed: January 12, 2016].