# Locomotion planning using OBs for a Robot Nao

MONTECILLO-PUENTE, Francisco Javier*†, SAMANO-ABONCE, Obed Noé and LÓPEZ-QUIJAS, Daniel.

*Instituto Tecnológico Superior de Salvatierra. CP. 38933*

Received July 28, 2016; Accepted October 27, 2016

---

## Abstract

Robotics in industry is a reality since 1990's. There are different kinds of robots working in manufacturing cells, industrial processes, medicine, servicing and even entertainment. However, robotics in daily live is at an early stage, this kind of robotics applications is called servicing robotics. There are still several challenges in servicing robotics to solve such as human-robot interaction, real-time 3D environment modelling, and autonomous locomotion planning, among others. In this work a methodology for locomotion planning for a humanoid robot Nao is presented. The main contribution of this work is that our planner takes into account three kinds of locomotion: frontal, lateral and four-contact points based. These are directly considered as part of the locomotion plan. In order to verify the proposed method three scenarios are tested on simulation. Finally, a discussion on the results is also presented.

**Locomotion, humanoids, motion planning**

---

**Citation:** MONTECILLO-PUENTE, Francisco Javier, SAMANO-ABONCE, Obed Noé and LÓPEZ-QUIJAS, Daniel. Locomotion planning using OBs for a Robot Nao. ECORFAN Journal-Democratic Republic of Congo 2016, 2-3: 30-38

---

* Correspondence to Author (email: famontecillo@itess.edu.mx)

† Researcher contributing first author.

## Introduction

Robotics in the industry has had significant success in recent years, its applications encompassing the automotive industry, auto parts, electronics assembly, pharmaceutical industry, aeronautics and aerospace as well as entertainment. In a report from the International Federation of Robotics IFR, World Robotics Report 2016, it is mentioned that 36,000 new robots were installed in the USA, Canada and Mexico; It also indicates that from 2010 to 2015 the main robot developers from the USA, Europe and Asia installed 80 000 robots.

However, one area of opportunity is service robotics, which refers to the use of robots in everyday human environments. Some efforts by the industry have been made to bring robotics to the home such as the robot robots iRobot, company founded by Rodney A. Brooks. Also humanoid type robot robots have come to market as the Softbank Robotics Pepper robot. Service robotics is one of the areas that are still under development and requires extensive study to be successful as industrial robotics. Within the challenges of service robotics are the human-machine interaction that resembles how humans interact: voice, body and facial gestures; Also the locomotion, planning of movements and the avoidance of obstacles, among others. The Robot Nao of Softbank Robotics is one of the most successful humanoid type robots, however development of new applications and usage is difficult due to its fragility in locomotion, (Who is not?). Therefore, this aspect should be further explored. In this paper, we present a locomotion planner for Nao that includes and exploits its locomotion capacities. Locomotion planners have been developed for different robotic platforms such as ASIMO, HRP2, HUBO, among others (Kajita et al., 2003), (Masato & Kenichi, 2007), (Ill-Woo et al., 2006).

Also, theoretical planners have been developed that in a real situation are difficult to execute successfully (Hauser et al., 2008). The robot has no integrated locomotion system that can be moved forward, backward, lateral, front and back following curved trajectories; In this work we propose a locomotion planning system that includes a new form of locomotion on four legs. In this locomotion planner for Nao three categories of displacement are categorized that are frontal, lateral and on four legs; the locomotions are prioritized according to their speed of movement and the transition points between the different modalities are determined.

With the aforementioned characteristics presents a new planner for NAO in simulation and for several different scenarios. With this new locomotion planner NAO's navigation capabilities are expanded in environments where it uses the four-legged locomotion mode, as illustrated in Figure 1.

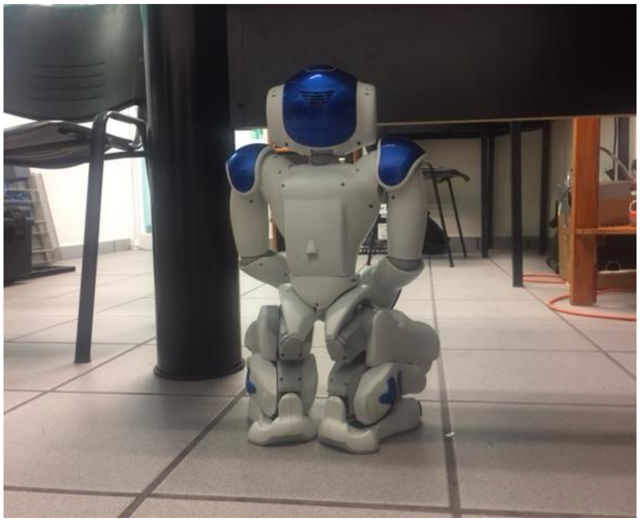


**Figure 1** This image shows a case where the NAO robot needs to move to the other side of the desktop. However it is obstructed by a chair, so it is not possible to move, the only possibility to move is four legs under the desk or chair.

The content of the article is presented as follows, the next section 2 presents a brief review of what is the movement planning, then describes the robot Nao and its modalities of locomotion in section 3. Then, in the Section 4 presents the collision detector used for the planner, later in section 5 the scheduler for the robot NAO is presented. The test scenarios and the results obtained are presented in section 6, then section 7 presents the conclusions and finally the references.

## Movement Planning

The problem of movement planning consists of a set of valid states and a set of actions, determining an obstacle-free path that allows a robot to move from an initial state to an end state (LaValle, 2006) (LaValle Et al., 2001). To pose a problem of movement planning and solve it requires the following parts:

a) Modeling of the robot geometry consisting of the computational representation of the real robot with geometric primitives such as triangles, spheres, boxes, among others.

b) Modeling the movements of the robot or object (hereinafter referred to simply as a robot), which consists in defining the independent degrees of freedom of movement that will be used to change the state of the robot.

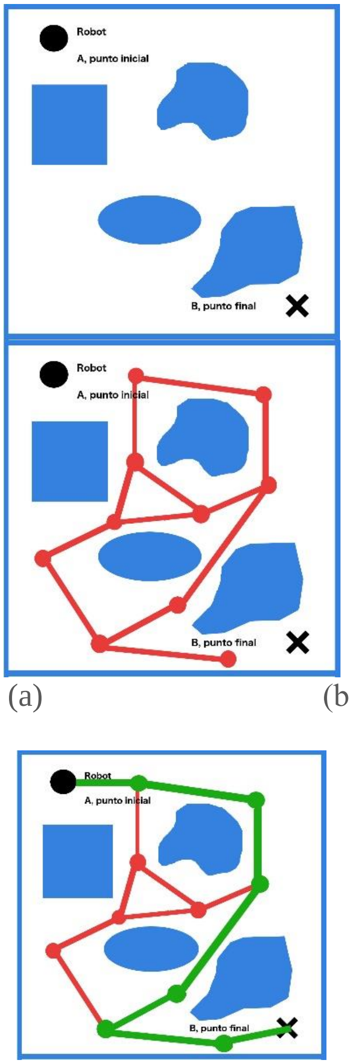
c) Modeling of the scenario and obstacles where the robot navigates, as a) refers to the computational representation of the scenario and objects present in it with which the robot interacts.

d) Collision detector, the collision detector is one of the most important parts of the planner is to determine if the representation of the robot in the current state is in collision with some element of the scenario.

e) Design of the strategy to create a navigation map generally based on a sampling scheme and graphs.

f) Strategy to determine if there is a solution for the robot to find a path from an initial state to the final state or, if applicable, a termination criterion.

In Figure 2, an example is shown to determine a route to go from point A to point B. The problem is to find an obstacle-free route with movements that the robot can perform.



(a) (b)

(c)

**Figure 2** Example of motion planning a) the robot is presented as a point, the obstacles in blue, the initial position A, the final position B; B) it is presented the sampling of the states where there is no collision and the graph that is formed, c) to determine the route between the points A and B is solved connecting A and B with the graph.

## Nao and his model of locomotion

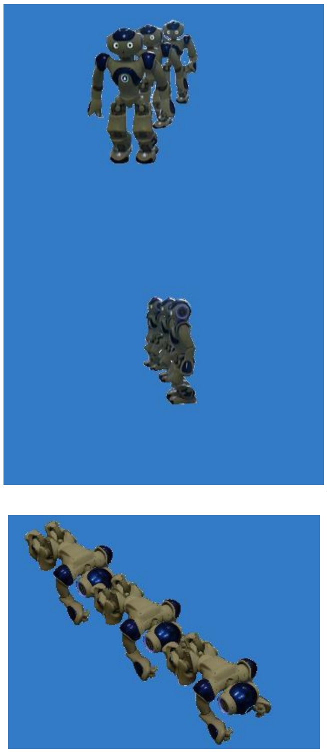
The robot Nao is a humanoid robot of 57.4 cm, 5.4 kg, with 25 degrees of freedom (dof) distributed as shown in Table 1.

| Body part | **Degree of freedom or Board** |
|---|---|
| Head 2 dof | HeadYaw, HeadPitch |
| Left Arm 6 dof | LShoulderPitch, LShoulderRoll, LElbowYaw, LElbowRoll, LWristYaw, LHand |
| Right arm 6 dof | RShoulderPitch, RShoulderRoll, RElbowYaw, RElbowRoll, RWristYaw, RHand |
| Pelvis 1 dof | LHipYawPitch, RHipYawPitch<br>* se controlan con un motor |
| Left leg 5 dof | LHipRoll, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll |
| Right leg 5 dof | LHipRoll, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll |

**Table 1** In this table the degrees of freedom of the robot Nao are presented in the different parts of the body or anthropomorphic structure.

The robot does not have a locomotion generator that is based on the position and orientation of the feet for the steps to be performed, then the generator calculates the trajectory of the joints (dof) keeping the robot balanced. If the robot is indicated the route that must follow the generator performs it, however this route must be previously planned so that the robot Nao does not move from an initial point to an end point.

Our proposal is to create a planner that includes Nao's locomotion generator and a four-legged modality. In particular, we consider the locomotion generation of the frontal and lateral robot together with the four-legged locomotion, see Figure 3. The four-legged locomotion integrated to the planner is the main contribution of this work. The planner considers that objects are not in motion, which in real scenarios this usually does not occur. However, this planner is the first one that arises in the Institute to begin to develop algorithms and methods for the future.



a) b)

c)

**Figure 3** In this figure the three modalities of locomotion of the robot are shown: a) frontal, b) lateral and c) four legs.

A geometric tool to represent physically objects are Oriented Bounding Box (OBB), in our case the OBB were used to represent the robot. Therefore, OBBs have three degrees of freedom two for position and one for orientation, see Figure 4.

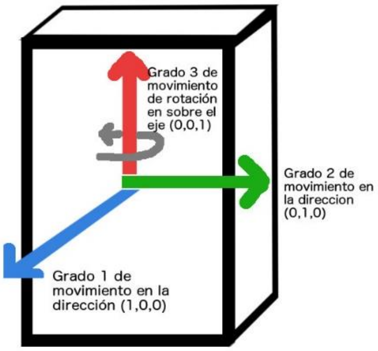


**Figure 4** This figure shows the three degrees of freedom of the OBB used to model the robot Nao, in this OBB the robot can be wrapped and perform locomotion.

The OBB were used to represent geometrically the three modalities of locomotion. The OBB shown in Figure 5 are the minimum boxes where the Nao body can be wrapped in each of the movement modalities, according to the dimensions of Table 2. Therefore, the robot geometrically is modeled by three OBB: 1) OBBf representing frontal displacement, 2) OBBl lateral displacement and 3) OBBcp four-legged displacement.

| OBB | **Dimensions** |
|---|---|
| Frontal | Height:55cm<br>Width:39cm<br>Depth:26cm |
| Side | Height:55cm<br>Width:26cm<br>Depth: 39cm |
| On four legs | Height:24cm<br>Width:39cm<br>Depth: 42cm |

**Table 2** This table shows the degrees of freedom of the robot Nao in the different parts of the body or anthropomorphic structure.

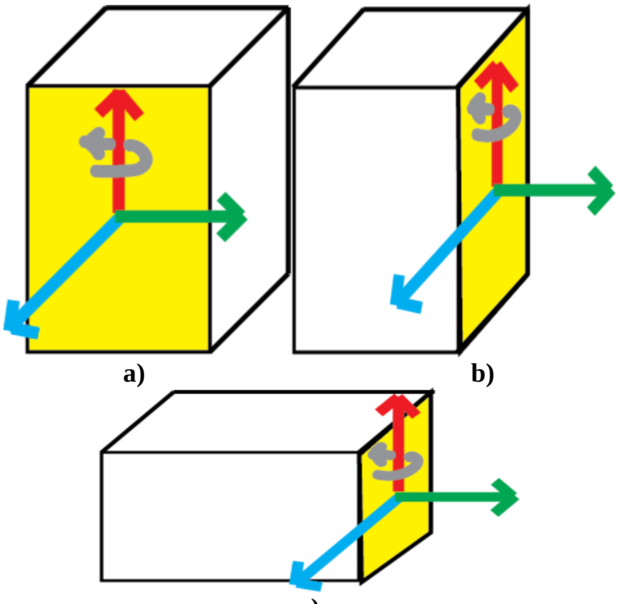


**Figure 5** OBB that envelop the robot in each of the modalities of locomotion. In the illuminated part in yellow the direction of movement is shown, in addition the three degrees of freedom of each OBB are shown: a) frontal, b) lateral, c) on four legs.

## OOBB Geometry Collision Detector

The collision detector is one of the most important components for a trajectory planner. In general, there are different methods to detect collision the most accurate, but with a lot of computational load, are based on verifying that all primitives between two geometries do not intersect. Others try to simplify the robot to geometric entities that involve large regions of the robot, using Axis-aligned Bounding Boxes (AABB), Oriented Bounding Box (OBB), spheres, cylinders or Combinations of these, (Stase et al., 2008), (Kanehiro et al., 2008), (Stolpner et al., 2012). The geometry of the robot is not shown in Table 3 which contains about 110 000 geometric primitives, in Figure 5 the robot is shown in the form of primitives. If geometries are used close to the real objects for the detection of collisions it is translated in computational time which for our case is not very convenient, since the presented planner is proposed to be used in line in interior environments.

| Geometric primitives | **Cantidad** |
|---|---:|
| points | 52 158 |
| Triangles | 74 893 |
| Squares | 40 390 |

**Table 3** Number of points and geometric primitives that make up the robot Nao.

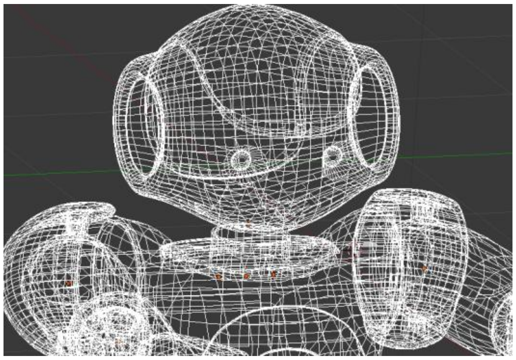


**Figure 6** Exact geometric model of a section of the robot Nao showing the triangular and square primitives. When exact geometries are used to determine collision, each primitive of an object is compared to those that make up the other, this is computationally expensive.

To reduce the computational load we model the environment and robots as OBB which is a good approximation for us, that we look for the robot to have a relatively large tolerance. For collision detection between two OBBs, a method called a separator axis is used, which consists of projecting the vector between the centers T and the radii of the two objects rA and rB on an axis, see Figure 7. The OBB Are separated if Equation 1 is satisfied.

$$|T \cdot L| > r_A + r_B \tag{1}$$

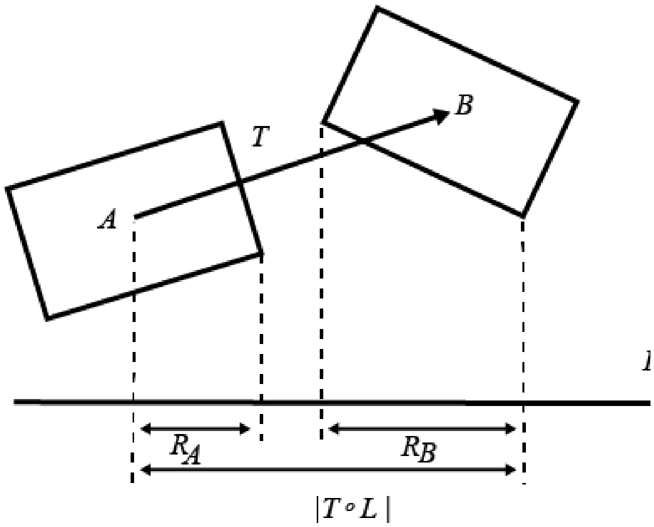


**Figure 7** This figure shows the parameters involved in to test if a face of an OBB are separated.

For the detection of collision between two OBBs it is required to calculate 15 axes separators only, (Ericson, 2004), so compared to the approximately 110 000 primitives of the geometric model is a large reduction.

## Locomotion planner for robot Nao

This section outlines the locomotion planner that includes the three modalities: front, side and four legs. Assuming that the scenario and the robot is modeled by OBBs, the steps for the planner are as follows:

a) Use a random posture generation method that includes position and orientation (x, y, θ□). In particular, we use a random function with a constant probability function.

b) Create a set of vertices V = {v1, v2, θ...}, where each vertex represents a pose and locomotion-free locomotion modalities, (xi, yi, □θi, F, L, CP) that do not Is in collision where F, L and CP correspond to the frontal, lateral and four legs that are satisfied. They take binary values F, L and CP.

c) With the set of vertices, the ones that are close to each other are searched to form edges, thus creating a graph G = {V, E}, where V is the set of vertices of b) and E is the set of edges. To create edges a method was used that finds the nearest neighbors. Then, to verify that the edge is valid this is followed by checking that it is not in collision with the objects of the stage, for this a scrolling step is used.

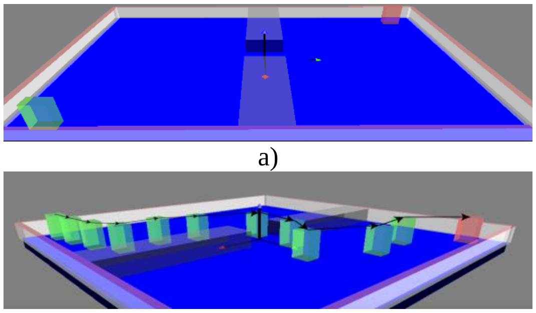
d) Once the graph created, given a slogan, that is, an initial posture and a final posture, it is necessary to look for a route that connects the two postures. For this, first the initial and final postures are connected to their closest vertex and then the graph is explored to find the route between the two vertices using the classic algorithms of graphs. Here, one of the modifications is prioritized to choose the vertex that has the modality of frontal locomotion, then Lateral and finally to four legs.

e) It verifies the search result that can be successful or fails. In the case of being successful a smoothing in the path is made so that it seems more realistic, in case of failure one can proceed to create more vertices and edges and repeat d) and e) or simply report a failure in determining a route.

## Test scenarios and discussion of results

This section presents the test scenarios and the results obtained for three scenarios that illustrate the cases for which the planner is designed. The implementation was done in C ++, using OpenGL 2.0 and Boost Graph Library on a MacBook Pro computer with 2.4 GHz Core i5 processor without optimizing code for parallel processing.

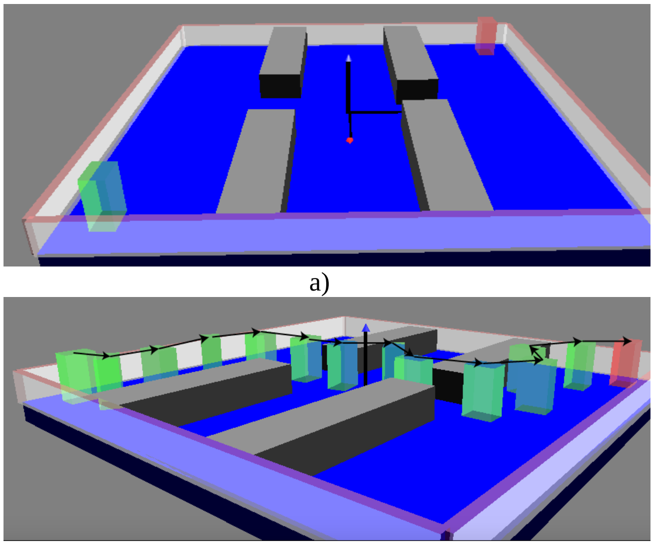
In the first scenario, the frontal mode is used only to solve the problem, Figure 8 shows the scenario and the solution applying the scheduler. For this solution only 100 vertices were created, to achieve these vertices those that were close to each other were eliminated. Then, the graph was limited to creating 300 edges, ensuring that the two parts of the scenario are connected. This was done to not have a very large graph and that the search times were fast, in general it was determined that it lasts approximately 0.6 ms in the search.


a)

b)

**Figure 8** This figure shows the scenario where the setpoint of moving from the green start point to the red destination point is displayed, an obstacle is presented in the center of the stage. In b) the solution of our method is shown, only some parts of the route are shown.

In scenario two there are two broad and one narrower passages, see Figure 9. To solve this problem, it is required that in the second passage the mode of lateral locomotion is used, although using the frontal locomotion can cross the second passage Very tightly, this is not used since in many of the postures generated are in collision but are not in lateral locomotion. The solution of this problem we use 200 vertices and 400 edges.


a)

b)

**Figure 9** This figure shows a) the scenario that consists of two passages and shows the setpoint that is the position of the OBB in red. In b) the solution of our method is shown, only some parts of the route are shown.

Finally, in the third scenario two obstacles are presented, one with frontal locomotion and the other that to cross it, it is necessary to use the modality on all fours since it is impossible to use the other modalities to pass through the second passage, see Figure 10.

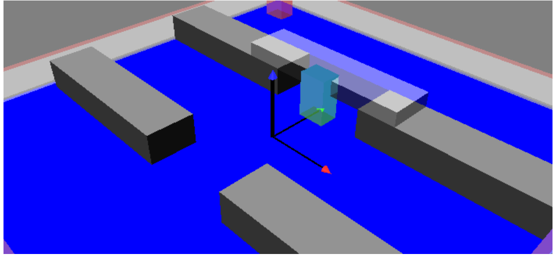


**Figure 10** OBB of the robot will always be in collision using lateral and frontal modalities in the second passage.

The solution found by our method for the third scenario is presented in Figure 11, for this solution we also use 200 vertices and 400 edges.

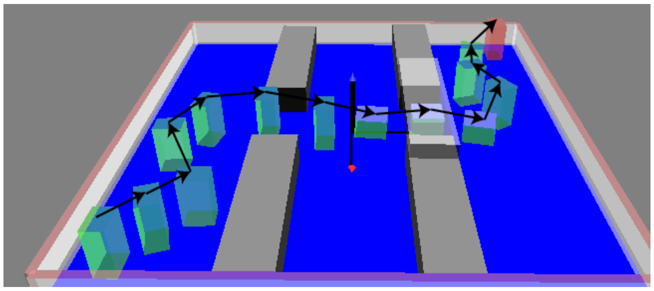


**Figure 11** Part of the route for stage 3 is shown in this figure, it can be seen that to pass through the second passage it was necessary to use four-legged locomotion.

In general, the most difficult part to solve locomotion planning instructions is the construction of graph. With the paths found by the algorithm these still have to be processed so that the robot can perform them, since the temporal variable must be added. In addition, in the execution the robot has an important sliding and sliding in the feet, here important work for the execution of trajectories with more exactitude is required.

For static real scenarios it is only necessary to have a good representation of the obstacles using OBB at different scales and apply the scheduler presented here. With some modifications this planner can also be used to plan movement on manipulator robots. In the case of scenarios with moving objects, other types of planners are necessary, but this work is a starting point to create this kind of planners in the future.

Another very important problem is when you do not know the geometry of the scenario so the robot must create it and at the same time planning. This problem in mobile robotics is known as SLAM, however for humanoid robots it is a more complicated problem because the geometry must have a very high accuracy.

## Conclusions

The planner shown in this paper presents in simulation three cases that illustrate the application about situations that may arise in reality. In particular, the scenario where the standing robot cannot cross the passage the planner determines a route where he uses locomotion on all fours.

This scheduler can be used in spaces where scenarios are known, modeled with OBBs and there are no mobile obstacles; which in reality is not very common reason why to develop planners that include obstacles in movement is an area of opportunity.

## References

World Robotics Report 2016. (2016). Obtenido de http://www.ifr.org/: http://www.ifr.org/news/ifr-press-release/world-robotics-report-2016-832/

Kajita, S., Kanehiro, F., Kaneko K., Fujiwara, K., Harada, K., Yokoi, K. & Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. IEEE International Conference on Robotics and Automation, 2003. Proceedings ICRA'03.

Masato, H. & Kenichi, O. (2007). Honda humanoid robots development. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Volumen (365), issue 1850,11-19.

Ill-Woo, P., Jung-Yup, K., Jungho, L. & Jun-Ho O. (2006). Online Free Walking Trajectory Generation for Biped Humanoid Robot KHR-3 (HUBO). Proceedings IEEE International Conference on Robotics & Automation, May, 2006,1231 - 1236.

Hauser, K., Bretl, T., Latombe, J.C., Harada, K. & Wilcox, B. (2008). Motion planning for legged robots on varied terrain. International Journal of Robotics Research.

LaValle, S. M. (2006). Planning Algorithms. Cambridge University Press.

LaValle, Steven M., Kuffner Jr. & James J. (2001). Randomized Kinodynamic Planning. The International Journal of Robotics Research.

Ericson, C. (2004). Real-Time Collision Detection. CRC Press, Inc. Boca Raton, FL, USA.

Stasse, O., Escande, A., Mansard, N., Miossec, S., Evrard, P. & Kheddar, A. (2008). Real-Time (Self)-Collision Avoidance Task on a HRP-2 Humanoid Robot. Proceedings of IEEE International Conference on Robotics and Automation · May 2008.

Kanehiro, F., Lamiraux, F., Kanoun, O., Yoshida, E. & Laumond J.-P. (2008). A Local Collision Avoidance Method for Non-strictly Convex Polyhedra. Proceedings of Robotics: Science and Systems IV.

Stolpner, S., Kry, P. & Siddiqi, K. (2012). Medial Spheres for Shape Approximation. IEEE Transations on Pattern Analysis and Machine Intelligence.