NAO ROBOT IN VIRTUAL ENVIRONMENT APPLIED ON VIPRO PLATFORM

Octavian MELINTE¹, Luige VLADAREANU¹, Radu Adrian MUNTEANU², Hongbo WANG³, Florentin SMARANDACHE⁴, Mumtaz ALI⁵

1 Institute of Solid Mechanics, Romanian Academy, Bucharest
2 Technical University of Cluj-Napoca, 15 C-tin Daicoviciu, Cluj Napoca, ROMANIA
3 Parallel Robot and Mechatronic Sys. Lab. Yanshan University Qinhuangdao, China
4 University of New Mexico, 705 Gurley Ave., Gallup, New Mexico 87301, USA, E-mail: fsmarandache@gmail.com
5 Department of Mathematics, Quaid-i-Azam University, Islamabad, 44000, Pakistan, E-mail: mumtazali7288@gmail.com

Abstract. This paper investigates the behavior of the Nao humanoid robot moving and avoiding obstacles in virtual environment and the integration in the Versatile Intelligent Portable Robot Platform (VIPRO). In order to achieve better results the robot walk was developed using a preview controller and the object that Nao will find in his path is pre-processed and stored in robot database. The stability during the walk was achieved by using the ZMP method which allows the dynamical control of the mechanism.

Keywords: robot, motion control, obstacle avoidance, VIPRO platform

1. INTRODUCTION

The research on Nao robot motion control and development of decision strategies is part of a larger project which will implement results obtained across different platforms in the VIPRO robotic platform. This platform will provide testing, simulation and implementation of programs developed by users in real robotic systems. The VIPRO intelligent control interfaces adapted to the surrounding environment were developed as alternative to classic AI, where the intelligent behavior was built in a top-down manner. Using this concept, the robot used the motion networks and real time control that simulates the artificial intelligence, adding the control complex laws in order to improve the stability of the autonomous robots [1].

For this purpose a strategy was used for dynamic stabilization and balance of the walking robot by applying the method and device for real time robot control in virtual projection [2][3] known as the Vladareanu-Munteanu method. This was integrated in the development of robot control. The Choregraph software development kit and Webots environment were selected based on analyzing the available solutions for to modeling and respectively simulation of the 3D objects, robots and the environments in which they are moving.

2. CONTROLLING NAO ROBOT

NAO is a humanoid robot developed by Aldebaran, it has a medium size of 57.3, weighs 4.3 kg and is used in research and education, in particular. Because programming can be written in several programming languages and programs can be compiled both local (on the robot) and remote, NAO can be used in various applications [4].

As shown in Figure 3 NAO robot has 25 degrees of freedom, their positioning on the robot is as follows:
- 2 degrees of freedom for head motion;
- 5 degrees of freedom on each hand,
- a degree of freedom for the pelvis;
- 5 degrees of freedom on each leg,
- an additional degree of freedom on each palm, for grasp.

Kinematics of the NAO hand consists of two degrees of freedom on the shoulder (pitch and roll), two degrees in the elbow (roll and yaw), one in the wrist (yaw) and one additional for gripping objects with its fingers.

NAO is equipped with actuators that meet the requirements for setting high performances: they are compact, lightweight, reverse controlled, efficient, accurate and reliable. Also, among the requirements for the actuators are the power over weight ratio, high torque for short amount of time, bandwidth and good response time. For NAO robot the feedback must be sent in less than 6 milliseconds [5].

![Fig.2. NAO joints and the degrees of freedom](image)

The robot is equipped with two processors, each with different roles. The first processor, equipped with a Linux operating system, is the main controller of the robot and the place which runs the main program. This microprocessor communicates with another lower level microprocessor, which is located in the robot torso and to process data from different sensors and motor encoders in the joints. This information is sent from the microprocessor to device controllers (dsPIC). In the case of joint motors, these controllers have a PI control law implemented. The lower level control is achieved at one millisecond. The upper level control loop runs every 20 ms, and data from sensors are updated every 20 ms. A representation of the entire system is shown in Fig. 3.

In terms of robot motion, the motion is divided into several stages. A humanoid robot walk is omnidirectional. Thus, by generating the motion trajectory, the foot position and orientation will follow this trajectory. Based on the step configuration the preview controller will calculate robot stability using method ZMP and will generate center of mass motion in relation to its motion. Using step position and calculated center of mass the leg motion can be calculated through inverse kinematics [6][7].

Although ZMP method is not suitable for dynamic environment where trajectories of motion change when something new occurs, it can be used if a controller that estimates the next steps is provided. The controller generates the trajectory of the center of mass by analyzing the next steps allowing better reaction to any changes with the robot maintaining its stability.
The controller analyzes the next steps based on step generation and calculates the optimal center of mass (CoM) that will keep the robot stable. Step references are provided by the step generator. This way the steps that are no longer important are being eliminated and the next steps are analyzed in order to achieve a preview of the motion resulting a walk that can adapt dynamically to the events that may arise while moving in the work space [8][9][10].

Using this method Nao robot control is based on the prioritization of commands, the most important being the balance control (COG Jacobi) and the least important is joint space control. In order to generate the walk trajectory, the motion trajectory for each leg is generated achieving robot motion, balance foot trajectory and the motion in joint space. Center of mass is calculated using the 3D model of the inverted pendulum, the robot being able to move forward and follow a circular path using open loop stabilization based upon the computing of center of gravity Jacobi.
In order to put the movement of a robot in the virtual/real environment to the test, the NAO robot should detect and overcome an obstacle placed on his path. When detecting the obstacles the robot should stop, analyze the possibilities of overcoming the obstacle depending on the size of the obstacle and to apply the best trajectory to pass by over the objects in his path. It should be mentioned that the obstacle has been processed and stored in advance in the robot database meaning that the robot knows the object properties when detects it.

3. EXPERIMENTAL PLATFORM

Because in the future the robot has to move independently in the environment, obstacles will not be pre-processed but analyzed in real time, the decision will be taken based on a decision making algorithm to be implemented (Fig. 5). The decision making algorithm will be implemented after processing data resulted by implementing fuzzy logic, Neutrosophic logic and other algorithms to be analyzed.

In our experimental setup, the robot motion was divided in several stages. In the first stage it went on a straight line until it met an obstacle. When detecting the obstacle the robot stops. This period of time is emphasized in the joint motion graphics by the fact that values no longer varies or are equal to zero at that moment.

After the decision was made the robot overcomes the obstacle on the left side, in our case, so he makes a lateral motion up until the distance between the robot and the object’s middle point is safe for moving.
forward. After this stage, Nao will move in straight line until the obstacle is passed and the robot returns to the initial trajectory by making a lateral movement opposite to the one carried out when detecting the object.

![Hip yaw pitch throughout the entire walk](image1)

**Fig. 7.** Hip yaw pitch throughout the entire walk

When it reaches the end of trajectory the robot performs a 180 degree turn and resumes the motion on the same trajectory, moving back to the starting position and so on. From the picture below we can see that the robot uses its HipYawPitch joint only when a 180 degree turn is performed (Fig. 7).

![Left hip roll throughout the entire walk](image2)

**Fig. 8.** Left hip roll throughout the entire walk

We took into account only four of the six joints because the motion of the other two is not so relevant, the ankle roll being identical and opposite with the hip roll.

![Left hip pitch throughout the entire walk](image3)

**Fig. 9.** Left hip pitch throughout the entire walk
In the above pictures the variations of joint angles (in radians) are shown, the data is provided by the motor encoders (blue) and by the reference joint angle which is generated at beginning of the motion (red). The error between these two parameters is a little higher for hip and knee pitch angles (0.05-> 0.1 radians) if compared with the hip roll. This error occurs after the initiation of lateral walk, as it produces some instability in the robot’s position, instability which the robot tries to compensate by increasing the angle in those joints.

4. CONCLUSIONS

The results gave a better understand of the robot walk using a preview controller and of object detection accuracy. Based on this initial results a more complex strategy for object detection will be implemented after processing data resulted by using fuzzy logic, Neutrosophic logic and other algorithms to be analyzed. In the future work the hip and knee pitch should be compensated in order to achieve a better control of these joints. The decision logic that will give a better accuracy for object detection will be implemented in the VIPRO Platform as part of the intelligent interface control module.

ACKNOWLEDGMENT

This work was accomplished through the Partnerships Program in priority fields - PN II, developed with the support of MEN-UEFISCDI, PN-II-PT-PCCA-2013-4, ID2009, VIPRO project no. 009/2014, PN-II-PT-PCCA-2011-3.1-0190 Contract 149/2012 and FP7-PEOPLE-2012-IRSES RABOT project no. 318902.

REFERENCES

2. VLADAREANU L. Versatile Intelligent Portable Rescue Robot Platform through the Adaptive Networked Control, Recent Advances in Mechanical Engineering.
6. XUE F., CHEN X., LIU J., NARDI D. Real time biped walking gait pattern generator for a real robot, RoboCup 2011: Robot Soccer World Cup XV 2011 Jul 5 (pp. 210-221). Springer Berlin Heidelberg 3.
8. GOUAILLIER D., COLLETTE C., KILNER C., Omni-directional closed-loop walk for NAO, Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on 2010 Dec 6 (pp. 448-454). IEEE.