

Using Robot Operating System (ROS) and Single Board Computer to Control Bioloid Robot Motion

Ganesh Kumar Kalyani¹, Zhijun Yang², Vaibhav Gandhi³, and Tao Geng⁴

Design Engineering and Mathematics department, Faculty of Science & Technology,
Middlesex University, London, UK

¹Gk434@live.mdx.ac.uk, ^{2,3,4}[Z.Yang, V.Gandhi, T.Geng]@mdx.ac.uk

Abstract. This paper presents a research study on the adaptation of a novel technique for placing a programmable component over the structural component of a Robotis Bioloid humanoid robot. Assimilating intelligence plays an important role in the field of robotics that enables a computer to model or replicate some of the intelligent behaviors of human beings but with minimal human intervention. As a part of this effort, this paper revises the Bioloid robot structure so as to be able to control the robotic movement via a single board computer Beaglebone Black (BBB) and Robot operating system (ROS). ROS as the development frame work in conjunction with the main BBB controller that integrates robotic functions is an important aspect of this research, and is a first of its kind approach. A full ROS computation has been developed by which an API that will be usable by high level software using ROS services has also been developed. The human like body structure of the Bioloid robot and BeagleBone Black running ROS along with the intellectual components are used to make the robot walk efficiently.

Keywords: Robot; Robot Operating System (ROS); Single Board Computer (SBC); system integration.

1 Introduction

There are different facets of a walking robot namely; structural components, sensing components and decision making components. The structural components include a manipulator (links, joints, etc.), the end-effector and actuators (servo motor, stepper motor, pneumatic and hydraulic cylinder etc.). Sensing components are devices that are meant to gather information about the internal state of the robot or to communicate with the outside world. The robot's decision making components include a processor (which is the brain of a robot that calculates the motion and velocity of robots joints), controllers that control and correlate the motion of actuators, and software (tools and libraries for conveying and collection of routine information) [1].

The walking humanoid robot requires the provision of a system integration arrangement within its decision making, sensing, and structural components. There has to be a centralized or a distributed control system which can be used in such cases. In the centralized setup, the main computer is normally aware of the sensory information around the robot body. At the same time, it becomes inconvenient to increase the number of joints as the main computer has to bear much calculation burden. A distributed

system may comprise of many more joints and many peripheral devices such as cameras, wireless LAN, and control area network module [2]. However, the sub-controllers and the communication bus lines between the main computer and the sub-controllers need to be provided. In cases where the main controller used is not a real-time operating system (RTOS) but only a general purpose operating system (GPOS), a Real-Time Extension (RTX) which is a commercial program capable of accessing the hardware directly has to be provided [2]. In this case, the number of joint motor controllers of the robot has to be worked out which depends on the degrees of freedom (DOF).

Generally, a walking humanoid robot is required to be configured with an autonomous motion controller, which takes care of the autonomous motion processing, e.g., integrating the odometer and the map information to guide the robot. Apart from this, the autonomous processor controls and directs the hardware while image processing tools like Open CV (intel C/C++ libraries for computer vision) or Direct show (Microsoft Multi-Media Development Tool) can be used for accessing images from the camera connected to the USB, and the program provided in the tools can carry out image processing [2]. The motion controller includes walking program and special action program written in suitable programming languages for e.g., C, C++. In the work presented in this paper, a robot's autonomous motion controller is designed using Robot Operating System (ROS) while keeping the basic and fundamental requirements of a prototype humanoid robot, the bioloid humanoid robot. The adapting strategy of the decision making component and ROS with BBB in the structural body of humanoid robot has been redesigned and implemented in this paper. ROS gives standard robot services such as hardware abstraction, low-level device control, implementation of commonly used functionality, topics and message services between processes, and package management. ROS based topic transmission and frame transform procedures are represented in a graph architecture where processing takes place in nodes that may subscribe and publish the sensor, pose, control, state, planning and other messages. In spite of the significance of unified software structure in robot control, ROS itself is not a real time software package. However, it is possible to combine ROS with real time hardware ROS is running on to form a real time framework. This framework can serve as an operating system and middleware for service robots. It gives not just standard operating services (hardware abstraction, contention management, process management) but in addition high level system functionalities (asynchronous and synchronous operations, centralized data base, a configured robot system etc.) [3].

A collection of nodes and programs called roscore are the pre-requisites of any ROS-based system. There must be a roscore running in order for ROS nodes to communicate [4]. Communication buses named ROS Topics are used while using the ROS modules. These comprise of anonymous publish/subscribe semantics that in-turn decouple the production of information from its consumption. Generally, the nodes are unaware of the extremities with which they are communicating. The nodes that are in search of data subscribe to the relevant topics i.e. the nodes that generate data publish to relevant topics. Thus, there can be multiple subscribers and publishers to a topic [5]. An `rqt_graph` is commonly used as a GUI plugin for visualizing the ROS computation graph [6].

The remainder of the paper is organized into three sections. Section II describes the theoretical considerations to be kept in mind while formulating the scheme for adaptability of installing ROS based autonomous motion controller into a walking humanoid robot. Section III describes the original decision making components of a walking humanoid robot (Robotis Bioloid) as well as that of the replaced robot autonomous motion controller (BeagleBone Black). In this section, the details of a ROS based motion control strategy are given. Section IV narrates the exchanging strategy that was adopted to address the motion control issue. Section V concludes the paper.

2 Theoretical considerations

2.1 General points regarding controllers

1. Powerful processor for providing Artificial Intelligence and overall control, while basic microprocessor as I/O managers must be used in walking humanoid robots.
2. Preferably use three-layer configuration:
 - (a) Top layer - Powerful PC as the “brain” of the robot
 - (b) Middle layer - Multiple sub-system managers for motion control, sensor management and processing.
 - (c) Lower layer - Large number of low performance microprocessors for managing miscellaneous tasks.
3. Basic microprocessor can be plugged in to a USB on the computer and it can send and receive commands over serial port and all other higher-level instructions could run on the computer. The computer could issue commands by taking decisions and the other processor could control the motors to perform the required tasks.
4. Decisions on processor can be taken considering the following:
 - (d) Processing Power - If controlling the motors is only required i.e. all processing is done on an external PC, then a motor driver circuit may be enough. For basic on-board processing like transformation of coordinates, simple command loops and inverse kinematics, a basic microprocessor may be sufficient. If automating the robot and implementing advanced features is desired, BeagleBone Black, Raspberry Pi or Intel Atom may be sufficient.
 - (e) Programming ability - Basic Python and C/C++ is required for BeagleBone Black, Raspberry Pi and Intel programming. However, wherever necessary, libraries must be included in addition.
 - (f) Compatibility - The chosen processor should support (both number and type) the motors used [7].

2.2 General points regarding the Software

1. ROS, a comprehensive robot-related software framework depends on the machine controller that integrates robotic functions [8].
2. Python API allows either to use all of the C++ from a remote machine or create Python Modules that can run remotely or on the robots.

3. Embedded software, running on a motherboard located in the head of the robot allows autonomous behaviour. Desktop software running on a computer located outside the robot, allows creation of new behaviours and the remote control of the robot [9].

2.3 Preliminary Adaptability Analysis

The above theoretical considerations have been carefully taken into account and a ROS based architecture is used for data acquisition for motor actuation and decision making processes of the robot. The BeagleBone Black single board computer (SBC) is used in place of the CM530 decision making component of the Bioloid humanoid walking robot (cf. Table 1).

3 Decision making component

The original main control unit of the humanoid robot kit is CM-530 controller. AX 12A Dynamixel servo motors, the gyro/accelerometer sensor, infra-red and distance measuring sensors also all communication devices needed (RoboPlus software) are easily connected to CM-530 (cf. Fig. 4). Different components in the system have their own identifications (IDs) so that the topics or messages can be addressed to specific target components. Position and speed of walking can be easily obtained from the integral motor Dynamixel AX 12A and its encoders. This robot can provide feedback for angular position, angular velocity and load torque. A number of Dynamixel units can be connected through daisy chain to save resources [10].

3.1 BeagleBone Black Processor

BeagleBone Black is a powerful processor that is used in the humanoid robot presented in this paper. The structural platform of the robot is constructed using 2 legs and 12 servos so as to have 12 DOFs. The BeagleBone Black is connected to the robot through software for controlling its motion and function [9] [11]. The complete interface of the BeagleBone Black with robot and computer is shown in **Error! Reference source not found.**

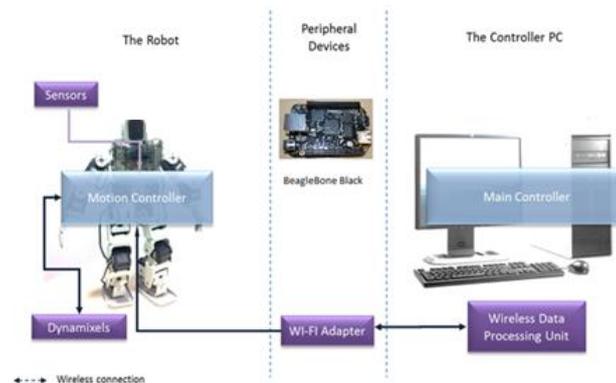


Fig. 1. Interfacing of the robot with computer [8]

Gyro/accelerometer sensors are mounted around the middle portion of the body. This senses the robot's position and controls its balance for walking, thus preventing it from falling. A wireless network is used to provide communication between the computer and the robot. The robot is controlled through the computer instructions and feeds back the data streams from the IR sensor for detection of obstacles in the front of the robot. Thus, the IR sensor acts as a guide for the robot in assisting it in its forward movement and avoiding collision with potential obstacles. This approach can be further used in Brain-computer interface related applications [12].

4 Exchange strategy for the adaptation process

4.1 System modification and Walking Methodology

In this paper, a Debian image with ROS indigo version implementation is installed on the BeagleBone Black. In order to maneuver the robot without interruption, a Wi-Fi module is enabled in the BeagleBone Black using necessary libraries and repositories. A USB hub is connected to the serial port of the BeagleBone Black to which all peripherals are connected. USB2Dynamixel [10] is a device to operate the dynamixel actuators which is connected through the Wi-Fi interface of the BeagleBone Black. Gyro/accelerometer sensor is connected to serial clock and serial data pins while infra-red sensor is connected to the analog to digital converting pins (ADC) of the BeagleBone Black. Wi-Fi adapter is mounted on the humanoid robot and is connected through the serial interface of the BeagleBone Black. The interfacing of the sensors with robot movement is shown in the Fig. 2 and Fig. 3.

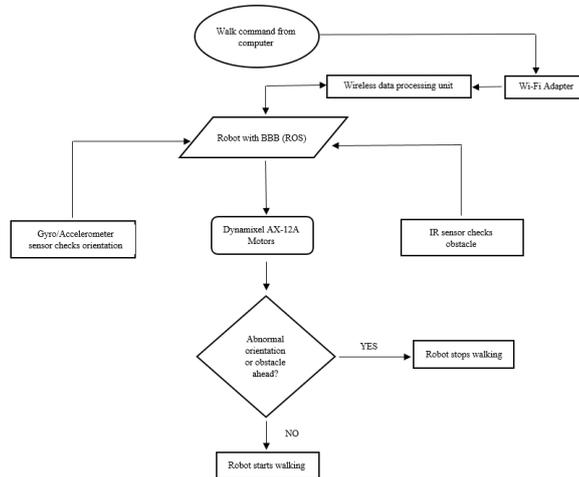


Fig. 2. Sensor Interface for robot movement

For controlling the robot movement, software code is written in python so as to control each servo. The humanoid robot is made to move forward at a desired speed as per the operator's instructions. The walking pattern of the robot is fine-tuned so that the robot maneuvers in small steps and take care of the balancing strategy [14]. The most important challenge in walking the robot is to prevent it from toppling over when one leg is lifted. This is overcome by using the accelerometer sensor. The robot is made to tilt on one side such that its center of gravity is in correct position so as to prevent it from falling.



Fig. 3. Figure showing the BBB, AX-12A motors, USB2Dynamixel connector, Wifi Adapter, IR Sensor and the Gyro+Accelerometer sensor used in the experiment.

The IR sensor detects an obstacle and send a feedback to the robot actuators to stop the robot in a balanced standing position. This is another important improvisation in the robot where it does not stop abruptly, and thus any chances of falling down while executing the halt operation are avoided. Keeping these basic and fundamental design requirements of the humanoid robot, the adapting strategy is evolved. Deciding cum enforcing components of the bioloid robot are replaced under this strategy with the BBB such that the biped robot performs walking, which was not present earlier. This enhancement was made possible using ROS and Python.

In the bioloid robot, the CM530 controller was the key for governing all the robotic movements and the control of the motors (cf. Fig.4). The idea was to replace this default controller with a different controller which can be easily programmable and can make the working of the robot more efficient. The deciding cum enforcing intelligence providing components of the BBB Board (incorporating Robot Operating System (ROS) and Python libraries and walking control) are implanted in the bioloid robot. While carrying out this process the original decision making components are removed from the bioloid robot. This approach has been implemented and shown in this paper. Such an installation procedure has been undertaken for the first time as far as the researchers' knowledge goes. The main components replacement is shown in Table 1.

The Bioloid robot gaits are re-designed in the way that the robot body leans to the left/right direction when its right/left leg lifts and move forward. This simple design effectively avoids the collision between the robot feet with the ground. A check is also made to see that the robot is in a stable position, which makes it to avoid falling. If the

parameters are more than the prescribed limits inscribed in an accelerometer sensor then the robot is made to tilt the other side to counter balance in order not to fall. The angle to which the robot leans is governed with the feedback from the accelerometer sensor. This angle is 20 degrees. If the robot tilts beyond 25 degrees then it would come back to its standstill position and stops further movement. When the robot moves forward, the IR sensors sense obstacles in front of the robot. If an obstacle is detected to be present at around 35cms then the robot stops moving further and comes to its standstill position. This dynamic model can be used, as a building block, to actuate the motors on legs and joints for a swing-stance period of the legs for its further movement [3]. At the same time, the robot stops further movement, and when the obstacle is removed, the robot continues its motion. Using the onboard IR sensor, the robot is allowed to navigate in places and a continuous feedback is sent from the IR sensor. The position of the pelvis center and the ankle in the view point of sagittal plane is as shown in Fig. 5 and Fig. 6. For the moment, the maximum walking speed of the robot is 0.5

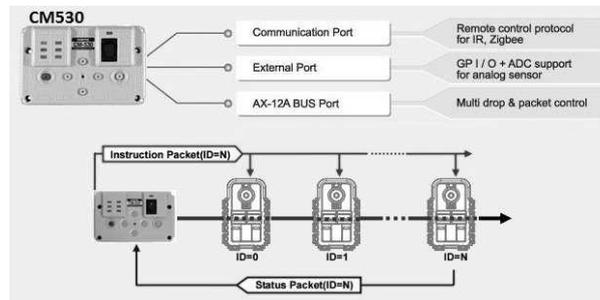


Fig. 4. Decision making Components of Bioloid Humanoid [10]

Table 1. Exchange of components [10], [11]

Sr. No.	Particulars	Existing Component	Replacing Component
1	CPU	STM32F103	AM33588
2	Controller	CM530	BeagleBone Black
3	Remote Controller	RC100	Wi-Fi
4	Software	RoboPlus with C++	ROS with Python

foot/second. Beyond this limit the robot may get unstable and falls down. Slippery or uneven surfaces also affects the robot and possibly make it fall down.

The RoboPlus software of the Robotis Bioloid robot is an icon type C-language based software meant for easy programming and managing motion and behavior. In the work presented in this paper, this component has been replaced with ROS and Python scripts for a more flexible control of the autonomous robot movement. ROS is a promising robot software environment due largely to its rich open source repository and innovative compilation and operating structure. This demo work provides a proof of the concept framework in which a simple ROS motion control scheme is used. One Python

ROS node is created to control the robot motion and publish one topic on the robot status. Based on this work, more potential nodes can be added for signal processing and motion control.

The biped walking gaits are imbibed to this revised hybrid humanoid robot using

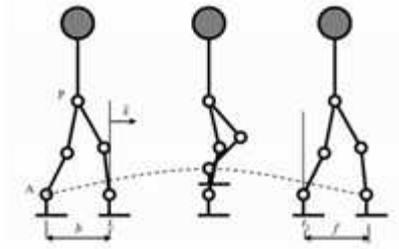


Fig. 5. Sagittal view for the walking pattern

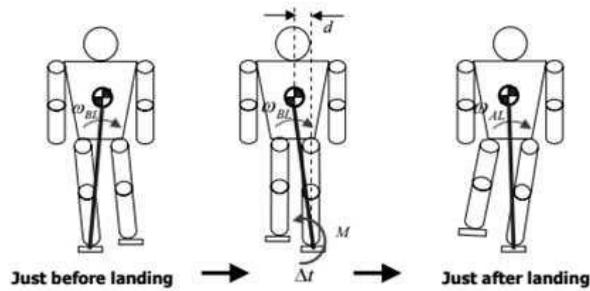


Fig. 6. Schematics of landing position control

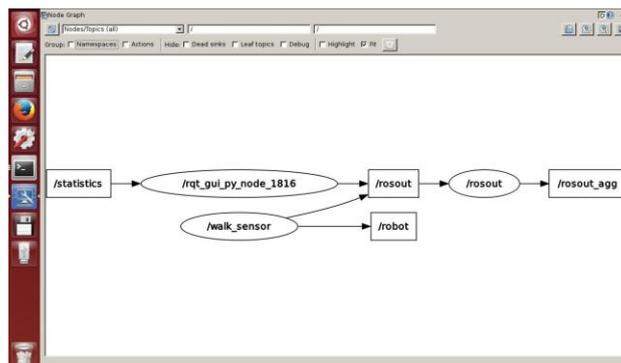


Fig. 7. Rqt_graph representing the dynamics of the system where the rosnode ‘/walk_sensor’ communicating on the ROS topic ‘/robot’.

ROS and Python libraries, walking algorithmic codes and controls for the first time in

this research study. During robot walking, ROS acts as the intermediate communication between the computer and the BBB. ROSCORE is the first program that had been run in one terminal when using ROS. ROSRUN is the command that instructs ROSNODE (roscnode displays information about the ROS nodes); which is the '/walk_sensor' here to publish on the ROS topic '/robot' (cf. Fig. 7). Thus, dynamic and stable walking is achieved.

4.2 Performance trials

Trials have been carried out using BeagleBone Black to access and control the motion of individual AX-12 Dynamixel servo motors. The accelerometer sensor is added to the interface so as to control the servo motors. The Bioloid humanoid robot structure that is incorporated with BeagleBone Black and ROS accesses individual servos and accelerometer sensor to make the robot walk with perfect balance and gait. Additional libraries are added so as to enable the BeagleBone Black with Wi-Fi adapter and IR sensor. IR sensor is mounted in front of the robot to send real time information to the computer. An obstacle in front of the robot is sensed using IR sensors, which stops further movement of the robot. Trials have been carried out on all these sensor functions, so as to enhance the capability of the Bioloid humanoid robot.

A video showing a demonstration of the work presented in this paper can be accessed in [15]. As perceived in the video, the robot is seen walking as described theoretically and makes good use of its sensors for walking and navigation, and does not trip over. Also, different walking speeds for the robot have been implemented namely, fast and slow. The robot maintains its balance very well in both the mentioned walking speeds.

5 Conclusion

As far as the authors are aware, this project study is the first of its kind, wherein a biped robot is controlled using ROS and BBB. In what is referred to as the robot brain, the main BBB controller, and the ROS software has been successfully implemented in the structural body of the bioloid walking humanoid robot. The initial performance of the Bioloid robot, with its revised brain i.e., the controller and the software seems promising. This transformed Bioloid humanoid walking robot is much enhanced in terms of its autonomous behavior based on the sensed data.

Although, initial tests prove the approach to be reasonably robust, and integration of the hardware with the software is successful, further room for improved performance still exists. One of the improvements that can be proposed here is the implementation of a USB camera along with some image processing in the software. This improvisation would further enhance the performance of the robot.

6 Acknowledgment

The project is funded by an EPSRC grant EP/P00542X. The authors would also like to thank School of Science and Technology, Middlesex University for funding this project.

References

- [1] A.Gandhi, Basics of robotics., February 24,2013
- [2] J.H.Kim, I.W.Park and J.H.Oh Design and walking control of the Humanoid Robot KHR-2 KAIST Humanoid Robot 2, ICCAS2004 August 25-27,
- [3] Kay, Jackie. "Proposal for Implementation of Real-time Systems in ROS 2". [Online]; http://design.ros2.org/articles/realtime_proposal.html
- [4] ROSCORE. <http://wiki.ros.org/roscore>, Oct. 2016.
- [5] ROS TOPIC <http://wiki.ros.org/Topics>, Oct. 2016.
- [6] Rqt_graph. http://wiki.ros.org/rqt_graph, Nov. 2016.
- [7] D.Paramkusam Things to consider before deciding on the processor, March,25,2016
- [8] T.Foote Visualizer of delta robots using ROS and EtherCAT, ROS Robotics News: April,15,2015
- [9] NAO Software 1.14.5 Documentation. Python SDK and software in and out of the robots, 27,06,2013
- [10] ROBOTIS e-Manual v1.25.00 , v1.27.00, ROBOTIS 12-06-2015
- [11] R.Grimmett Mastering BeagleBone Robotics, Published by Packt Publishing Ltd. December 2014
- [12] V.Gandhi, EEG-Based Mobile Robot Control Through an Adaptive BrainRobot Interface, Vol.44 Sep 9,2014
- [13] J.Y.Kim, I.W.Park and J.H.Oh Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor, Journal of Intelligent and Robotic Systems April 2007, Volume 48, Issue 4, pp 457-484
- [14] Z.Yang Dynamic control of walking leg joints: A building block model perspective, ICNC 2011: 459-463.
- [15] <https://www.youtube.com/watch?v=f4AoTJh2cXg>