*Vietnam Journal of Science and Technology* ***60*** *(6) (2022) 1162-1178*

doi:10.15625/2525-2518/16891

# CONTROL SYSTEM ARCHITECTURE OF AN INTELLIGENT HUMANOID ROBOT

**Nguyen Dinh Quan[1, *], Ha Huy Hung[1], Le Dinh Son[2], Tran Van An[2], Nguyen The Hung[3], Cao Viet Linh[4]**

[1]*Faculty of Aerospace Engineering, Le Quy Don Technical University, 236 Hoang Quoc Viet, Bac Tu Liem, Ha Noi, Viet Nam*

[2]*Faculty of Information Technology, Le Quy Don Technical University, 236 Hoang Quoc Viet, Bac Tu Liem, Ha Noi, Viet Nam*

[3]*Faculty of Technical Management, Le Quy Don Technical University, 236 Hoang Quoc Viet, Bac Tu Liem, Ha Noi, Viet Nam*

[4]*Institute of Radar, Military Institute of Science and Technology, 17 Hoang Sam Street, Cau Giay District, Ha Noi, Viet Nam*

[*]Email: *ndquan@lqdtu.edu.vn*

Received: 22 January 2022; Accepted for publication: 22 July 2022

**Abstract.** This paper presents the hardware and control software architectures of an intelligent humanoid robot. The robot has a mobile base consists of three omnidirectional wheels that allows it to move freely with three degree-of-freedom (DOF), two 6-DOF arms and 3-DOF neck and head that allows it to perform most of the common movements of human. Detail hardware components are given to show our mechanical design solution of the robot. The control software structure of the robotic system is constructed in the robot operating system (ROS) framework which is mainly used as a bridge to connect the control modules and various peripheral devices to ease our robot system task management. We have also shown the detail structure of the robot control system which consists of all key control  modules which enable the robot functions: from upper level with AI-based techniques such as image and sound processing to middle level with the robot motion controllers and then to the lower level with the management of atuators and sensors. The proposed architecture is being developed and tested on a real humanoid robot prototype called Bonbon to support Enghlish teaching in elementary schools.

*Keywords:* humanoid robots, control system architecture, social robots.

*Classification numbers*: 5.3.9.

## 1. INTRODUCTION

System architecture design is an important part of the robot system development [1]. One polular solution in delveloping the software control of intelligent robots is to integrate the reactive actions and schedulings such as RAPs system (Reactive Action Packages) in a three-layer structure where the middle layer is the RAP system [2 - 3].

In [4], Bonasso proposed a system architecture which was constructed from the bottom layer where robot behaviors were programmed by Rex language as synchronization networks [5]. All calculation model on computer using Rex language guaranteed the consistency of the internal state variables of the robot systems and external variables of surrounding environment. The middle layer of the structure was a chain of conditions implemented using GAPPs (Goals as parallel programs) [6] that allowed users to activate and stop Rex models until the robot tasks were completed. The chains of sequential actions based on GAPPs can be synthesized from many traditional task planners [7].

The study in [7] presented a three-level (3L) structure including: planning, executive and behavior control. The 3L structure has been used on many robot generations [8] up until now. Several popular intelligent control systems that were built based on 3L structure can be listed such as ATLANTIS [9], Saridis [10] or LASS [11].

To implement the robot control system, it is also imporant to select suitable programming framework. For industrial applications, realtime systems such as VxWork, B&R or QNX have provided reliable solutions for many robotic systems for several decades. However they are comercialized and lacks the ability to easily integrate third-party devices to the robotic systems. In recent years, the robotic commnunity has been shifted to use robot operating system (ROS) [12], a powerful open source framework that supports many devices including sensors, actuators and advances computation packages and allows users to integrate their own libraries into the robotic control systems.

In recent years, humanoid robots have been received worldwide attention. They have appeared in many application fields. Some well-known humanoid models with walking gaits can be listed such as: ASIMO [14-15], HUBO [16], ATLAS [17], REEM-C [18]... Wheeled humanoid robots are also widely developed. Among them PEPPER is a famous model [19-21]. It is one of the world's first social humanoid robot able to recognize faces and basic human emotions. The PEPPER's control system was developed on NAOqi framework [22], a cross-platform which supports cross-language with capability of parallelism, resources, synchronization and events. However the number of modules in NAOqi is limited. Due to the expanding of ROS, NAOqi API is also made available to work with ROS.

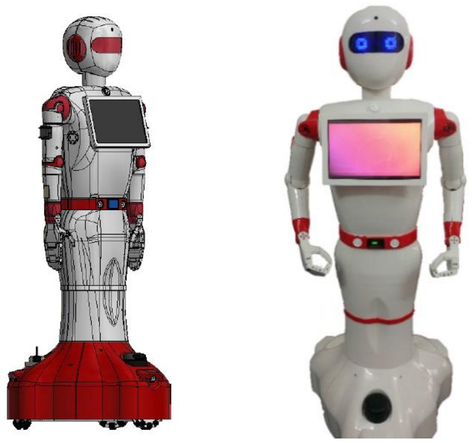
In this paper, we will present the hardware and control software architectures of our new wheeled humanoid intelligent robot called Bonbon. On the mechanical design, we only inherit the general structure of the PEPPER humanoid robot and develop our own hardware design solution. We have also developed our own control system architecture based on the 3L structure. Most of the control software modules are integrated in ROS framework for a fast and flexible integration solution. Besides the ROS network, we utilize event-based threads to manage important control processes such as task scheduling, motion control of the robot mobile base and upper body parts.

The structure of the paper is divided as follows. Section 2 presents briefly the Bonbon humanoid robot and its kinematic properties including major mathematic modeling for the robot motion control problems. Section 3 presents our solutions on the hardware and software system of the robot where we describe in detail not only the general structure but also descriptions for important aspects of the hardware and software control system as well as the robot abilities in order to act properly in social environments.

## 2. THE BONBON HUMANOID ROBOT

Figure 1 shows the general 3D model and prototype of the Bonbon humanoid robot. It has a mobile base with three omnidirectional wheels that allows the robot to move freely in three degrees of freedom (DOF). The robot upper body has two 6-DOF arms so that the robot can imitate most common actions of a person in normal conversations. There are 3 motors integrated in the neck that helps the head can move in 3-DOF orientated motions.

The humanoid robot is intended to use to support English teaching in elementary schools. So it should be equipped with the ability to understand and interact well in human environment.



(a) 3D model (b) Real robot

*Figure 1*. The Bonbon humanoid robot.

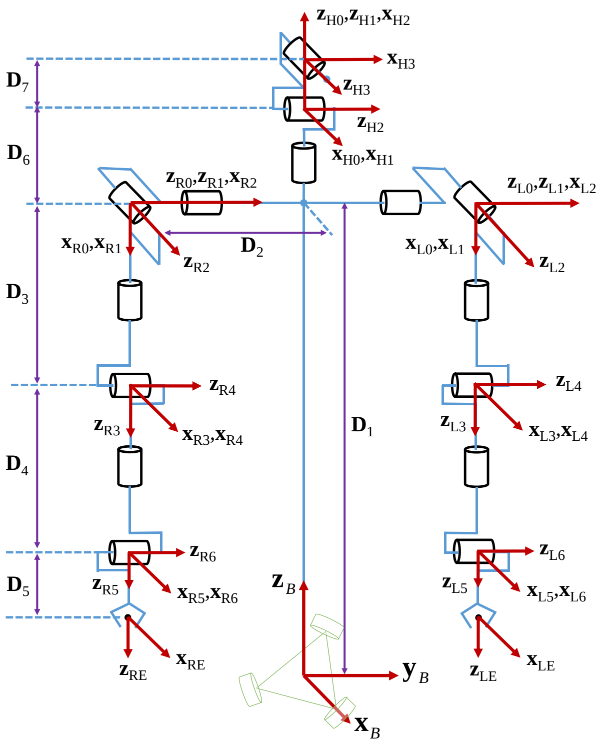



*Figure 2*. Geometry structure of Bonbon robot.

Figure 2 shows the geometric structure of the Bonbon robot. The robot body can be separated into two main parts:

- Mobile base: three omnidirectional wheels
- Upper body: tree structure including the two arms, the neck and head.

### 2.1. Kinematic model of the mobile base

Table 1 shows the kinematic parameters of the mobile base. The base geometry is illustrated in Figure 3.

*Table 1*. Kinematic parameters of the mobile base.

| Wheel | $\alpha$ | $L$ | $\varphi$ | $r$ |
|---|---|---|---|---|
| 1 | 0 | $L$ | $\varphi_1$ | $r$ |
| 2 | $2\pi/3$ | $L$ | $\varphi_2$ | $r$ |
| 3 | $-2\pi/3$ | $L$ | $\varphi_3$ | $r$ |

Here, $\alpha_i$ is the angle between the base frame axis $X_B$ and the axis of wheel $i$, $L$ is the distance from the center point of the mobile base to the center point of each wheel, $r$ is the wheel radius, $\varphi_i$ is the joint position of wheel $i$, $\theta$ is the heading angle of the mobile base with respect to the global coordinate frame.

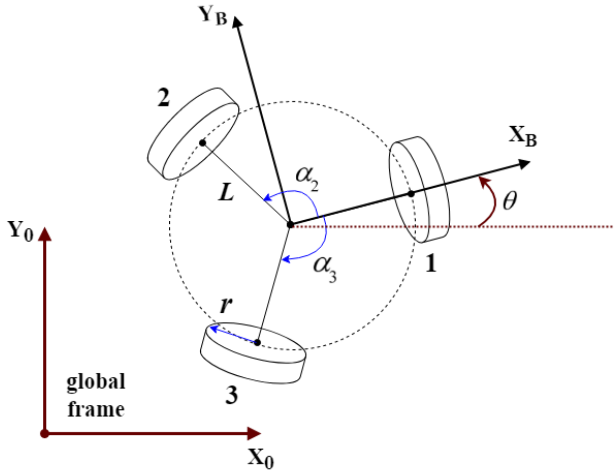



*Figure 3*. Geometric structure of the robot base.

The kinematic model of the base can be computed from the non-slipping kinematic constraints as follows:

$$\mathbf{J}_1\mathbf{R}(\theta)\dot{\mathbf{q}} - \mathbf{J}_2\dot{\boldsymbol{\varphi}} = 0 \tag{1}$$

where $\mathbf{q} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ and $\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$ are the position and velocity vectors of the robot base expressed in the global frame, $\boldsymbol{\varphi} = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}^T$ and $\dot{\boldsymbol{\varphi}} = \begin{bmatrix} \dot{\varphi}_1 & \dot{\varphi}_2 & \dot{\varphi}_3 \end{bmatrix}^T$ are the position and angular velocity vectors of the omnidirectional wheels, and:

$$\mathbf{J}_1 = \begin{bmatrix} 0 & -1 & -L \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & -L \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & -L \end{bmatrix}, \ \mathbf{J}_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}, \quad (1)$$

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

From these relations, we have the direct kinematic model of the mobile base:

$$\dot{\mathbf{q}} = \mathbf{J}\,\dot{\boldsymbol{\varphi}} \quad (3)$$

and the inverse kinematic model of the mobile base:

$$\dot{\boldsymbol{\varphi}} = \mathbf{J}^{-1}\,\dot{\mathbf{q}} \quad (4)$$

with **J** is the Jacobian matrix of the robot base:

$$\mathbf{J} = \mathbf{R}^{-1}\,\mathbf{J}_1^{-1}\,\mathbf{J}_2 \quad (5)$$

The matrix **J** only depends on the heading angle $\theta$ of the robot base. The direct kinematic model (4) is used as a basis to estimate the pose of the robot base in global frame, meanwhile the inverse kinematic model (5) is used to compute the control signal for the wheels in motion control problems for the base.

### 2.2. Kinematic model of the upper body

As we have seen in Figure 2, the coordinate frames are constructed by following Craig's rules [13]. The robot frame $\langle \mathrm{O}_B \rangle$ is attached to the robot base.

Table 2 describes the kinematic parameters of one robotic arm where $\theta_j$ is the angle between link *j-1* and link *j* and has the value of $q_j\ (j = 1,..,6)$. The transformation matrix between any two consecutive coordinate frames is:

$$\begin{aligned} {}^{j-1}\mathbf{T}_j &= \mathbf{Rot}(\mathbf{x},\alpha_j)\mathbf{Trans}(\mathbf{x},d_j)\mathbf{Rot}(\mathbf{z},\theta_j)\mathbf{Trans}(\mathbf{z},r_j) \\ &= \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (6)$$

*Table 2*. Kinematic parameters of one robot arm

| $j$ | $\alpha_j$ | $d_j$ | $\theta_j$ | $r_j$ | $\theta_j(t=0)$ |
|---|---|---|---|---|---|
| 1 | 0 | 0 | $q_1$ | 0 | 0 |
| 2 | $\pi/2$ | 0 | $q_2$ | 0 | $\pi/2$ |

| | | | | | |
|---|---|---|---|---|---|
| 3 | $\pi/2$ | 0 | $q_3$ | $D3$ | $\pi/2$ |
| 4 | $\pi/2$ | 0 | $q_4$ | 0 | 0 |
| 5 | $-\pi/2$ | 0 | $q_5$ | $D4$ | 0 |
| 6 | $\pi/2$ | 0 | $q_6$ | 0 | 0 |

The transformation matrix of the robot arm from the base frame to the sixth joint frame is compute as follows:

$$ {}^{0}\mathbf{T}_6 = {}^{0}\mathbf{T}_1(\mathrm{q}_1)\,{}^{1}\mathbf{T}_2(\mathrm{q}_2)\ldots{}^{5}\mathbf{T}_6(\mathrm{q}_6) \tag{7}$$

This matrix ${}^{0}\mathbf{T}_6$ is the same for the two left and right arms of the Bonbon robot. We use the frame $\langle \mathrm{O}_B \rangle$ as the common reference frame for the two robotic arms (left arm and right arm).

The direct kinematic models for the two robotic arms are constructed as follows:

$$ {}^{\mathrm{B}}\mathbf{T}_{\mathrm{R}} = \mathbf{Z}_R\,{}^{0}\mathbf{T}_6(\mathbf{q}_R)\mathbf{E}_R \tag{8}$$

$$ {}^{\mathrm{B}}\mathbf{T}_L = \mathbf{Z}_L\,{}^{0}\mathbf{T}_6(\mathbf{q}_L)\mathbf{E}_L \tag{9}$$

where $\mathbf{Z}_R$ and $\mathbf{Z}_L$ are the transformation matrices between the base frame of right and left arms to the reference frame $\langle \mathrm{O}_B \rangle$ respectively, $\mathbf{E}_R$ and $\mathbf{E}_L$ are the transformation matrices between the sixth coornidate frame of right and left arms to the coordinate frame that attach to right and left hands respectively (subscriptions *R*, *L* indicate the terms belong to the right and left arms):

$$ \mathbf{Z}_R = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -D2 \\ -1 & 0 & 0 & D1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{E}_R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & D5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$ \mathbf{Z}_L = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & D2 \\ -1 & 0 & 0 & D1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{E}_L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & D5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

The direct kinematic models (9) and (10) are used in the motion control problems of the right and left arms.

For the neck and head structure, since the three serial joints $q_{H1}, q_{H2}, q_{H3}$ in fact represent for the three rotating angles yaw, pitch and roll respectively, the motions of the head can then be controlled directly by adjusting the values of these three joints.

## 3. HARDWARE AND SOFTWARE ARCHITECTURE OF THE ROBOTIC SYSTEM

### 3.1. Hardware architecture

Figure 4 shows the main hardware components of the Bonbon robot. Figure 5 presents the robot hardware flowchart. The mobile base is equipped with three BLDC motors. Each 6-DOF robot arm has six DC servo motors. The robot neck has three DC servo motors. All the motors are selected with suitable power, output speed and output moment to enable smooth motions for the robot parts on its upper body.

We integrate a mic-array on top of the robot head to get the sound coming from different directions. There are two cameras mounting on the forehead: one is used for communication, one is used for image processing functions. Each robot eye is an 8-by-8 led matrix which is enough to show various basic eye expressions. One stereo camera is mounted under the main display on the robot chest. This camera is used for advance functions such as people tracking and following tasks.

There are two computers in the robotic system. One embedded computer with CPU core i7-8559U is used to manage all motion control tasks as well as safety functions of the robot. One center control computer with a 512-core GPU and a CPU 8-core ARM 64-bit is used to manage all image and voice processing tasks as well as task planning function of the robot.

For safety navigation, we use one laser scanner mounted on the front of the mobile base with a $360^0$ scanning range and several ultrasonic sensors mounted around the base and in front of the robot belly. These sensors allow the robot to be able to move around safely while avoiding static and dynamic obstacles appeared on its paths.

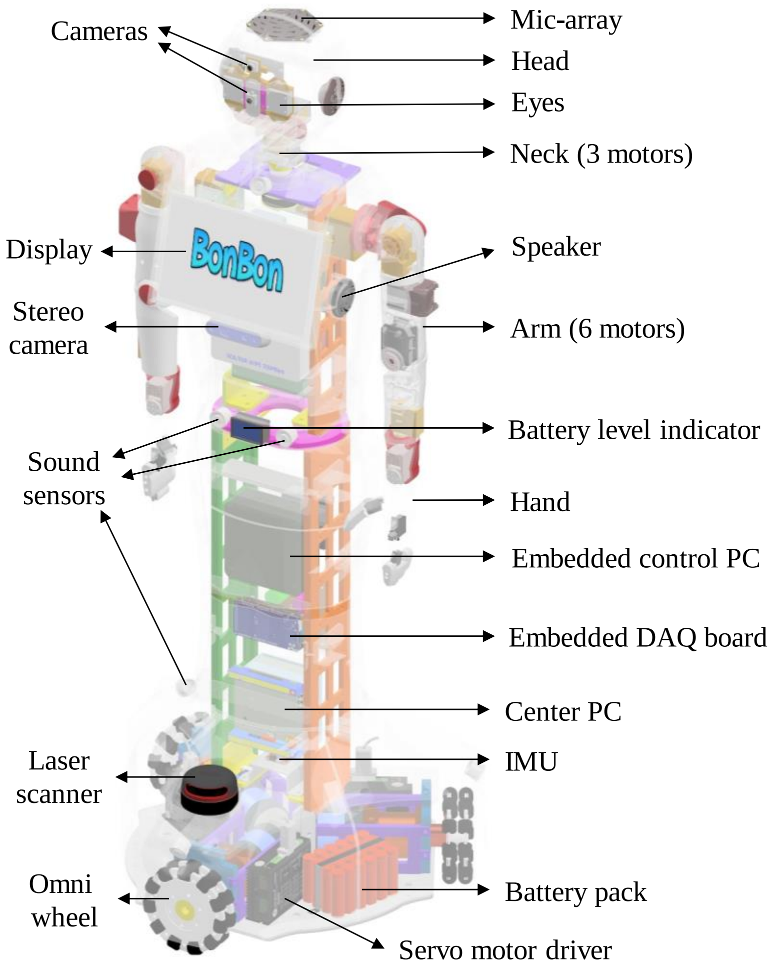



*Figure 4*. Main hardware components of the Bonbon robot.

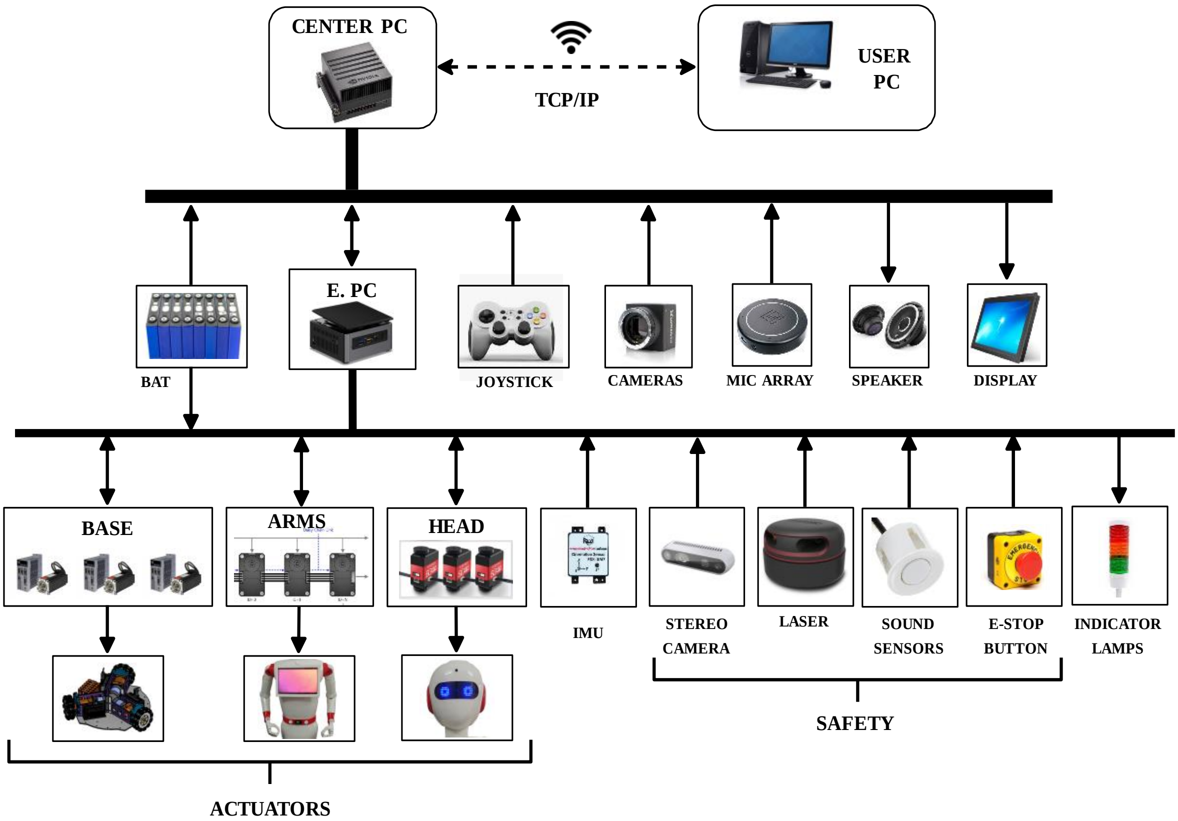



*Figure 5.* Hardware architecture of the Bonbon robot.

### 3.2. Software architecture

Figure 6 shows the overview of the robot control software system. It has four layers:

- **System management**: this layer runs on the remote PC and has three main functions which are data acquisition (getting information on robot states, from cameras and mic-array), system monitoring (monitoring the robot states and all the tasks the robot is performing) and remotely control (manually control the robot from the remote station)
- **Task planning**: this layer run on the center PC, manages image and voice processing tasks. It analyzes the robot situations, forecasts suitable actions and then plans sequence of behaviors for the motion control modules to carry out.
- **Executive**: this layer gets the commands from the center PC, decodes command data and sequences the tasks given and assigned them to specific motion control modules to execute. The executive layer runs on the embedded control PC.
- **Behavior control**: this layer manages all the actuator control loops in the system as assigned by the executive layer. It manages all the peripherals, collects and processes raw input data before sending it to the higher-layer modules.

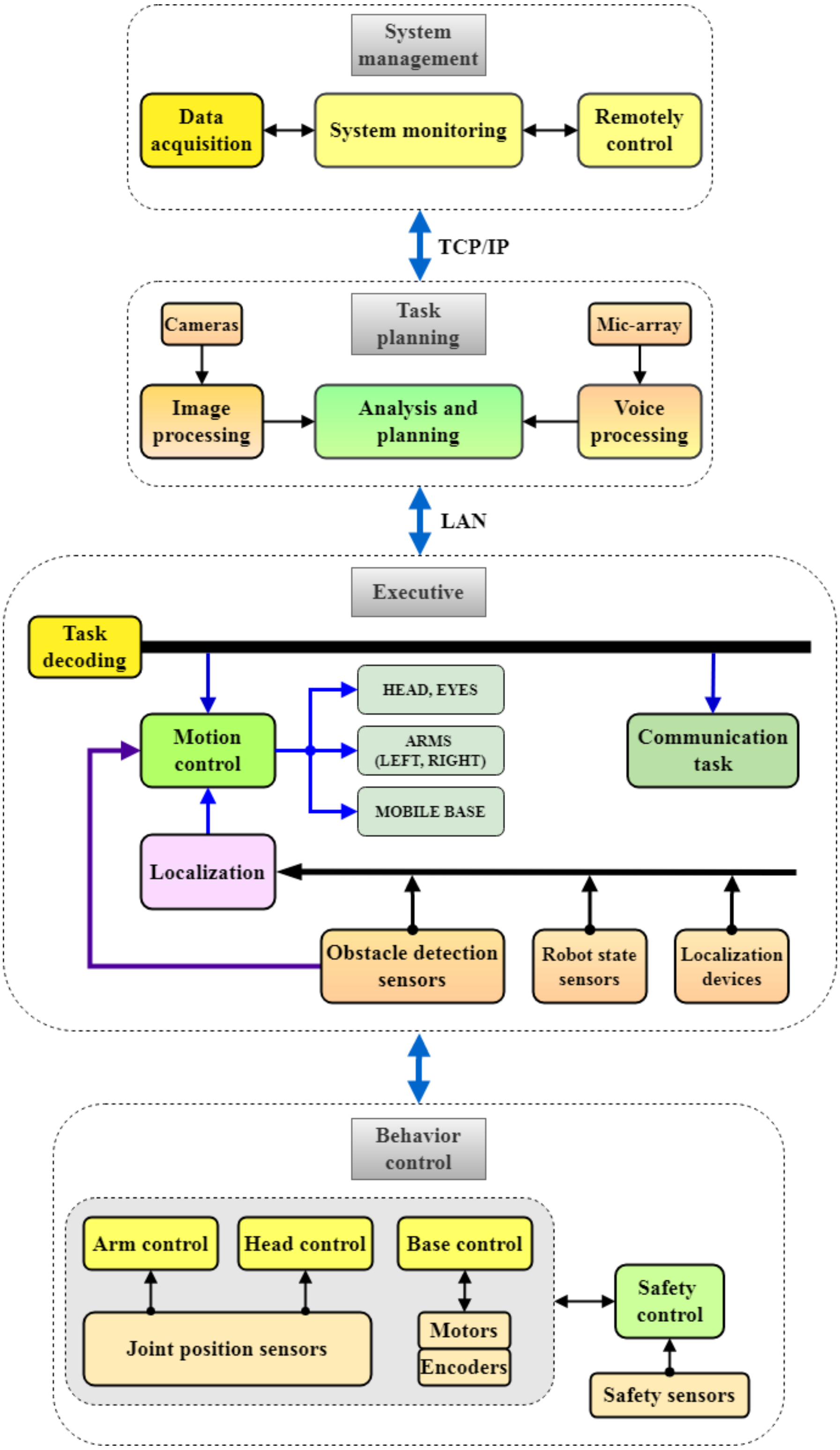



*Figure 6*. Software control system architecture of Bonbon robot.

### 3.3. Developing control software on ROS

Implementing all control algorithms and control modules in a robotic system is not an easy task. For the control system of the intelligent robot in this work, to realize the software architecture as shown in the previous section, we have ultilize several programming techniques such as parallel computing and sequential scheduling. Each control module can be considered an independent process that can run in its own thread. Since different modules have different timing loops, it is important to find a way that can exchange data between processes in realtime. Event-based model is the solution for this.

ROS framework supports the event-based function via the "Subscribe" and "Publish" method. All data between different processes can be exchanged by sending some messages in the ROS network to pre-defined "Topics" [12].

The management of software packages in the same computer can be done efficiently with ROS framework. However, when coming to data exchange remotely between the robot system and the remote control center, it is still possible to use ROS but there are limitations. Because of this, we have developed our own event-based modules for managing tasks by using network sockets such as UDP or TCP/IP to interchange data in the local network or/and through the internet remotely.

Figure 7 presents the main software modules that is running on the two control PCs of the robot. Most of the motion control modules are developed on ROS framework and run on the embedded control PC. In the following parts, we will describe shortly some important function blocks of the robotic control system.

- **Task planning**: this is the center node of the center control PC. It can be considered as the robot brain while handling the planning of all the tasks for the robot to perform.
- **Sound recognition**: this module get the sound from the mic-array (mounted on the top of the robot head) and perform a speech-to-text process. We use Google's speech-to-text API as the core and add our own improvement to increase the accuracy of the API.
- **Face recognition**: this module gets the image stream from one camera mounted on the robot forehead and processes to get the information of the person within the robot field of view including: identity, relative coordinates of the person's face with respects to the robot (this information is used in the face tracking function of the robot).
- **Gesture recognition**: this block identify the gestures of the person that is interacting with the robot. If the person has a special movement (e.g. want to shake hand with the robot), after recognizing the gesture the robot will perform a motion in accordance with the person's gesture. This function helps to increase the ability to react naturally in common situations when the robot is placed in a social environment.
- **Task processing**: this block acts as a scheduler of the tasks that have been sent to the embedded PC from the task planning module. When there is a command being sent from the center PC, the command data is decoded in the task processing module and re-scheduled into specific tasks. Each task can be assigned to the base or to the robot arms and the head which depends on the detail of the task. The task processing node also collects information on the robot states from various nodes (e.g. battery level, joint positions or obstacles) and send these information to the center PC.
- **Motion manager**: this module handles all the motion control of the robot including the base and the upper-body parts (two robot arms and the head). It also manges the teaching function for the robot moving parts in order to create new movements to add to the motion library.

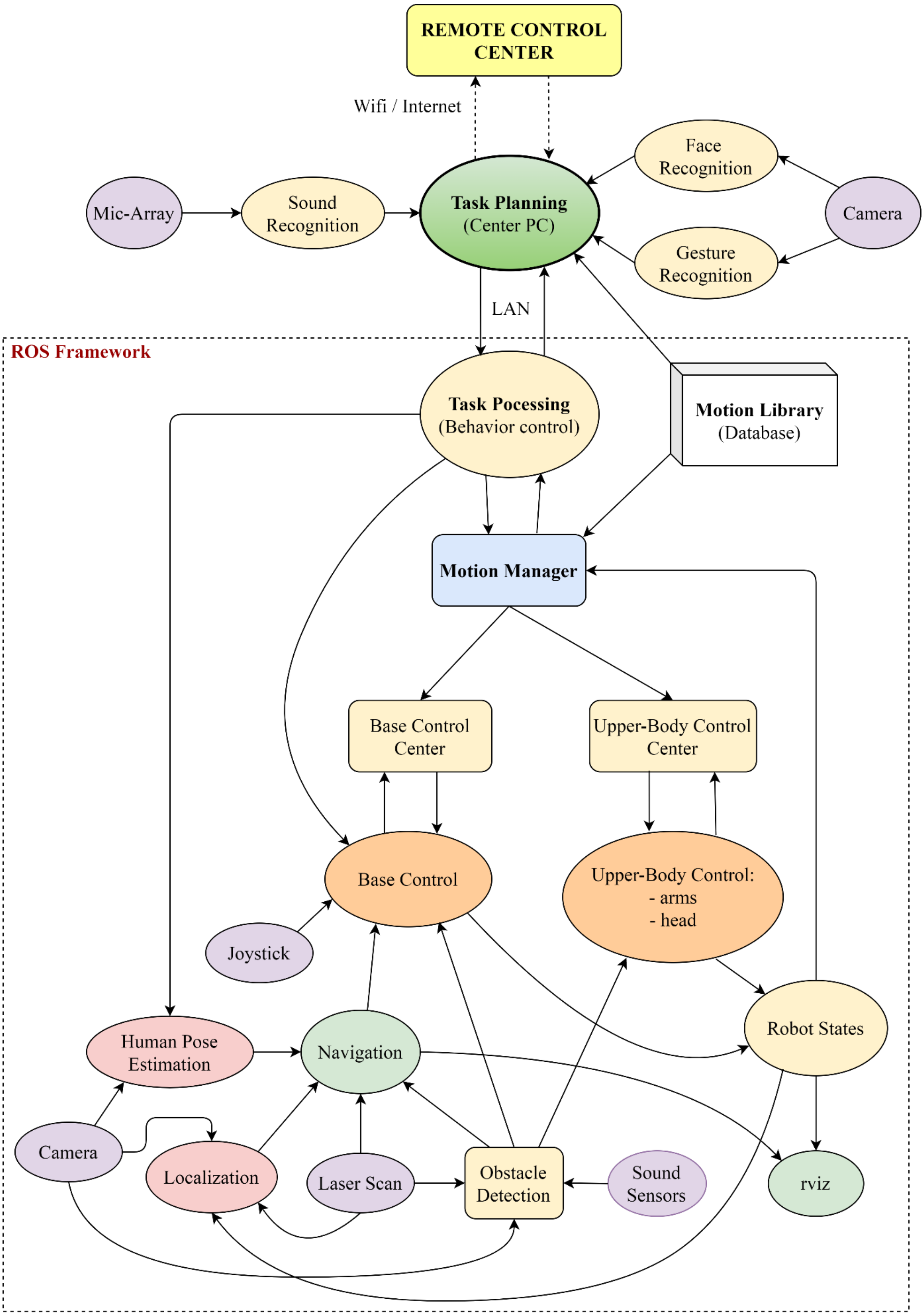
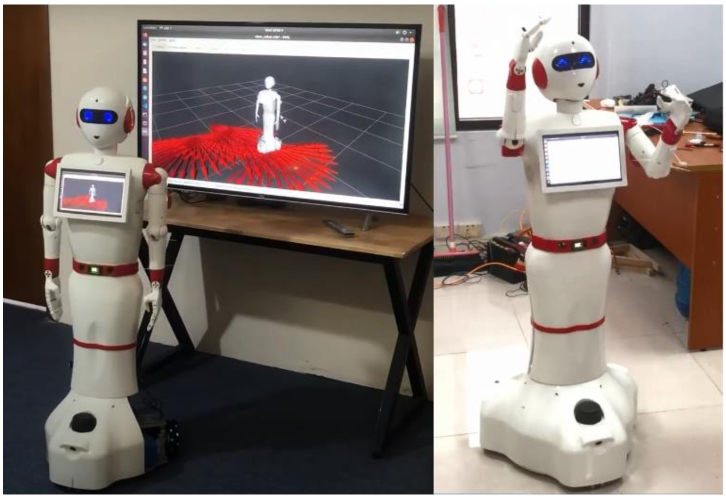



*Figure 7*. Main software modules of the Bonbon robot.



(a) Visualize robot motions in Rviz (b) Bonbon robot is performing a dance

*Figure 8*. Bonbon robot in action.

- **Base control center**: manages the motion of the mobile base including three operating modes: manual mode (signals received from remote PC or joystick), auto mode (perform actions of the base according to the desired trajectory stored in a motion file from the motion library) and navigation mode (perform navigation related task such as moving freely in space, following a person or approaching a person while avoiding all obstacles in the robot's paths).
- **Upper-body control**: manages the motion of the arms and head. Most of the motions come from the motion library. This node is only activated in the "auto mode" when the robot is interacting with people or doing a performance (e.g. talking with someone or performing a dance).
- **Navigation**: this node handles all the tasks related to automatically navigate around. This node takes the information from several sensor sources to compute desired linear and angular velocities for the mobile base including:
    - **Laser scan**: giving information of the surrounding environment on 2 2D plane.
    - **Obstacle detection**: fusion of safety sensors (laser scan and ultrasonic sensors) to create information about all the obstacles around the robot.
    - **Localization**: giving information of the current pose of the robot including its coordinates and heading angle.
    - **Human pose estimation**: giving information of the people that the robot is interacting with such as: relative distances or relative heading angle from the robot to the target person.

Figure 8 shows the real robot Bonbon in action. All the robot motions can be visualized in a Rviz window (see Figure 8a). We can monitor all of the robot states including its coordinates, heading angle of the base, joint positions and sensor feedbacks as well as obstacles appeared in the scanning range of the sensors.

### 3.4. Manage the robot behavior in social activities

When bringing the humanoid robot to the human environment, it is necessary to integrate an ability to formulate a chain of actions for the robot. Building the behaviors for the robot need to consider many factors:

- Safety
- Adaptablility in interactive situations with human
- Continuity in motions

Usually in communication functions (e.g., chatbots), the robot sometimes only need to show a movement at certain moments during the communication process. However, there are scenarios in which the robot need to perform a series of different behaviors continuously to increase the attraction and expression, thereby making a strong impression on the interactors (e.g. see Figure 8b). To do this, it is necessary to have a mechanism to analyze the situation, calculate and give the appropriate sequence of behaviors for the robot to perform while synchronizing with the robot's speech.

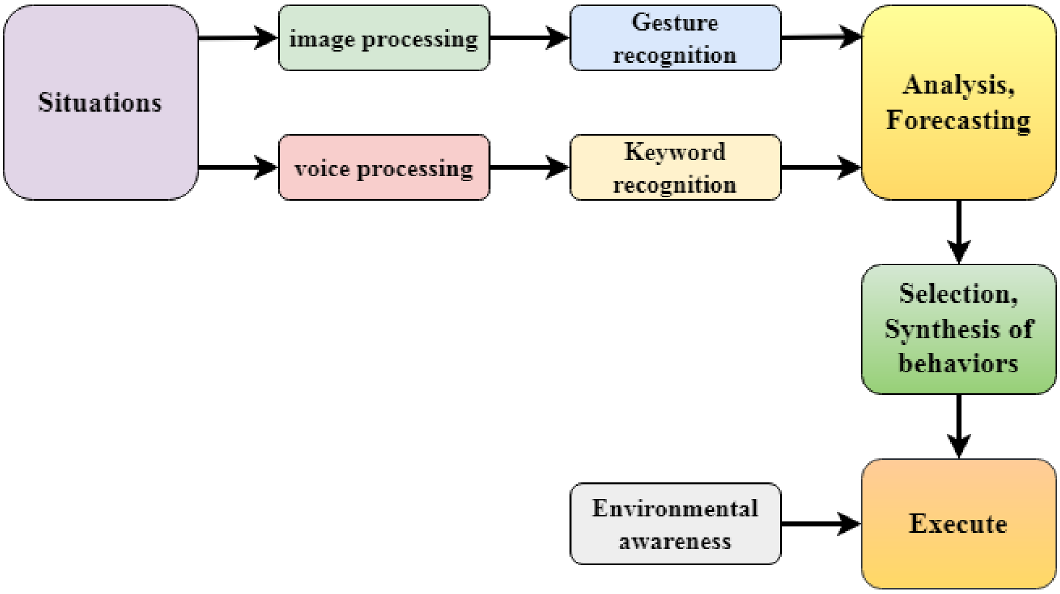



*Figure 9*. Generating sequence of behaviors for the robot.

Figure 9 depicts a procedure to select a robot's behavior sequence from its motion library in the database. The input to the system comes from the context in which the robot is experiencing. This context can be reading a piece of text or it can also be analyzed from the words or expressions of someone.

The camera systems will perform the task of acquiring images and recognizing gestures of people standing opposite to the robot (or can also identify the general situation of the whole space in front of the robot when there are many people). The voice processing block will take care of pre-analyzing the text for the auto-narrative situation or processing the sound source obtained from the microphone system to identify important keywords. These keywords can help the robot know what behavior it should take (for example, protest or emphasize...).

The information after being decomposed from voice (or words) and image processing will be pushed into the analysis and forecasting block. This is the central processing block, which will determine the direction of the robot's response. The output of this block is specific requests for possible behaviors. This information is provided to the behavior selection and synthesis block from the available motion library of the robot, from which it calculates and adjusts to build a sequence of behaviors consistent with the results of analysis and prediction previously reported. This sequence of behaviors may include a single movement, or may include multiple movements performed at the same time for the eyes, head, hands, and mobile base. The "analysis, forecasting" block and "selection, synthesis of behaviors" block are embedded in the "task planning" module. The end of the process is the execution of the robot with additional information from the sensor system that perceives the surrounding environment.

In essence, the problem of building a robot's behavior sequence is to equip the robot with the ability to react to a known or random situation or in other words, equip the robot with a personality when placing it in a human environment.

Note that the robot will have to interact mainly with elementary school children, so it is necessary to build a behavioral and expression data set suitable for this age.

In order to simplify the selection of suitable behaviors for specific scenarios, we group the movements that the robot can perform from the motion library into several categories:

- List of emotions (mainly relate to the expressions of the eyes following by a few motions of the hands and head):
  - Sadness
  - Anger
  - Love
  - Surprise
  - Fear
  - Joy
- List of sentence types (include both motions of body parts and the eyes):
  - Question
  - Statement
  - Command
  - Surprise
  - Protest
  - Emphasize
- List of gestures:
  - Hand movements: shaking, waving, thump up/down…
  - Head movements: shaking, tilting…
  - Base movements: spinning around, moving around with the same posture…

## 4. CONCLUSION

This paper has presented the structure of a humanoid intelligent robot both in terms of hardware and software architectures.

The humanoid robot has been designed in a friendly and lovable way to interact with young children in an elementary school environment. It has the ability to perform complicated human-like movements thanks to its flexible mechanical system.

The motion control software modules are developed mainly on the ROS framework, allowing the ability to integrate and manage quite flexibly all resources as well as the adjustment of function blocks. Here, we only use ROS framework as a mean to interchange the information between software modules in the robotic control systems. For important and complicated processes, event-based and parallel threads is our solution to control the behavior of the robotic system.

In order to be able to interact with humans, the robot system is equipped with intelligent processing modules (including face, voice and gesture recognition), allowing it to analyze, predict and perform friendly actions appropriate to the contexts.

The Bonbon robot can be considered one of the first intelligent humanoid robots in Viet Nam. Although it can do many tasks, it still has issues that need to be improve in further studies:

- *On the hardware design*: The selection of actuators plays a very important role for the robot motion control problem. It is not easy to find good servo motors that can be used for the robottic arms due to the limitation in small size while requiring powerful torque. Most of the available ones only allow position control modes which limit the control reaction speed of the arms. Besides, the optimal compromise between the robot high computation and control performance and the power consumption is also a difficult issue to be solved.
- *On the software control system*: ROS framework mainly supports us in managing the interchange of information between peripheral devices and various control modules flexibly. The stability of the control system is still greatly depended on the general software structure (which is the 3L structure in this case) and our own event-based solutions for dealing with complicated control processes. On the one hand, ROS community (or ROS2, recently) also provides advanced computation packages such as MoveIt (for control the robotics arms) or Navigation (for control the mobile base). However, most of these packages are useful for more general control problems. On the other hand, for specific control problems such as controlling the humanoid robot in our case, we need to develop our own controllers to strictly manage the behavior of each robot part in realtime following our own rules. These controllers are developed based on the mathematical models of the robot parts (upper body and mobile base). Using too many ROS-based tools is not always a safe solution because they have risks that we cannot control. Therefore, the most effective solution is probably to take advantage of only the best features of ROS to bring harmonious integration into our own robot control solutions.

The Bonbon robot has been being tested. Some function blocks are still being developed with better solutions to control problems for higher efficiency.

***Acknowledgements.*** The research leading to these results has received funding from the Vietnam national project No. KC-4.0-04/19-25.

***CRediT authorship contribution statement.*** Nguyen Dinh Quan: Methodology, Control system design and implementation. Ha Huy Hung: Mechanical design. Le Dinh Son: Supervision, Funding acquisition. Tran Van An: Develop system management and task planning layers. Nguyen The Hung: Social behavior analysis. Cao Viet Linh: Control software development.

***Declaration of competing interest.*** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## REFERENCES

1. Arkin R. - Behavior-Based Robotics, MIT Press, USA, 1998.
2. Firby R. J. - Adaptive Execution in Complex Dynamic Worlds, Ph.D. Thesis, Yale Univ., New Haven, 1989.
3. Firby R. J. - Task networks for controlling continuous processes, Proc. 2nd Int. Conf. AI Plan. Syst., 1994.
4. Bonasso R. P. - Integrating reaction plans and layered competences through synchronous control, Proc. Int. Joint Conf. Artif. Intel., 1991.
5. Rosenschein S. J., Kaelbling L. P. - The synthesis of digital machines with provable epistemic properties, Proc. Conf. Theor. Asp. Reas. Knowl., 1998.
6. Kaelbling L. P. - Goals as parallel program specifications, Proc. 6th Natl. Conf. Artif. Intel., 1988.
7. Kaelbling L. P. - Compiling operator descriptions into reactive strategies using goal regression, Tech. Rep., TR90-10, Teleos Res., Palo Alto, 1990.
8. Bonasso R. P., Firby R. J., Gat E., Kortenkamp D., Miller D. P., Slack M.G. - Experiences with an architecture for intelligent, reactive agents, J. Exp.Theor. Artif. Intell. **9**(2/3), 237–256, 1997.
9. Gat E. - Integrating Planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots, Proc. Natl. Conf. Artif. Intel. (AAAI), 1992.
10. Saridis G.N., Architectures for intelligent controls, Intelligent Control Systems: Theory and Applications, ed. by S. Gupta , IEEE Press, Piscataway, 1995.
11. Alami R., Chatila R., Fleury S., Ghallab M., Ingrand F. - An architecture for autonomy, Int.J. Robot. Res. **17**(4) 315-337, 1998.
12. Stanford Artificial Intelligence Laboratory et al. - Robotic Operating System, 2018. Available at: https://www.ros.org.
13. Wisama K., Jean-Francois K. - Minimum operations and minimum parameters of the dynamic model of tree structure robots, IEEE Journal of Robotics and Automation, **RA-3**(6) (1987). https://doi.org/10.1109/JRA.1987.1087145.
14. Hirose M., Ogawa K. – Honda humanoid robots development, Philos. Trans. Ser. A Math. Phys. Eng. Sci. **365**(1850), 2007.
15. Sakagami Y., Watanabe R., Aoyama C., Matsunaga S., Higaki N., Fujimura K. - The intelligent ASIMO: system overview and integration, IEEE/RSJ International Conference on Intelligent Robots and System. IEEE **3** (2002) 2478-2483.
16. Park I. W., Kim J. Y., Lee J., Oh J. H. – Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot - 3: HUBO), $5^{th}$ IEEE-RAS International Conference on Humanoid Robots (2005) 321-326.
17. Twan K., Sylvain B., Gray T., Tomas D. B., Tingfan W., Jesper S., Johannes E. and Jerry P. - Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas, International Journal of Humanoid Robotics **13** (01) (2016). https://doi.org/10.1142/S0219843616500079.

18. Leobardo C., Oscar E. C. E., Alexander G. L., Eduardo J. B. C. - Stabilization method for dynamic gait in bipedal walking robots, IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), 2016.
19. Softbank robotics – pepper robot, 2018. https://www.ald.softbankrobotics.com/en/cool-robots/pepper.
20. Tanaka F., Isshiki K., Takahashi F., Uekusa M., Sei R., Hayashi K. - Pepper learns together with children: development of an educational application, 2015IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids) 2015, pp. 270-275.
21. Perera V., Pereira T., Connell J., Veloso M. M. - Setting up pepper for autonomous navigation and personalized interaction with users, CoRR abs/1704.04797, 2017.
22. Softbank robotics – naoqi framework, 2018. http://doc.aldebaran.com/2-5/ref/index.html.