COMPLETE
Journal of Computer, Electronic, and Telecommunication
ISSN 2723-4371, E-ISSN 2723-5912
doi.org/10.52435/complete.v2i1.203

Parameter Adjustment of EROS Humanoid Robot Soccer using a Motion Visualization

Anhar Risnumawan[1,*], Rokhmat Febrianto[2], Indra Adji Sulistijono[3], and Eny Kusumawati[4]

[1,2,3,4]Mechatronic Engineering Division, Politeknik Elektronika Negeri Surabaya (PENS)
[1]anhar@pens.ac.id, [2]rokhmat@me.student.pens.ac.id, [3]indra@pens.ac.id, and [4]eny-k@pens.ac.id,

Abstract: Humanoid robot is a robot whose overall appearance is formed based on the human body and can interact with equipment and the environment created by humans. The robot's balance becomes fundamental in carrying out various tasks in designing humanoid robots. To deal with this, the adjustment of the humanoid robot movement is crucial in this work, research related to the virtual visualization of robots. Virtual robot visualization can be done by creating a simulator that contains dynamic parameters, including the physics of the robot. With the simulation containing dynamic parameters, the humanoid robot movement can be tried many times until the robot movement is robust. Applying the URDF (Unified Robot Description Format) model to the Gazebo simulator, which is supported by the ROS (Robot Operating System) framework, can make a simulator with dynamic parameters mimicking a real environment. In order to make a robust robot motion, feedback is needed in position and torque to find out the difference between simulation and reality. On the other hand, simulations can be done without cost or risk and, most importantly, mimic the actual robot soccer environment.

Keywords: Humanoid Soccer Robot; Motion Visualization; ROS; URDF; Gazebo Simulator

---

1. Introduction

Today, the application of humanoid robots in daily activities is no longer considered unusual and has been frequently used by researchers and users to complete a given task. The development of humanoid robots has attracted much interest for researchers in robotics. These can be observed by educational institutions and technology companies that have competed to show the results of research on humanoid robots such as ASIMO robots by Honda [1], Valkyrie robots by NASA [2], T-HR3 robots by Toyota [3], DLR-TORO robots [4], and the NimbRo-OP2X robot [5].

One of the organizations highly engaged in humanoid robots, Robocup, has predicted and aims to develop a humanoid robot team that can win against the world championship team in 2050 with the rules of FIFA. In Indonesia itself, the trend of humanoid robots is still developing. The Indonesian Higher Education deeply supports the Kontes Robot Indonesia (KRI) held annually. Humanoid robots are in the division of the Kontes Robot Sepak Bola Indonesia (KRSBI), which tests the team robot performance from several universities.

Each year the robot soccer game rules are continuously revised to mimic the actual FIFA soccer rules. The new rule of a soccer field that previously used carpet changes into synthetic grass, affecting robot stability and balance significantly. Types of synthetic grass are bumpy and not always flat, making the robot's movements weave or unstable, which can cause the robot to lose balance and fall quickly. Teams often employ enormous hand-tuning of robot parameters to overcome these problems before the match starts. Correct parameter adjustment of the humanoid robot movement is essential so that the robot can at least perform fundamental soccer skills, such as walking, kicking, and robustly. However, this method is less effective in practice because it requires much time to try the adjustments' results while potentially breaking the robot.

Designing a new humanoid robot would not be trivial without a sophisticated simulation (visualization), especially when designing a walking gait. A sophisticated visualization such as designing a new motion, new kicking motion, and get-up motion from falling, and testing those motions in a 3d simulator that considers the physics of a real environment, such as gravity and mass, are crucial for humanoid robots soccer. Previously, much trial and error had been performed until a more reasonable performance was achieved. Although the trial-and-error process is common and simple in practice, the age of the robot motor servos can degrade significantly. With the help of visualization in a 3d simulation, a nearly actual robot model, through a commonly URDF (Unified Robot Description Format) model, can be deployed in the simulation, and all the required parameters can be adjusted before actual deployment and without degrading robot motor servos.

In connection with these problems, it is necessary to make a virtual visualization in simulation [6], which contains dynamic parameters. With the simulation that contains dynamic parameters, the Humanoid robot movement can be tried many times until the robot movement is robust or follows what we want. However, the simulator still has some challenges, namely comparing simulation results and reality. By applying the URDF Model to the Gazebo simulator and supported by the ROS (Robot Operating System) framework, a simulator can be made with dynamic parameters. In order to make a robust robot motion, feedback is needed in position and torque to find out the difference between simulation and reality. On the other hand, simulations can be done without cost, risk, and, most importantly, fast. An example of a virtual humanoid robot visualization, along with its kinematic diagram, is shown in **Figure 1**.

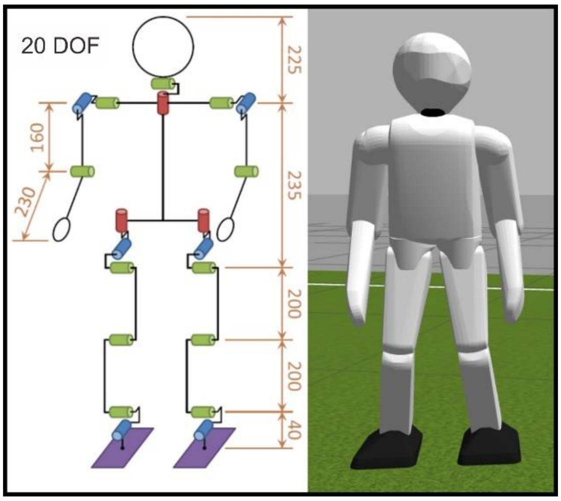

Figure 1. Kinematic diagrams and virtual visualizations of EROS humanoid robots sourced from NimbRo.

2. Related Work

The humanoid robot is a branch of a robot with a body shape built to resemble a human body. In general, humanoid robots have a torso, head, two arms, and two legs, although some humanoid robots may only model parts of the body, for example, from the waist up. One of the advantages of creating humanoid robots is that they can interact with the human environment. Some assignments can be completed with wheeled-type mobile robots by giving a little touch of human behaviour, such as the robot arm. However, to interact with humans and be confused with many obstacles, humanoid robots have a distinct advantage in terms of creation. Below are examples of humanoid robots that have been developed in various technology industries and educational institutions.

One example of humanoid robots that are often used is DARWIN-OP. DARWIN-OP (Dynamic Anthropomorphic Robot with Intelligence - Open Platform) is an open-sourced miniature humanoid robot platform with a sophisticated high computational power, sensors, and dynamic motion capabilities developed and produced by robotics manufacturers Robotis in collaboration with Virginia Tech, Perdue University, and the University of Pennsylvania. This robot has 20 Degree of

Freedom (DoF), which is thought to be enough to do mobility in playing football. The primary purpose of DARWIN-OP is for research and programming in the scope of humanoid robots, artificial intelligence, vision, inverse kinematics, walking algorithms, and linguistics. Recently, DARWIN-OP has developed a third robot called ROBOTIS OP3. A striking difference from the previous series is to replace the actuator that previously used Dynamixel MX-28 with Dynamixel XM-430.

Other related work comes from the Nimbro team "NimbRo-OP2: Grown-up 3D Printed Open Humanoid Platform for Research," on the 17th IEEE-RAS Int. Conference on Humanoid robots (Humanoids) [7]. This research aims to introduce the NimbRO robot to the public. He stands 135 cm tall and weighs up to 18 kg, which aims like a predecessor robot that can interact with the surrounding environment. This robot is also equipped with features that can adjust the robot's movements as desired by simulating the ROS. The advantage of this robot compared to the Igus Humanoid-OP robot is a better robot leg structure. The robot may cause severe damage if it falls due to its height and weight. Thus, it is crucial to perform tuning and testing while in simulation at first before testing on the real robot.

3. Materials and Methods

3.1. System Overview

EROS humanoid robot is designed to play soccer. This EROS robot has a height of 90cm and a total weight of around 7 kg. It uses power from a 4-cell Lithium Polymer (LiPo) battery of 2 pieces to cover the electrical parts of this humanoid robot. A monocular camera used has 720p HD specifications using 110° wide angles. The camera is located in one of the humanoid robot's eyes. The robot main hardware is NUC Core i7 as a primary processor and uses 4GB DDR3 memory and SSD as storage. The system runs on the Indigo and Ubuntu 14.04 ROS (Robot Operating System) platform.

Designing the circuitry is inseparable from the electronic components used. The overall block diagram can be seen in **Figure 2**. Electronic components are needed to power the robot, starting from making power boards, controller circuits, and wiring schemes. Electronic components are also useful for accessing data from sensors and igniting actuators. A block diagram of the electrical subsystem of the robot design is shown in **Figure 2**. At the heart of the electrical design is the Robotis OpenCR 1.0 sub-controller board, which is equipped with U2D2 as a low-level control to access sensors and servo motors.

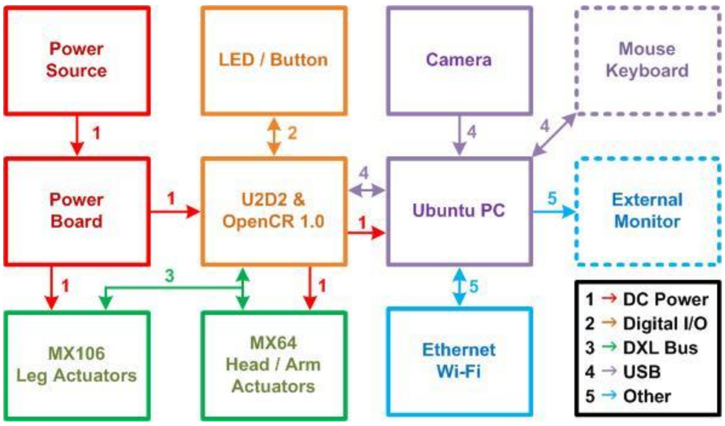

Figure 2. An overall diagram of the system block of electronic components and their connections to the EROS humanoid robot.

Robotis OpenCR 1.0, as the core of a sub-controller, can get real-time data with feedback input as much as 20 degrees of freedom in humanoid robots. Compared to the Robotis CM740 used by the NimbRO Platform, Robotis OpenCR 1.0 performance is improved because of the difference in chips between the two modules made by Robotis. The Robotis OpenCR 1.0 allows the EROS platform to implement closed-loop control faster, which, previously on the NimbRO platform, employs the older Robotis sub-controller CM740.

3.2. Movement Visualization

The overall system design is explained in this section. Making a simulator that can visualize humanoid robot movements requires motion control, which will process the URDF Model. The trajectory editor will create a pose called motion files, and a set of motion files will create movements called motion modules. URDF models that already contain information from motion modules will produce dynamic motion in the robot control block [8]. The motion control system design can be seen in **Figure 3**. Hence dynamic motion will enter the hardware interface block, which can later be realized directly on humanoid robots through Robotis OpenCR 1.0 & U2D2 or into the Gazebo simulator to be visualized virtually for simulations.

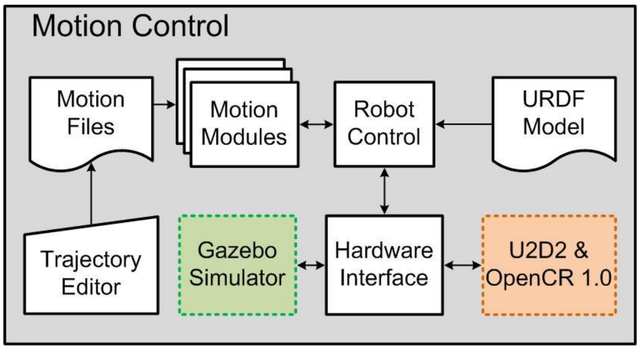

Figure 3. The architecture of motion control in the EROS humanoid robot software.

The chart of motion control will be embedded in the ROS framework to manage the overall work of this humanoid robot platform. Motion control certainly gets help from several other components as an input and a place to distribute the output that has been processed therein. In general, these components can be divided into three parts—first, the URDF model as a virtual humanoid robot visualization. Second, gait generation is a process of making various humanoid robot movements, supported by the GUI (Graphical User Interface) to simplify the adjustment process. The third is the data collection design as the process of exiting and entering all data from motion control to the actuator or sensor.

3.2.1. URDF Model

An essential part of the simulation is forming a virtual humanoid robot model with several parameters owned by real humanoid robots. Examples of these parameters are the inertia and mechanical parts of the robot. It is necessary to change to URDF format so that it can be read on ROS. The URDF package contains several XML specifications, for example, building robot models, sensors, and scenes. Each XML specification has an appropriate parser in one or more languages.

Each URDF package has specific things for each robot model. The specifics are the joints or joint points of two or more parts. Joints owned by humanoid robots, in general, are like joints owned by humans. The URDF model is built by paying attention to all components related to the robot model that will be built in it because the URDF model represents the original robot model in the virtual world that will be built for future development.

In changing to a URDF package, it is necessary to divide it into several parts for its definition. Each part must be defined as position, rotation, and other information. It is intended that each section can have information such that a real humanoid robot owns it. With this, a program that includes all parts that already have that information can be made into one EROS humanoid robot that can be read on ROS [9], [10].

3.2.2. Gait Generation

In making humanoid robot movements, several formulas must be considered, especially the kinematic part of the humanoid robot. On this platform, the EROS humanoid robot uses the RBDL library, which helps form movements. RBDL contains a very efficient code for forwarding and reversing dynamics for kinematic chains and branching models.

Also, it contains code for forwarding and reversing kinematics and closed-loop models. Models can be loaded from Lua scripts or URDF files [11]. Using RBDL, matters relating to forward and reverse dynamics will be resolved quickly. The movement configuration used by EROS must be broadly modified to match the same walking algorithm as the TeenSize robot initially by NimbRo. The chosen parameters are given in Table 1.

Table 1. Gait configuration parameters that we selected.

Variable	Value	Index
C1	0.02	Halt Position Leg Extension
C2	0.1	Halt Position Leg Roll Angle
C3	0.02	Halt Position Leg Pitch Angle
C4	0.03	Halt Position Foot Roll Angle
C5	0	Halt Position Foot Pitch Angle
C6	0.02	Constant Ground Push
C7	0	Proportional Ground Push
C8	0.3	Constant Step Height
C9	0.12	Proportional Step Height
$C\tau0$	0	Swing Start Timing
$C\tau1$	2.3876	Swing Stop Timing
C10	0.17	Sagittal Swing Amplitude Fwd
C11	0.12	Sagittal Swing Amplitude Bwd
C12	0.1	Lateral Swing Amplitude
C13	0.05	Lateral Swing Amplitude Offset
C14	0.015	Turning Lateral Swing Amplitude Offset
C15	0.2	Rotational Swing Amplitude
C16	0.05	Rotational Swing Amplitude Offset
C17	0.035	Lateral Hip Swing Amplitude
C18	0	Forward Lean
C19	0	Backward Lean
C20	-0.07	Forward and Turning Lean
C21	3.5	Gait Velocity Limiting Norm p
C22	0.0085	Sagittal Acceleration
C23	0.01	Lateral Acceleration
C24	0.009	Rotational Acceleration
C25	0.09	Constant Step Frequency
C26	0.008	Sagittal Proportional Step Frequency
C27	0	Lateral Proportional Step Frequency

The process of adjusting humanoid robot movements through several stages can generally be represented in **Figure 4**. The initial stage is to design a new motion from the GUI and performs motion parameters adjustment according to what we want. The results of these movements can be tried on a gazebo simulator located on ROS. This process will be repeated until the movement is appropriate and robust. This can be done by setting the time and position of each movement of the GUI [12].

Figure 4. The appearance of the implementation of the whole gait generation. Starting from the rightmost picture that is making robot movements from the GUI (Graphical User Interface) that has been provided, switch to the image in the middle to simulate it on the ROS framework. The last step is the leftmost picture to practice directly on humanoid robots.

The next step is to practice the movement with real humanoid robots. This stage will get a big challenge because some of the movements will be slightly different from the simulations that have been done. Several mechanical factors affect humanoid robots, but the effect is not too significant because the help of GUI and simulation can reduce the length of the adjustment process using a real humanoid robot. The duration of this whole process depends on the user experience to adjust the motion of the humanoid robot.

3.2.3. Data Collection Design

The data needed in this system is the moment of inertia, weight, and origin of each joint in the humanoid robot; the data will be obtained from the URDF Model. The URDF Model gets all the data from the CAD humanoid robot design that has been made by estimating the moment inertia value and weight from the CAD humanoid robot design. Determining the area of origin of each joint refers to the CAD design humanoid robot in a default state that is standing.

The following data needed is feedback from the servo to find out the real situation in the robot [13]. The feedback data from the servo will be compared with the URDF Model to determine the error between the virtual visualization and the actual situation. The following is a block diagram of how hardware works in **Figure 5**.

Block diagram in **Figure 5** shows the workflow of communication to get feedback carried out between Servo – Control board – Hardware Interface (on Mini PC). Feedback from the servo can be used as input for the Motion Control system, which will influence the determination of Dynamic Motion.

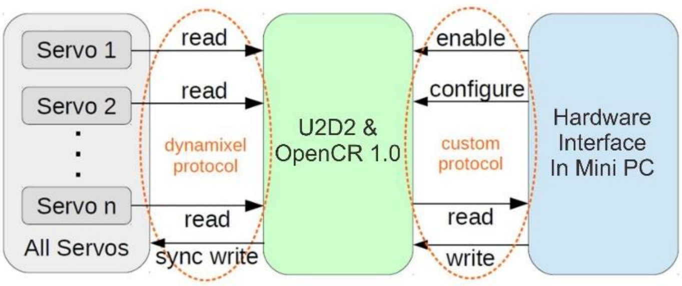

Figure 5. Block diagram of how data flows.

5. Results

Experiments were carried out using robots with the specifications described in the system review. Following the movement adjustment process described in the gait generation will move. To evaluate the performance of these movements needed, data can be a reference for the success of humanoid robots working well. The position data of humanoid robot parts are taken after the adjustment process, as in **Figure 6**, before being given feedback from the servo compared with position data of humanoid robot parts that have been given feedback from the servo.

For example, the position data of the humanoid robot section taken is at the ankle pitch. Experiments were carried out by making humanoid robots move forward and then taking 10 seconds to compare material between movements that are not given feedback from the servo (open-loop control) with movements that are given feedback from the servo (closed-loop control). The whole process of motion adjustment refers to the gait generation, which is generally represented in **Figure 6**.

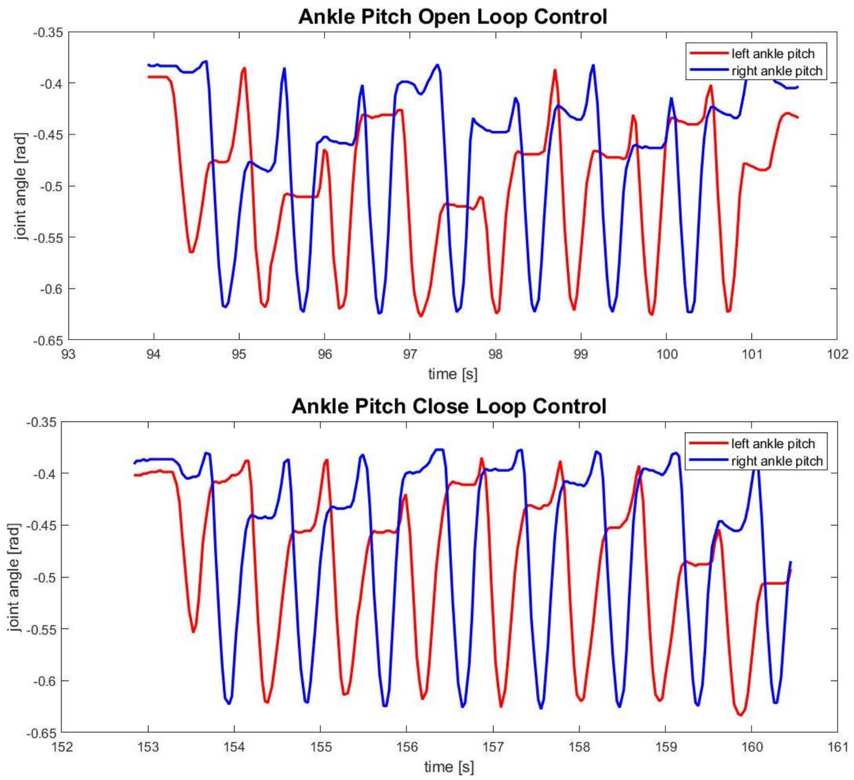

Figure 6. Comparison of the position chart of the humanoid robot's ankle pitch between movements that are not given feedback (open-loop control) to movements that are given feedback (close loop control). This shows that using a close loop can improve the stability of the motion of humanoid robots.

The graph provided by the motion given feedback from the servo (closed-loop control) is better than the movement that is not given feedback from the servo (open-loop control). This can be seen in **Figure 6**. The closed-loop position data from each part of the ankle pitch gives a steady trend.

Following the example, the position data of the humanoid robot section is at the knee pitch. Experiments were carried out by making humanoid robots move forward and then taking 10 seconds to compare material between movements that are not given feedback from the servo (open-loop control) with movements that are given feedback from the servo (closed-loop control). The whole process of motion adjustment refers to the gait generation, which is generally represented in **Figure 7**.

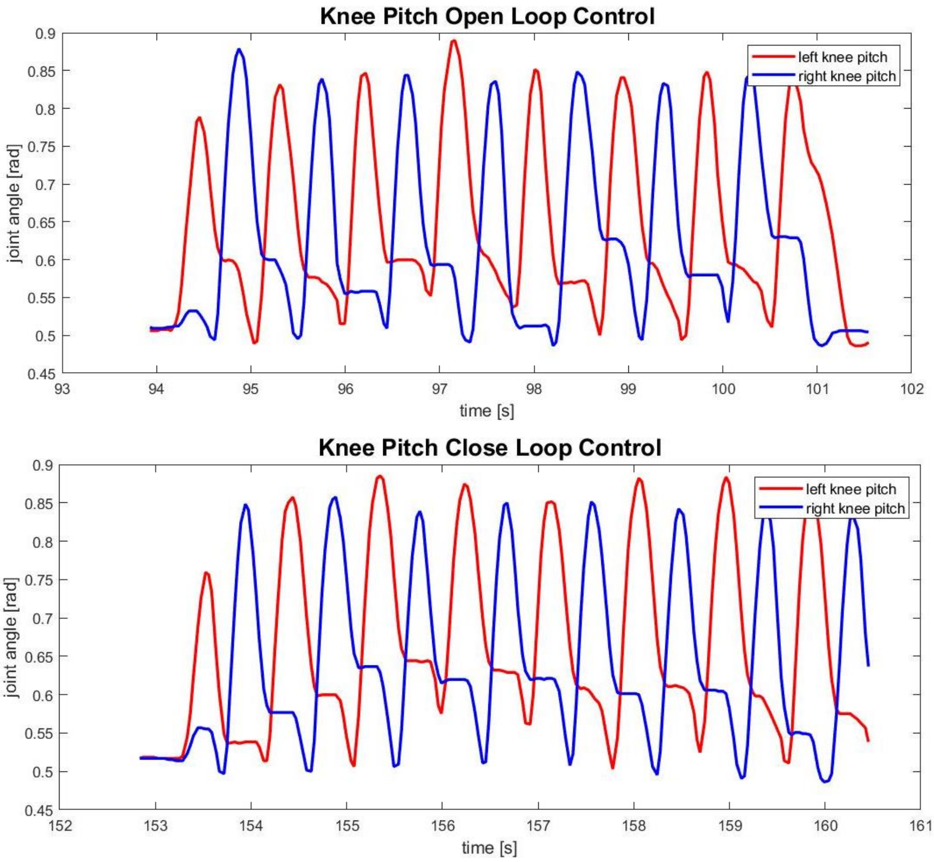

Figure 7. Comparison of the position chart of the humanoid robot's ankle pitch between movements that are not given feedback (open-loop control) to movements that are given feedback (close loop control). This shows that using a close loop can improve the stability of the motion of humanoid robots.

The data flow from all servos can be shown in **Figure 5**. The mini PC connects directly to the servos using U2D2 hardware to increase the data streams, providing real-time commands and position feedback from all servos. Sensor IMU (Inertial Measurement Unit) provides the robot's acceleration and orientation, and buttons are provided by the OpenCR board using the same bus of U2D2.

6. Conclusion

In this paper, automatic motion visualization based on ROS has been developed on the EROS humanoid robot platform to accelerate the process of adjusting humanoid robot movements when changing terrain, as stated in the RoboCup rules, also minimizing damage to humanoid robots when adjusting movements. Combining URDF models and gazebo simulators on ROS helps adjust the movement virtually. The existing GUI on the ROS makes it easy to change the position of each servo on humanoid robots and speed up knowing whether or not the movement. Experiments show satisfactory results for adjusting humanoid robot movements can be done virtually. The movement that is set virtually results in movements that tend to be stable in real humanoid robots; this has been proven in experiments that have been carried out. Our work has been successfully implemented in practice and achieved remarkable results in the actual application of humanoid robots.

7. References

[1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, 'The intelligent ASIMO: System overview and integration', in *IEEE/RSJ international conference on intelligent robots and systems*, 2002, vol. 3, pp. 2478–2483.

[2] N. A. Radford *et al.*, 'Valkyrie: Nasa's first bipedal humanoid robot', *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[3] T. G. Newsroom, 'Toyota Unveils Third Generation Humanoid Robot T-HR3.(21 November 2017)', *Retrieved March*, vol. 21, p. 2018, 2017.

[4] J. Englsberger *et al.*, 'Overview of the torque-controlled humanoid robot TORO', in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 916–923.

[5] G. Ficht *et al.*, 'NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot', in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–9.

[6] A. Farchy, S. Barrett, P. MacAlpine, and P. Stone, 'Humanoid robots learning to walk faster: From the real world to simulation and back', in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 39–46.

[7] G. Ficht, P. Allgeuer, H. Farazi, and S. Behnke, 'NimbRo-OP2: Grown-up 3D printed open humanoid platform for research', in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 669–675.

[8] P. Allgeuer, H. Farazi, M. Schreiber, and S. Behnke, 'Child-sized 3D printed igus humanoid open platform', in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 33–40.

[9] R. W. Contributors, 'ROS/Introduction-ROS Wiki'. Aug, 2022.

[10] R. O. S. Wiki, 'URDF', *URL: http://wiki. ros. org/urdf.(Accessed 10 April 2022)*, 2022.

[11] 'Rigid Body Dynamics Library: Mainpage'. https://rbdl.github.io/ (accessed Mar. 23, 2022).

[12] M. Missura and S. Behnke, 'Self-stable omnidirectional walking with compliant joints', 2013.

[13] H. Farazi *et al.*, 'RoboCup 2016 Humanoid TeenSize Winner NimbRo: Robust Visual Perception and Soccer Behaviors', *arXiv preprint arXiv:1809.11127*, 2018.