Realization of Arms Movements for 3D Robot Model and Synchronization with Real Humanoid Robot

E. Kendereši, M. Raković, M. Nikolić, M. Gnjatović, B. Borovac
University of Novi Sad/Faculty of Technical Sciences, Novi Sad, Serbia
kenderesi.endre@gmail.com; rakovicm@uns.ac.rs;
milutinn@uns.ac.rs; gnjatovic@uns.ac.rs; borovac@uns.ac.rs;

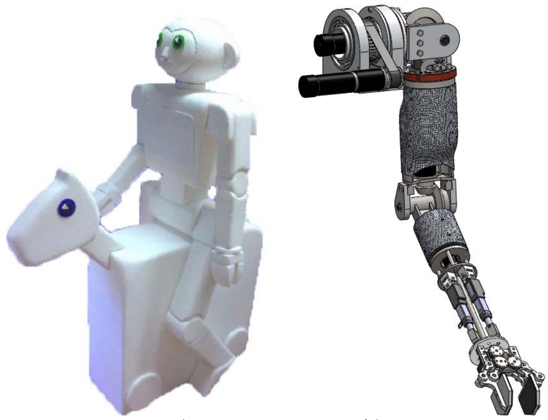
***Abstract*—The paper describes the simulation software for preview of the motion of humanoid robot: either graphical in 3D environment or on the real robot (KHR-1 HV was used). Such software enable synthesis (both, online and offline) and execution of humanoid movements. Execution can be performed either on the 3D model or on the real robot. Such software brings two major advantages. First, to avoid self collision and to verify movement shape all synthesized movements can be previewed before executing on the real robot. Second, such system can be used for early stages of testing different sub-systems, for example, visual system, cognitive algorithms etc.**

I. INTRODUCTION

The development of humanoid robot as assistive technology to help the children with developmental disorders is complex and challenging task. It requires involvement of researchers from different engineering fields. Because of the interaction of the robots with children it is also necessary that artists (for design of robot appearance) and the medical researchers participate in the project realization. Work on the ongoing project: “Design of robot as assistive technology in treatment of children with developmental disorders” (funded by the Ministry of science and education, RS) is aimed to result in such a robot.

Robot involved in the therapy with children should be appealing and has to ensure verbal and non-verbal communication with them. The overall robot appearance is very important. It is expected that the children will easier accept anthropomorphic robot (the robot having head, arms and legs). Also, during the treatment, robot should possess the information about the position of the child, the environment and the objects used during the therapy. This means that robot should also possess vision, speech recognition and synthesis abilities and cognitive system.

It is already mentioned that the overall appearance of robot is very important. This implies the constant coordination in the work of the artists and the designers of mechanical system. In Fig. 1. is presented the artistic view of robot appearance and the last version of mechanical design of the arm [1]. After completing the design, procurement of all the necessary components, production of parts and assembly of robot should follow. This process is time-consuming and can significantly slow-down the development of other subsystems of the robot.



a) b)

Figure 1. Robot assistant for children's therapy; a) appearance of the proposed robot, b) mechanical design of the robot’s arm

To achieve better efficiency and ensure early testing and implementation of other subsystems, the software for simulation of movement is of high importance. The development of simulation software which is used to support the development of real robot is common practice [2-4].

In this paper is presented the first version of simulator for the movement execution of the robot. The basic functions simulator provide are synthesis (generation) and realization of complex arms and head movements as well as the graphical representation of the robot model in 3D environment. Because the KHR-1 HV robot was available

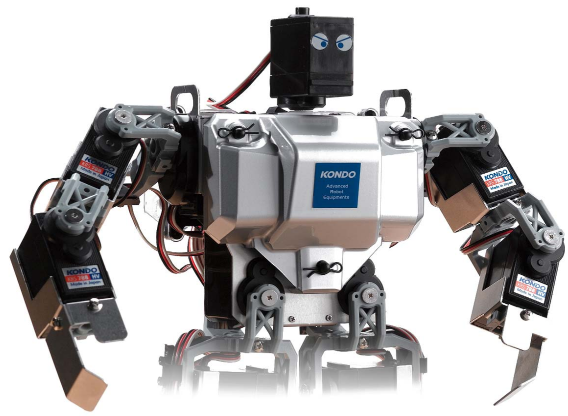


Figure 2. Robot KHR-1HV

(Fig. 2), the simulation software is realized for this robot. By replacing the original robot controller with custom designed the realization of synthesized (and simulated) movements on real robot is enabled.

II. Model of Robot and View in 3D Environment

Since current version of the simulator is dedicated to KHR-1 HV, first its kinematics structure will be explained, and how the joints motions are specified.

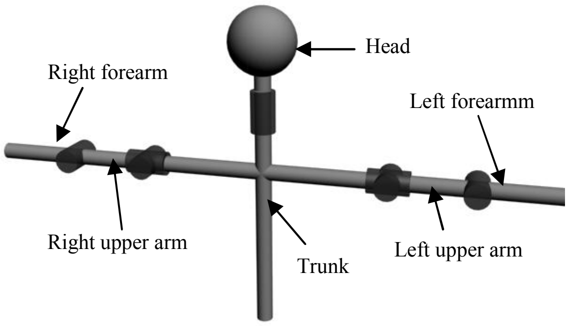


Figure 3. Kinematic structure of robot KHR-1 HV

In Fig. 3. is shown the kinematic scheme of the robot, consisting of 6 segments connected with 7 single degree-of-freedom (DoF) joints. Each shoulder has 2 DoFs (4 in total for shoulders) and one DoF each elbow. Seventh DoF belongs to joint between the trunk and the head. Each DoF is driven with mini RC servo. Motion control of each joint is performed by providing one PWM signal with appropriate frequency and duty cycle [5].

For the graphic representation in 3D environment, all segments (trunk, upper arm, forearm and head) are drawn in CAD software. The kinematic model of robot is realized using the Object-Oriented Graphics Rendering Engine (OGRE) [6]. OGRE is the set of libraries which provide a high level interface based on the world objects (like segments of robots) and it deals with underlying system graphic libraries like Direct3D and OpenGL.

In Fig. 4 is shown the 3D model representing the good copy of upper part of KHR-1 HV robot. The user can move each joint by using predefined keyboard shortcuts or by dedicated window for fine jogging of robot joints.

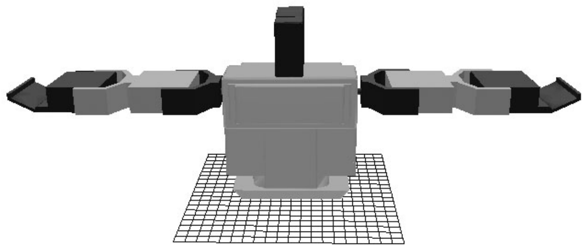


Figure 4. Rendering of robot model using OGRE

III. Hands Movements Synthesis

Complex movements can be defined (synthesized) using one or more simple movements (primitives). Simple movements are defined by starting and final robot posture and movement velocity. To define simple movement, the user should set starting and final posture and motion velocity. Movement can be made more complex by adding new simple movements, i.e. by combining simple movements the user can compose complex movement. For example, clapping the hands can be composed of two simple movements: spreading and closing the arms (ensuring hands perform clapping).

A. *Generation of complex movements*

The described procedure for generating complex movements by combining the simple movements is not new. In [7, 8] is described how the walk for biped robot can be synthesized from simple movements.

To generate movements, the user should define and store all robot postures used in simple movements. Thus, relational database consisting of seven tables has been designed. It should be noted that if it is required to add new primitive, its starting posture should coincide to last (final) posture of the already defined movement. Since the simple movements are derived from initial and final posture of the robot, complex movements are formed by successive execution of the simple ones. Relation between tables corresponds to relations between postures, simple and complex movements. The tables (with attributes) and relations are shown in Fig. 5.

In table ***Movement type*** are stored data used to describe the movement type. Absolute and relative movement types can be defined. For relative movements, the final

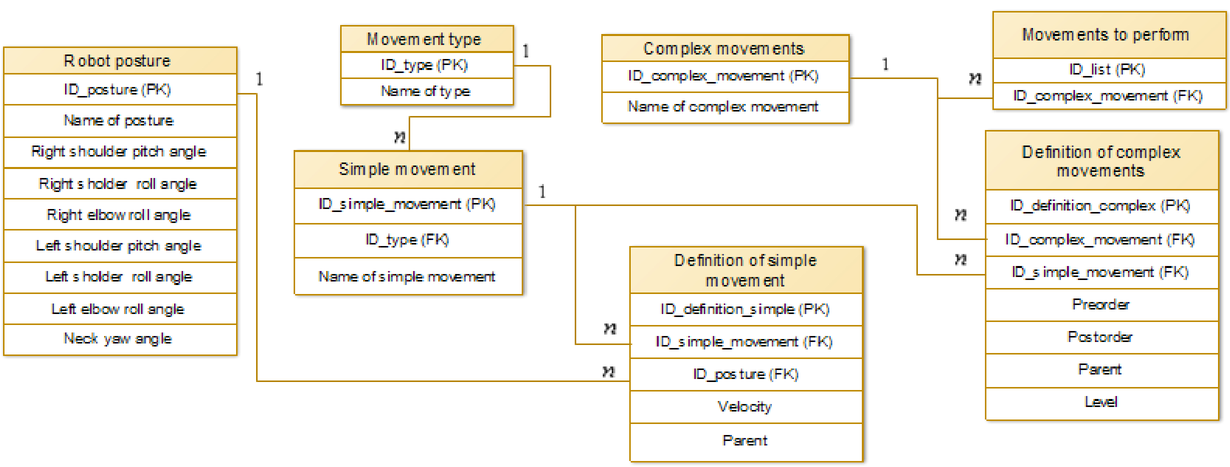


Figure 5. Preview of data table relations

robot posture is obtained by increasing (or decreasing) current angular values at joints in such a way to preserve actual shape of the movement (for example, raise hand higher). For absolute movements, joint angles are such defined to ensure final desired hand posture. Absolute movement can be executed from various initial postures, final posture is always same. Example of command where absolute movement has to be executed is “put hand on your head”. The user also can introduce new movement type, for example, movement delay, to keep the joints still for some time.

Table ***Robot posture*** stores values angles of all robot joints for every single posture. The definition of simple, and indirectly of complex, movements is based on these table entries.

Simple_movement table represents the list of existing simple movements. When the user inserts new record, the name and type of simple movement must be set.

Definition of simple movements stores the robot postures, movement velocity and order of postures for each simple movement. Attribute *ID_simple_movement* means that this record denote particular simple movement, attribute *ID_posture* define postures used to compose the movement. Attributes *Parent* and *Velocity* are used to determine the order and velocity of postures to be attained during the movement.

Tables ***Complex movements*** and ***Definition of complex movements*** are used to store the data about existing complex movements and about composition of each complex movement. Table ***Complex movements*** stores the list of previously defined movements, while table ***Definition of complex movements*** stores the all the simple movements used to compose complex movements. The attribute *ID_complex_movement* determines that the record belong to particular complex movement. Structure of complex movement is represented as binary tree. This enables two or more simple movements to be executed in parallel. For example, if it is requested from the robot to raise its arms, the movement can be performed in several ways. Robot can raise both arms at same time, or it can raise the one arm at a time. If robot should perform movements in parallel, then the simple moments should be at the same level in binary tree. And robot should perform one movement at a time, they should be organized as parent movement and child movement.

Table ***Movements to perform*** stores the list of complex movements which will be executed on robot. Complex movements are preformed in order determined by *ID_list* attribute. The movement with smallest *ID_list* value is preformed first. When the movement is finished, the record is deleted from table, and next movement is selected.

B. Description of software

Software for simulation and robot control is divided in following parts:

- *Robot 3D model view* – graphic representation of robot model in 3D environment (Fig. 6.).
- *Jogging the robot joints* – fine tuning of each joint.
- *Postures and movements* – recording and handling of posture, composition of new simple and complex movements, and
- *Settings* – program set and establishes communication with real robot.

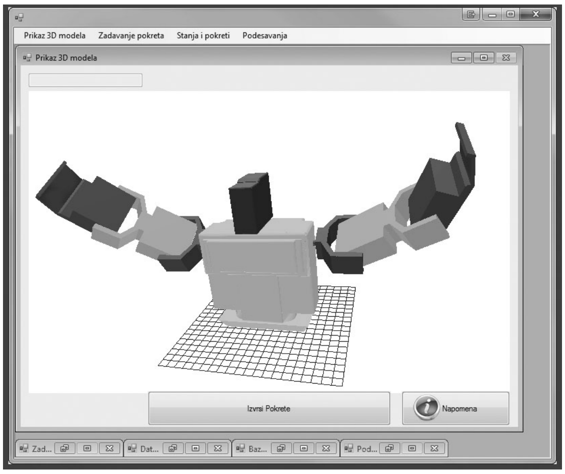


Figure 6. Robot model view window

Robot 3D model view handles the view of 3D humanoid robot model and visualizes the performing movement. The user can change zoom and camera position.

Jogging the robot joints enables each joints to be fine tuned (resolution is 0.01^o). *Postures and movements* are commands to record current model posture and to compose the movements. Also, this program deals with all other tables and their functionalities (definition of movement types, definition and modification of simple and complex movement, modification of recorded postures etc.).

Settings allow the user to set the limits for each joint to match the real robot. This part of program also handles the communication with real robot.

IV. Connection of Simulator and Real Robot

Before executing the first movement on real robot, it should be brought into initial (home) position (this is also home position for 3D model). The middle of joints range is selected as home position for KHR-1 HV robot. In that posture angle values of all joints are set to 0^o (posture shown in Fig. 4.)

All joint angles ranges of real robot are [-90^o $+90^o$] except for shoulder pitch joint for which range is [-135^o $+135^o$]. Existing control board for robot is replaced with Start USB for AVR prototype board equipped with Atmel AT90USB162 microcontroller.

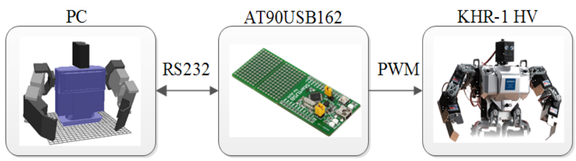


Figure 7. Connection between main PC, microcontroller and robot

Microcontroller should generate appropriate PWM signal according to the specified angle and velocity for each joint. Communication with main computer is performed using serial RS232 communication. In Fig. 7. is illustrated the connection between main computer, microcontroller and robot.

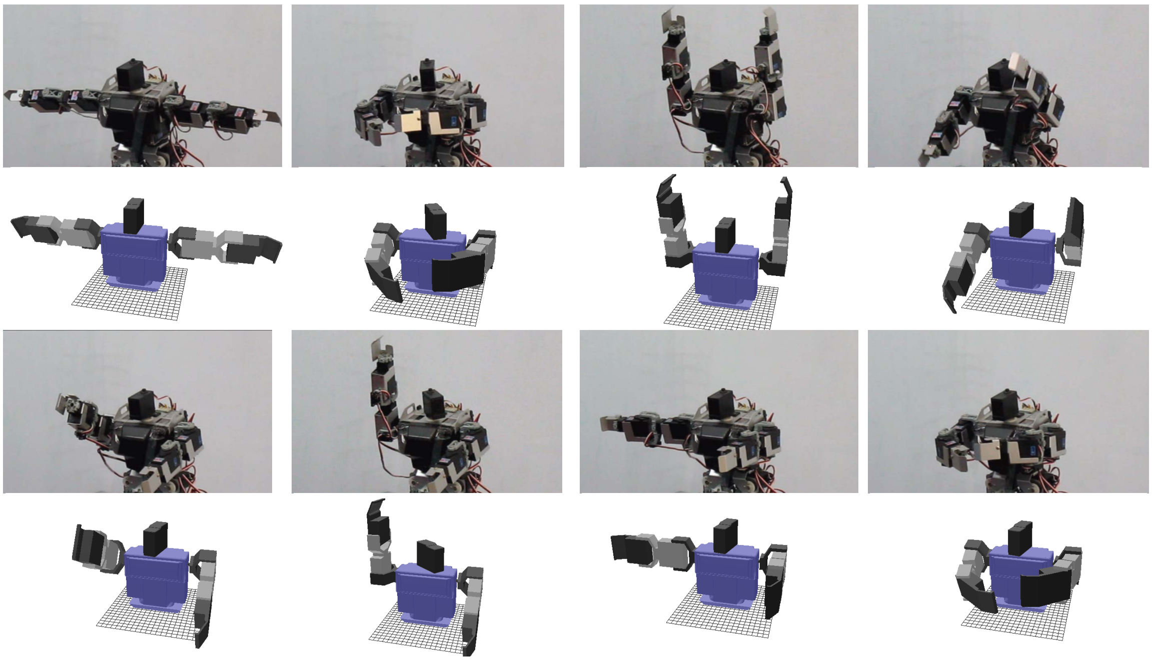


Figure 8. Complex movement performing on real robot and 3D model

Angles and velocities to be performed by the robot are received from main computer in the form of message strings. Each servo drive is labeled (A, B, C, D, E, F and G). Each letter is followed by two numbers representing desired position and velocity of the joint.

The number of different angles that can be achieved with servos is limited by the clock rate and by number of controlled joints. The timer interrupt period is limited by clock rate; shorter period means finer servos resolution. If the period is to short, the microcontroller is not able to receive entire message from main computer and error occurs. This problem is more obvious when the number of servo motors is increased.

The best performance of robot with seven servos controlled by AT90USB162 microcontroller is achieved with timer interrupt that allows 55 independent positions of servo. This means that the resolution of servos with 180^{o} range is 3.27^{o}, and with 270^{o} range is 4.09^{o}.

For testing the software capabilities several basic movements are defined to mimic building blocks of exercise robot intend to perform in future: raise the right arm, raise the left arm, spread the arms, gather the arms, move head to the left, move head to the right etc. The simple movements are combined to form exercises, i.e. complex movements, used for treatment of children with developmental disorders. In Fig. 8 is shown the parallel performance of same exercise on real robot and 3D model. It can be seen from figure that the similarity of movements is good.

V. CONCLUSION

This paper reports the development of simulation and control software for the robot KHR-1 HV. The procedure for generating the complex movements by combining the simple ones is explained.

Presented software is starting point in developing simulator for humanoid robot which will be used as assistance in treatment of children with disabilities. The current solution can be upgraded by using the more powerful microcontroller with dedicated PWM output signals. In this way the resolution of servos will be much better.

ACKNOWLEDGMENT

This work was funded by the Ministry of education and science of the Republic of Serbia under contract III44008 and by Provincial secretariat for science and technological development under contract 114-451-2116/2011.

REFERENCES

[1] Savić S., Jurošević M.: “Design Of Modular Robot Arm With 7 Degrees Of Freedom”, Proceedings of ETRAN, Zlatibor, Serbia, 11-14 June, 2012

[2] Tikhanoff V., Cangelosi A., Fitzpatrick P., Metta G., Natale L., Nori F., “An open-source simulator for cognitive robotics research: The prototype of the icub humanoid robot simulator,” in Proc. IEEE Workshop Perform. Metrics Intell. Syst., Washington, D. C., 2008

[3] Shafii N., Paulo R. L., Rossetti R. J. F., “Two Humanoid Simulators: Comparison and Synthesis”, in Proc. IEEE 6th Iberian Conference on Information Systems and Technologies (CISTI), Porto, Portugal, 2011

[4] Medrano-Cerda G.A., Dallali H., Brown M., Tsagarakis N.G., Caldwell D.G., “Modelling and simulation of the locomotion of humanoid robots”, In UK Automatic Control Conference, Coventry, September 2010.

[5] Arndt D., Bobrow J. E., Peters S., Iagnemma K., Dubowsky S., “Two-Wheel Self-Balancing of a Four-Wheeled Vehicle”, IEEE Control Systems, Vol 31. No. 2. pp. 29-37, April 2011.

[6] Grinblat I., Peterson A., “OGRE 3D 1. 7 Application Development Cookbook”, ISBN: 978-1849514569, Packt Publishing, 2011

[7] Borovac B., Raković M. Nikolić M., “Online Humanoid Robot Walk Generation using Primitives”, 9th IEEE Int. Symp. On Intelligent Systems and Informatics (SISY), Serbia, 2011

[8] Vukobratović M., Borovac B., Raković M., Nikolić M., “Generating Complex Movements of Humanoid Robots by Using Primitives”, Communications in Computer and Information Science, Volume 82. ISBN 978-3-642-16369-2. Springer, Berlin Heidelberg, 2010