Transactions of the VŠB – Technical University of Ostrava, Mechanical Series
No. 2, 2016, vol. LXII
article No. 2015

Stefan GRUSHKO[*], Zdenko BOBOVSKÝ[**]

TELEOPERATED HUMANOID ROBOT

TELEOPERAČNĚ ŘÍZENÝ HUMANOIDNÍ ROBOT

Abstract

Article describes technical solution of teleoperated humanoid robotic system. To acquire position data of operator’s body Kinect sensor is used. In article are described mathematical equations used to transform data from Kinect sensor to positions of each servomotor of the robot. Article also describes software and electric structure for both components of the system: robot and operator’s PC. All software solutions are developed using C#. For dynamic simulation of the system a detailed model of the robot has been created in V-Rep, simulation receives same data as real robot.

Abstrakt

Článek popisuje technické řešení teleoperačně řízeného humanoidního robotického systému. K získávání údajů o poloze těla operátoru je používán senzor Kinect. V článku jsou popsány matematické rovnice použité pro transformaci dat ze senzoru Kinect do pozic jednotlivých servomotoru robotu. V článku je také popsána struktura softwarového a elektronického zařízení obou části systému: robotu a PC operátora. Všechna softwarová řešení jsou vyvinuta s použitím programovacího jazyka C#. Pro dynamické simulace systému byl vytvořen podrobný model robotu ve V-Rep, simulace získává stejná data jako skutečný robot.

Keywords

Teleoperated robot, Kinect, Bioloid, movement recognition, humanoid robot, teleoperation.

1 INTRODUCTION

Teleoperated robots are one of the main trends in robotics of last years. They can be used in wide range of application from day-to-day life to extremely dangerous actions, or those which human cannot perform. Many of them can be used for actions where unhuman strength and accuracy is needed. Surgical robots, for example, can perform operations with minimally invasive surgical procedures.

Also it is planned to use teleoperated robots with virtual reality headsets [1] to improve telepresence programs (such as Skype). Such systems can make cardinally new step in the way of interaction of people on the distance.

By now there have been implemented few systems of teleoperated robots, some of them have been using special suits [1],[2] to acquire operator’s body pose, other have been using systems of machine vision [3] in order to do the same. While systems that are using suits provide more precise motion recognition they have certain disadvantages.

---

[*] Bc, Department of Robotics, Faculty of Mechanical Engineering, VŠB-TUO, 17. Listopadu 15, Ostrava-Poruba, e-mail stefan.grushko.st@vsb.cz

[**] Ing., Ph.D., Department of Robotics, Faculty of Mechanical Engineering, VŠB-TUO, 17. Listopadu 15, Ostrava-Poruba, e-mail zdenko.bobovsky@vsb.cz

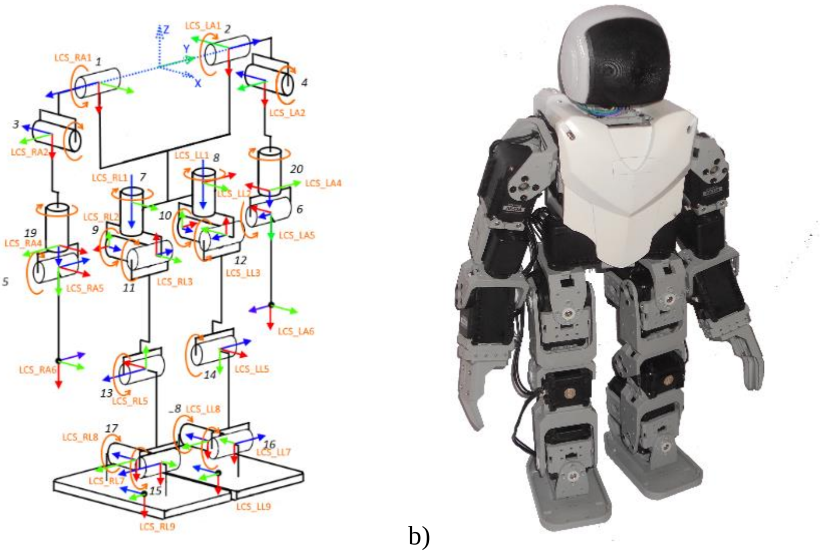
Goal of this project was to create a robotic system to test principles of teleoperation of a humanoid robot, thus main task of the system was to repeat all movements of the operator. It was decided that in this case using of Kinect sensor to recognize operators posture is more convenient, than using specialized suit. System uses Robotis Bioloid humanoid kit [4] with improved kinematic structure and Netduino Plus 2 as main controller.

2 HARDWARE

System has been divided on two parts: the robot itself and the PC that performs all the complex calculations. Sides communicate via Bluetooth. Side of PC includes PC itself and a Kinect sensor. Side of the robot include mechanical structure of the robot, servomotors, sensors and the main controller of the robot. In this section only part of the robot will be described.

2.1 Mechanical part

Robot is based on Robotis Bioloid kit with modified kinematic structure (Fig. 1) that allows robot to follow human movements of arms more precisely. Each arm has one additional DOF – servomotors with ID’s 19 and 20 that allow rotation of the arms in yaw axis. Also position of servomotors with ID 5 and 6 has been changed so they perform rotation in pitch axis.



a) b)

Fig. 1 Mechanical structure of the robot: a) kinematic; b) real.

For demonstration on NATO Days 2016 in Ostrava new head, chest and controller’s cases have been 3D printed using Rapid Prototyping technology.

2.2 Electric part

As main controller of robot was used Netduino Plus 2 with Netduino Shield for STEM that has been designed on Department of Robotics. Netduino Shield allows to control (both reading and writing commands) Dynamixel servomotors.

Complete structure of electric part used on the side of Bioloid is on Fig. 2.

Robot is equipped with an analog 3-axis accelerometer to measure its actual space orientation. The accelerometer is mounted in the head of the robot (Fig.8 – Coordinate system of the robot, Z axis is directed upwards).

Dynamixel servomotors allows to connect themselves in daisy chain architecture – motors of each limb have been serially connected and then attached to STEM Netduino Shield.

Robot communicates with PC using Bluetooth module.

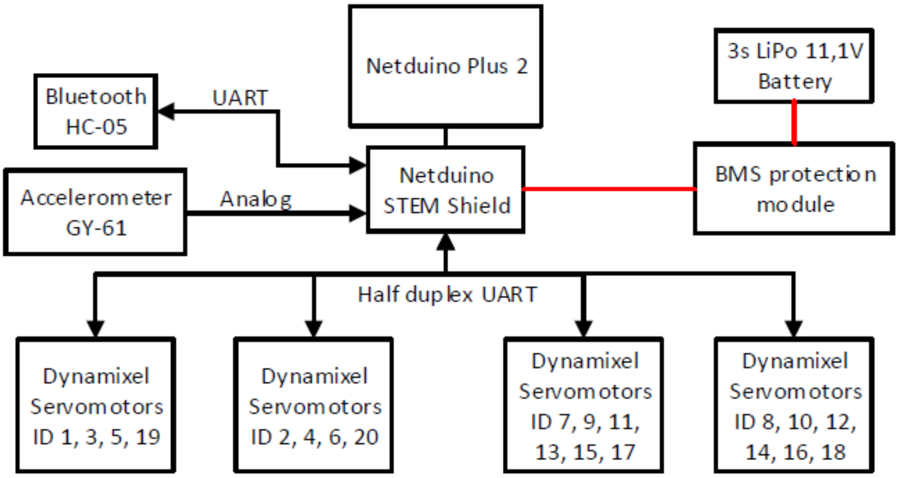


Fig. 2 Principal diagram

For teleoperated mode of the robot 3s LiPo battery is used as energy source, battery is protected from over-discharging by BMS (Battery Management System) protection module. While tested robot is hanged on frame (Fig. 3) construction that allows it to move all limbs freely and stabilizes the robot in space.

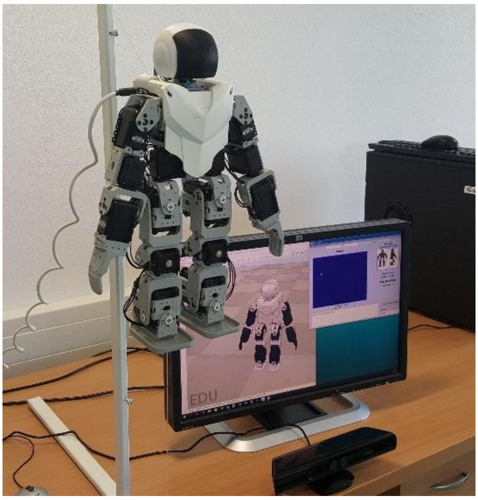


Fig. 3 Testing stand with robot and control and visualization applications (V-rep)

3 SOFTWARE

For controlling of the robotics system have been written applications for both robot and PC. Windows Kinect SDK 1.8 has been used to acquire data from Kinect sensor. Applications are written in C# programming language in MS Visual Studio IDE and are divided to classes.

Since system has been divided on two parts microcontroller doesn’t have to have high processing performance and all complex algorithms run on PC.

3.1 Software of robot

Main thread of Netduino application receives pose data from PC application and then adjust position of each motor to this new pose. Dynamixel servomotors allow to simultaneously adjust positions of multiple motors using “Sync” type of command. This type of command has been implemented with C# class [5] that has been developed on Department of Robotics. It allowed to decrease time that is needed to apply new pose to the robot.

Background thread is responsible for digitalization and sending data from accelerometer to the PC application. For receiving and processing data from an accelerometer analog accelerometer C# class[6] written on Department of Robotics was used.

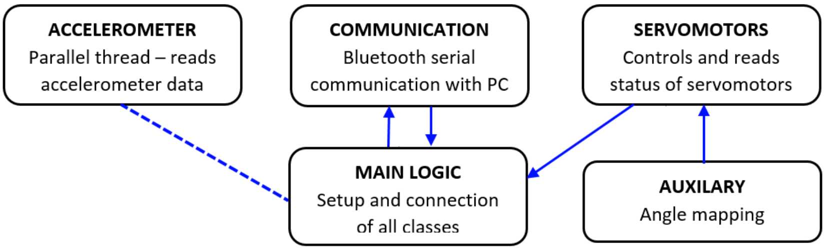
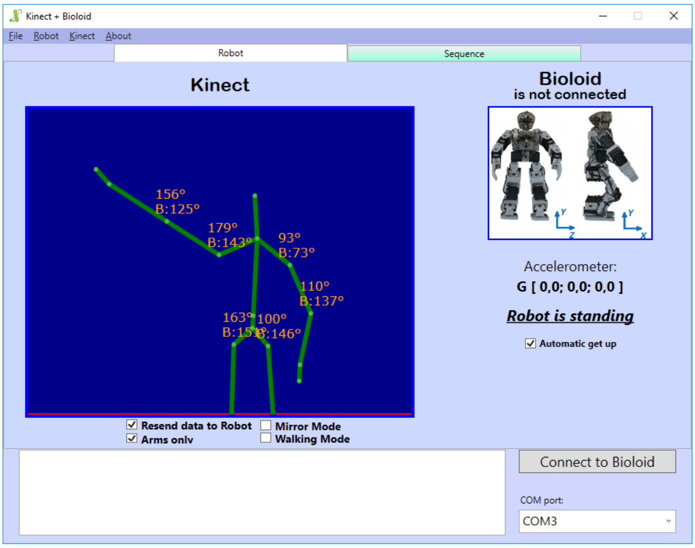


Fig. 4 Software structure of system – part of the robot

3.2 Software of PC

Application for PC (Fig. 5) communicates with robot via Bluetooth serial communication with Baud Rate 57600.

Main program collects skeleton data from Kinect, background thread calculates angles for each servomotor form this data and then sends them to the robot with interval of 20ms.



Bioloid is not connected

Accelerometer:
G [0,0; 0,0; 0,0]

Robot is standing

☑ Automatic get up

☑ Resend data to Robot ☐ Mirror Mode
☑ Arms only ☐ Walking Mode

COM port: COM3

Fig. 5 PC application

The application has two tabs: “Robot” and “Sequence”.

The tab “Robot” allows to choose serial port for robot connection, displays actual data for robot’s accelerometer and Kinect sensor, shows actual space orientation of robot. Also on this tab user can switch between modes such as: automatic get up after falling, mirror movements mode and move only arms.

On tab „Sequences“ user can make adjust and modify own move sequences, such as move sequence of getting up after falling in each direction. Each sequence is created from single poses and have its own time to execute each pose. User can add new pose to any movement sequence – then application send request to the robot its current pose. After receiving this request robot acquires current position of each servomotor and then sends its back to PC application. When movement sequence is played PC application automatically generates sub-poses between main poses and sends them to robot, so movement is smooth. Total number of sub-poses is adjusted automatically and depends on time to play one pose and on minimal interval that is required to send new pose to robot (limited because of communication speed). Certain move sequences cannot be deleted, such as getting up and walking sequences.

When sequence starts to play, PC application automatically sends a request of current pose to the robot and then uses data about current position to achieve first pose in sequence more smoothly (by creating sub-poses). Sequences can be then saved to *Sequence.xml* file. After start application loads all saved sequence from this file.

Kinect for Windows v1 is capable to recognize the position of 6 persons and only 2 full skeletons. It sends found skeletons in separate thread (along with thread of visual and audio information). PC application chooses skeleton that is closest to the Kinect sensor and also checks whether the skeleton is in an appropriate distance range from the center of Kinect’s view.

Parallel thread that runs with intervals of 20ms performs calculation of angles of each joint of skeleton and sends them via serial communication to the robot.

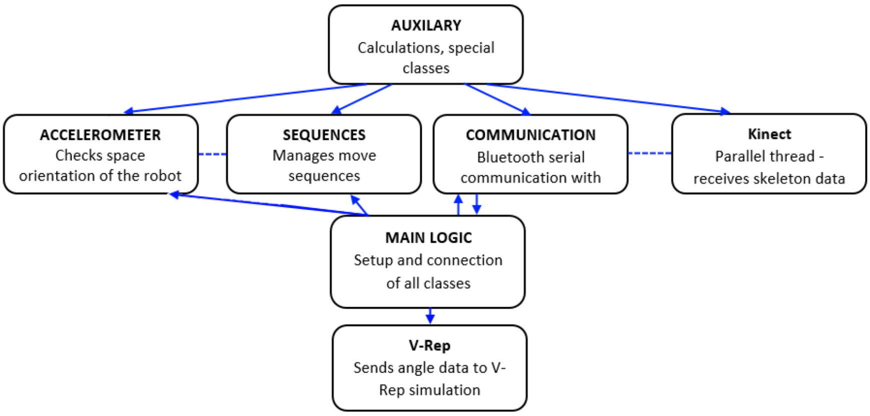


Fig. 6 Software structure of the PC application

For both arms and legs angles calculation same algorithm was used.

Human body has a lot more DOFs that Bioloid humanoid robot does, but DOFs of human limbs were reduced to 4 DOF limbs of the robot. Also, task of 6 DOF control of each leg was reduced to 4 DOF: motors of ankle adapt to currently calculated angle of motors of the hips, so algorithm tries to save parallelism of the robot’s feet due to XY plane of robot.

Kinect provides actual positions X,Y,Z of each joint of the recognized skeleton. From this data are calculated (equations 1-4) vectors r_1, r_2, r_3 that then are used for calculation of a position of each servomotor.

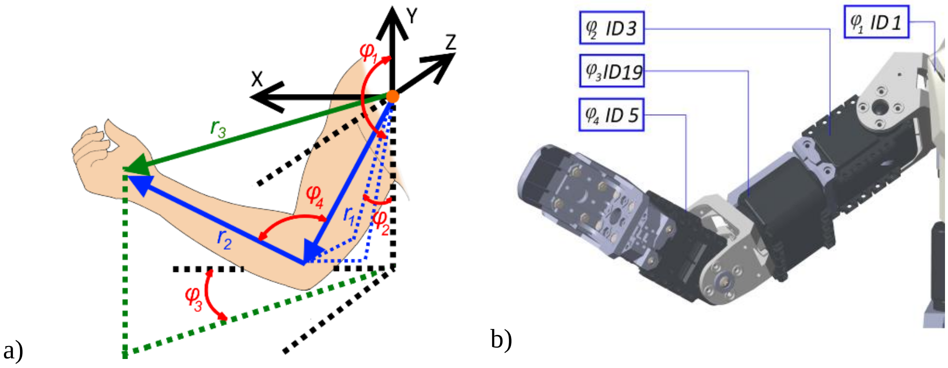


Fig. 7 Calculation of the angles for 4 DOF limb: a) calculation of the angles from Kinect data; b) corresponding to the servomotors of the robot.

Calculations for elbow's yaw and shoulder's motors – angle between two vectors (Fig. 7):

$$\varphi_1 = \arcsin\left(\frac{r_{1x}}{|r_1|}\right), \tag{1}$$

$$\varphi_2 = \arcsin\left(\frac{r_{1y}}{|r_1|}\right), \tag{2}$$

$$\varphi_3 = \arcsin\left(\frac{r_{2x}}{|r_2|}\right), \tag{3}$$

Adjusting the servomotors that perform the pitch rotation in elbows - from the law of cosines:

$$\varphi_4 = \arccos\left(\frac{|r_3|^2 - |r_2|^2 - |r_1|^2}{2 \cdot |r_1| \cdot |r_2|}\right), \tag{4}$$

where:

φ_1 – angle of pitch rotation of the shoulder of the robot [°],

φ_2 – angle of roll rotation of the shoulder of the robot [°],

φ_3 – angle of yaw rotation of the shoulder of the robot [°],

φ_4 – angle of rotation of the elbow of the robot [°],

r_1 – 3D vector from shoulder to elbow,

r_2 – 3D vector from elbow to wrist,

r_3 – 3D vector from shoulder to wrist,

r_{nx} – X component of vector *n*,

r_{ny} – Y component of vector *n*.

The same calculations are provided for other arm and both legs of the robot. Because the working range of Dynamixel AX-12A servomotors in joint mode is limited to 0..300° each of the calculated angles have to be edited by adding or subtracting of calibration angle, so the result fits to the kinematic structure of the robot. Each joint of the robot has its own limitations of angles that decreases probability of self-collision of the system.

The system recognizes 6 states of his space orientation (each possible orientation has been dedicated a number 0..5 – Fig. 8):

- 0 – Staying straight.
- 1 – The robot has fallen on its lying on front side.
- 2 – The robot has fallen on its lying on its right side.
- 3 – The robot has fallen on its lying on its back.
- 4 – The robot has fallen on its lying on its left side.
- 5 – The robot is upside down.

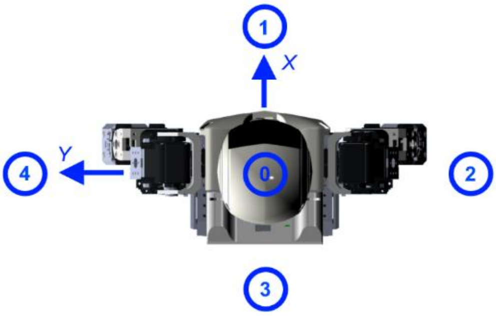


Fig. 8 Labelling of space orientations of the robot

After a fall has been detected the robot automatically starts to play appropriate sequence of getting up.

2.1 V-rep simulation

Simultaneously with sending data to the real robot data is sent to virtual simulation of robot in V-rep[7]. Kinematic structure simulation of the robot is same as real kinematic structure of the system.

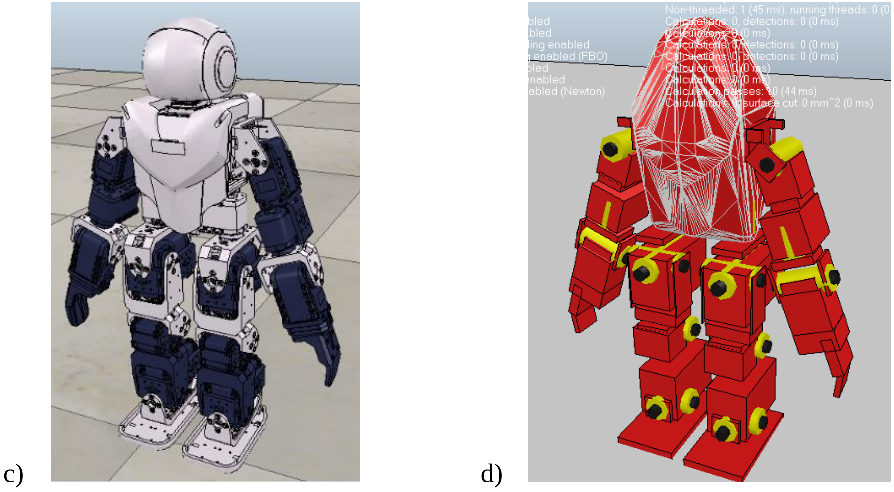


Fig. 9 V-rep simulation model: a) visual; b) dynamic.

In the simulation simplified dynamic model of the robot has been used. This model requires less processing power and allows the simulation run more smothly.

4 CONCLUSIONS

Described architecture of the software can be used in similar projects with different kinematic structure. The system can easily be modified to use operator’s posture data from a motion capture suit instead of using a Kinect sensor. Since working on the system is still in progress it fulfils its main goal – copying movements of operator.

Several subsystems will be added to the robot in the future – anti-collision system, stabilizing subroutine while copying mode is on. Anti-collision system should run on Netduino controller and use simplified 3D model of the robot to recognize collisions. Stabilizing subroutine will recognize the position of the center of the mass of the robot and automatically stabilize it if deviation from a stable position occurs.

To decrease hesitating and sudden movements filtering can be used, so data from Kinect will be first filtered and smoothed, before sending to the robot. Filtering will affect system’s performance, but will reduce influence of inaccurate skeleton recognition.

More stable walking sequence or dynamic generating of gait can be solved by neuron network in collaboration with genetic algorithms.

ACKNOWLEDGMENTS

This article has been supported by specific research project SP2016/142 and financed by the state budget of the Czech Republic.

REFERENCES

[1] LabAtar - Humanoid Robot Teleoperation System. http://letsmakerobots.com/ [online]. 2015 [cit. 2016-10-22]. Dostupné z: http://letsmakerobots.com/node/44269

[2] Humanoid Robot Mahru Mimics a Person's Movements in Real Time. IEEE SPECTRUM. Http://spectrum.ieee.org/ [online]. 2010 [cit. 2016-10-22]. Available from: http://spectrum.ieee.org/automaton/robotics/humanoids/042710-humanoid-robot-mahru-real-time-teleoperation

[3] Romeo Robot Immersive Teleoperation with VR Headset & LEAP. RoboticGizmos. http://www.roboticgizmos.com/ [online]. 2010 [cit. 2016-10-22]. Available from: http://www.roboticgizmos.com/robot-vr-teleoperation/

[4] Bioloid Kit. Robotis. http://www.robotis.com. [online]. 2015 [cit. 2016-10-22]. Available from: http://www.robotis.com/xe/bioloid_en

[5] GRUSHKO, Stefan, Ing. Zdenko BOBOVSKÝ, PHD. *Dynamická knihovna pro řízení pohonu Dynamixel*[online]. In: Department of Robotics [cit. 2016-10-22]. Available from: http://robot.vsb.cz/aplikovane-vystupy/software/210/

[6] GRUSHKO, Stefan, Ing. Zdenko BOBOVSKÝ, PHD. *Dynamická knihovna pro analogový akcelerometr*[online]. In: Department of Robotics. 2015 [cit. 2016-10-22]. Available from: http://robot.vsb.cz/aplikovane-vystupy/software/215/

[7] V-Rep: Virtual robot experimentation platform. In: Coppelia Robotics [online]. 2016 [cit. 2016-10-22]. Available from: http://www.coppeliarobotics.com/

[8] ŠELOMENCEV, E.E. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ KINECT УПРАВЛЕНИЯ АНДРОИДОМ BIOLOID [online]. Tomska polytechnická univerzita [cit. 2016-10-22]. Available from: http://www.lib.tpu.ru/fulltext/c/2013/C01/V2/198.pdf

[9] FILIATRAULT, Sylvain, Ana-Maria CRETU. *Human Arm Motion Imitation by a Humanoid Robot*. 2008. Department of Computer Science and Engineering. Université du Québec en Outaouais.