

Łukasz Kwaśniewicz*, Wiesława Kuniszyk-Jóźkowiak, Grzegorz M. Wójcik and Jolanta Masiak

Adaptation of the humanoid robot to speech disfluency therapy

DOI 10.1515/bams-2016-0018

Received September 27, 2016; accepted October 24, 2016; previously published online November 16, 2016

Abstract: The paper describes an application that allows to use a humanoid robot as a stutterer's assistant and therapist. Auditory and visual feedback has been used in the therapy with a humanoid robot. For this purpose, the common method of "echo" was modified. The modification is that the speaker hears delayed speech sounds uttered by the robot. The sounds of speech coming from an external microphone are captured and delayed by a computer and then, using User Datagram Protocol (UDP), sent to the robot's system and played in its speakers. This system allows the elimination of negative feedback and external sound field's noise. The effect of this therapy is enhanced by the fact that, in addition to the effect, relating to the action of the delayed feedback, the speaker has company during the difficult process of speaking. Visual feedback has been realized as changes in the robot's hand movements according to the shape of the speech signal envelope and possibility of controlling speech with a metronome effect.

Keywords: echo method; humanoid robot; stuttering therapy; visual feedback.

Introduction

Speech fluency disorder, commonly known as stuttering, affects approximately 1% of the population of our planet. It causes a lot of trouble and difficulties for the stutterers at school, and it also creates complexes and mental anguish.

***Corresponding author: Łukasz Kwaśniewicz**, Institute of Computer Science, Department of Neuroinformatics, Marie Curie-Skłodowska University, ul. Akademicka 9, 20-033 Lublin, Poland, E-mail: lukasz.kwasniewicz@poczta.umcs.lublin.pl

Wiesława Kuniszyk-Jóźkowiak: Faculty of Physical Education and Sport in Biała Podlaska, Józef Piłsudski University of Physical Education in Warsaw, Akademicka 2, 21-500 Biała Podlaska, Poland

Grzegorz M. Wójcik: Institute of Computer Science, Department of Neuroinformatics, Marie Curie-Skłodowska University, ul. Akademicka 9, 20-033 Lublin, Poland

Jolanta Masiak: Neurophysiological Independent Unit, Medical University in Lublin, Głuska 2, 20-439 Lublin, Poland

The etiology of this disorder is an unsolved mystery so far. All over the world, intensive research is conducted on efficient methods of disposal of that speech defect. The most popular and most effective method is based on the exercises of speech with the delayed auditory feedback (DAF). This method of therapy was proposed and applied for the first time by Bogdan Adamczyk, a professor at Marie Curie-Skłodowska University (UMCS) [1]. This method, called "echo", was widespread. In 1987, it was used in more than 300 treatment centers in Poland, where there was an analog device, using simultaneous recording and playback of delay, built at the UMCS's Institute of Physics [2]. It was also used in the form of an echo-type speech correction phone system [3]. Later, digital devices and computer programs were used in the therapy to produce artificial "echo" [4, 5]. The idea of the echo method is that, when a stutterer speaks simultaneously with his own speech delayed by approximately 0.2 s, speech disfluencies disappear [6]. Frequent exercises help in such situation to accrue a long-lasting effect of improving the flow of speech. The DAF is still used in speech disfluency therapy, often in combination with other methods such as filtered feedback (FAF) [7–13].

To obtain fluency in the "echo" method, it is common to practice daily in the presence of a therapist or another person. What plays an important role in the therapy is a choir factor. It has been shown that stuttering disappears in the situation of choral speaking with others by reducing the fear of speaking [14]. The average exerciser does not feel alone or stressed about his or her actions, which in this case means speaking. That is the role of a humanoid robot, which delayed the sounds of the exerciser's own speech in the echo method. It is possible to increase the therapeutic effects by adding delayed visual feedback [15]. The present paper describes the design of a humanoid robot program, allowing the implementation of the visual and auditory feedback.

This article presents the implementation of the echo method using a humanoid robot. Humanoid robots' application has already been proven effective in the case of aphasia [16].

In the implementation, as presented in this article, a humanoid robot utters the exerciser's delayed speech sounds, so the stutterer has a feeling of speaking with the choir. It also controls the speech process by the robot's

arm movement in tact of the spoken sound's envelope or with a specified speed. The robot acts as a companion and therapist, so the treatment should be more effective and attractive. The attractiveness of this method is particularly important in the treatment of children, in which it is important to overcome the reluctance to speak.

Humanoid robot NAO

The humanoid robot NAO was made by Aldebaran Robotics. The robot's dimensions are $573 \times 311 \times 275$ mm. The weight of NAO is 5.2 kg.

Besides 26 servos and sensors responsible for the movement, such as two gyroscopes, accelerometer, eight tactile sensors, and 36 Hall effect sensors, it has a lot of other components – two speakers, four microphones, two 1220 p cameras that can capture up to 30 images/s, two passive infrared sensors, and two sonar channels: two transmitters and two receivers.

The robot is controlled by the Linux system, which is installed on the machine equipped with Intel Atom Z530 processor, 1 GB of RAM, and 2 GB of flash memory, with the possibility of extension to 8 GB, thanks to the card reader [17].

NAO supports Wi-Fi (bgn) and Ethernet, which are currently the most widespread network communication protocols. In addition, infrared transceivers, located in his eyes, allow connection to objects in the environment. NAO is compatible with the IEEE 802.11b/g/n Wi-Fi standard and can be used on both WPA and WEP networks, making it possible to connect him to most home and office networks. NAO's OS supports both Ethernet and Wi-Fi connections and requires no Wi-Fi setup other than entering the password [18].

Echo therapy robot application

The schematic diagram of application is presented in Figure 1.

The application consists of two parts. The first part, the client application, is installed on the desktop computer or laptop, and the second part, the server application, is installed on the NAO's system.

This division guarantees no feedback between the microphone, into which the exerciser speaks, and the

sound of the robot's speakers. It also allows one to place the microphone close to the mouth of the speaking person. It eliminates external noise and improves the sound quality.

This method allows the conversation of many stuttering people with a single robot.

Although the robot NAO is equipped with microphones, these microphones are not used for sound capture. This is caused by the occurrence of feedback and noise, which makes the therapy with the robot impossible.

The client application has a graphical user interface. It connects automatically to the program, which is installed on the NAO robot's system, selecting IP address and port. These two parts of the program are made in Java language, and they work both with the Windows and Linux systems, provided that Java Virtual Machine has been installed.

Client application

The graphical user interface is presented in picture number 2.

The presented application contains four modes of work: auditory feedback, visual feedback, auditory and visual feedback, and "metronome" mode.

Auditory feedback mode

In the described feedback, the client application, which is installed on the user's computer, captures the sound from the microphone, writes it to the buffer, and sends this buffer with the required delay, using the Internet and User Datagram Protocol (UDP) protocol, to the server application installed on the humanoid robot NAO. Then, all of the buffer's content is played by the robot's speakers.

Before treatment, the user (or speech therapist) sets the required delay (Figure 2). The delay can be regulated from 0.15 to 0.5 s, according to formula (1):

$$dt = (\text{buffer size} / \text{sample rate}) \quad (1)$$

The sample rate is specified in the object's field of Audio Format class.

The function, which returns Audio Format object, is described in the Appendix.

While calculating the buffer size, it must be remembered that the sample size is 16 bits, or 2 bytes, so the result must be multiplied by 2, according to formula (2):

$$\text{buffer size} = dt * \text{sample} * 2 \quad (2)$$

In this application, Java Sound API has been used to capture and playback sounds.

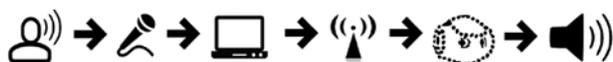


Figure 1: Schematic diagram of echo therapy robot's application.



Figure 2: Client application's graphic user interface.

Also, Mixer, Line, and AudioFormat classes have been used. They allow to choose devices, set audio data format, and stream this data to the buffer. Listing 1 presents the audio format that has been used.

The application searches the network for the server application then connects to it and sends information about the delay and informs that it is a client application.

If the buffer is filled, it is sent to the server application by UDP protocol.

Sending sound to the robot

The main task of the application is to send sound from the microphones, which are plugged to the user's computer, by network to the robot's speakers with some delay, causing the impression of echo. This action can be described as data streaming.

The biggest difference between streaming and downloading is the fact that when data are downloaded they cannot be opened until download is finished. Streaming allows to open data while they are received.

There are two types of data streaming: (a) on demand – streams are available on the server for a long time and they are ready to playback and (b) live – data are available only at a particular time, such as radio broadcast, and there is no possibility to rewind broadcast [19]; this mode was used in the described application.

In this case, the server is the process running on the robot's system. This process waits for the data, which are sent via UDP protocol by the client – the process can be run on the computer with any system that has Java virtual machine. Then, data are sent to the robot's speakers by the server application.

UDP gives the application's programs direct access to services, providing datagrams. This allows applications to exchange information over the network with minimal overhead, resulting from the implementation of the protocol.

UDP is an uncertain, connectionless protocol of datagram transfer. It is uncertain because it does not have mechanisms to check whether the data arrived correctly to its destination. UDP data transfer protocol is the best

Listing 1: Function that returns an object of AudioFormat class.

```
public static AudioFormat getAudioFormat () {
// sample rate:
float sampleRate = 16000.0F;
// sample size - the number of bits per sample:
int sampleSizeInBits = 16;
// number of channels:
int channels = 1;
// determines whether the data is signed or unsigned type:
boolean signed = true;
// determines whether a single sample is stored in the order of big endian or little endian
bigEndian boolean = false;
// Returns AudioFormat with the values, which are set above:
return new AudioFormat (sampleRate, sampleSizeInBits, channels, signed, bigEndian);
}
```

		UDP Header																															
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

Figure 3: UDP header.

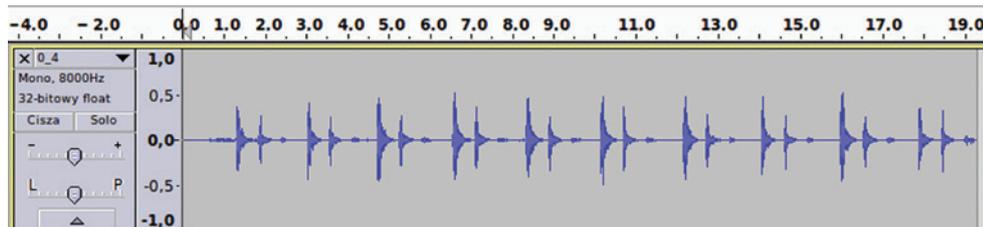


Figure 4: Record that is opened by the Audacity program.

choice for streaming audio in real time. For the price of losing a small amount of data packets, which do not affect the proper operation of applications, UDP can reduce the size of the protocol header, minimize delays in the delivery of data, program additional features, and keep one-to-many transmission [20].

The first word of the UDP header uses 16-bit numbers of the source port and destination port. The aim is to provide data to the correct application.

Figure 3 shows the UDP message format.

The application, which sends the buffer contents using the UDP protocol, uses a Java class library: DatagramSocket and DatagramPacket. The Object constructor, named DatagramSocket, contains the following parameters: Soundpacket — an array of bytes representing the buffer, Soundpacket.length — length of the array, the destination address, and port number of the recipient.

Thus prepared, the object is sent with the method of DatagramSocket class:send (datagram).

Server application

After receiving the relevant information from the client application, the audio format and server buffer are set; then, the server buffer’s content is played using the same classes as in the client part, through the speakers of the humanoid robot, as in the case of recording sound from a microphone into a buffer.

Tests of delays in playback audio, conducted using the Audacity program, show that an additional delay, caused by UDP data transfer over the network, affects

the performance of the application and should be taken into account while calculating the size of the buffer.

Ten measurements were made for the delays of 0.05, 0.1, 0.15, 0.20, 0.25, 0.30, 0.35, and 0.40 to calculate the difference between the required delay and the actual delay, amended by the additional delay resulting from the transmission of data.

The measurement is made by recording, with the aid of an additional external microphone, the operation of this application. User hits the microphone of the computer, on which the client application is working, this hit creates a short sound, and then the humanoid robot plays that sound with delay.

Then, the recording is played in the Audacity program. Figure 4 shows that the sound caused by hitting the microphone and its repetition by the robot can be seen in the program as two “peaks”.

The time difference of those two peaks shows the actual delay of the audio, repeated by a robot.

Table 1 shows the means of 10 measurements results — an actual delay and the difference in relation to the required delay for some selected delays.

Table 1: Table of measurements.

Set delay	Real delay	Difference
0.05	0.193	0.143
0.1	0.236	0.136
0.15	0.294	0.144
0.2	0.336	0.136
0.25	0.387	0.137
0.3	0.439	0.139
0.35	0.491	0.141
0.4	0.545	0.145

Listing 2: Part of the code that is responsible for writing the signal into the buffer and generating delay.

```
// Information about the line – description of the class that implements the interface TargetDataLine and Audio format:
DataLine.Info dataLineInfo = new DataLine.Info (TargetDataLine.class, getAudioFormat ());
// copy of the class that implements the interface TargetDataLine:
TargetDataLine targetDataLine = (TargetDataLine) AudioSystem.getLine (dataLineInfo);
// Opens lines with a specific Audio format:
targetDataLine.open (getAudioFormat ());
// Allows the line to engage in data I / O:
targetDataLine.start ();
// Set the size of the buffer according to the formula 3:
tempBuffer = new byte [(int) (dt * getAudioFormat (). getSampleRate ()) * 2 - 0.14 * getAudioFormat (). getSampleRate () * 2];
// Record of the data, captured from line to the buffer:
while (true)
{
targetDataLine.read (tempBuffer, 0, tempBuffer.length);
// further the application sends data from the filled buffer by using the UDP protocol:
sendWithUDP (tempBuffer);
}
```

The delay, resulting from the transfer of the signal, is equal to about 0.14 s and that is the reason why the actual time delay is adjustable from 0.15 to 0.40 s. This is the range used in the therapy of echo method. The buffer size should be calculated with taking into account the difference in accordance with formula (3):

$$\text{real buffer size} = dt * \text{sample rate} * 2 - 0.14 * \text{sample rate} * 2 \quad (3)$$

Listing 2 presents the source code that is responsible for generating the delay and writing signal into the buffer,

Visual feedback mode

This mode is based on the fact that the robot raises and lowers its hand in tact of the user’s speech. To calculate the envelope’s value for the particular signal’s windows from the buffer, the auditory signal, written in the buffer and described in detail above, is transformed from the byte array to the integer array — sample’s value. After that, the signal is transformed by the time function called “rectangle window” (Listing 3). Then, the right envelope is calculated as formula (4) for each window:

$$\text{wyn} = 20 * \text{Math.log10}(\text{Math.sqrt}(\text{suma}) / 100) \quad (4)$$

The result is scaled on the value from – 1.2 to 1.2.

The range of the robot’s hand movement in radians (up and down) gives the natural effect of “conducting”, controlling the robot’s hand movement by the auditory signal.

Results that are smaller than 65 are conceded as the silence and they set the position of the robot’s hand on the lowest value (hand lowered down).

The value of the robot’s hand is the parameter of the function moveHand() from the class HandControl, responsible for the robot’s hand movement. The class HandControl uses the library, provided by the robot’s producers — Aldebaran Robotics. Then, the session of the robot’s movement is created (Listing 4):

```
Application application=new Application(args,robotUrl);
application.start();
motion=new ALMotion(application.session());
```

The name of joint which the robot will move — “LshoulderPitch” — is added to the collection ArrayList<String>. In this example, it concerns the movement of the robot’s left arm (Figure 5).

```
names=new ArrayList<>();
names.add(“LshoulderPitch”);
```

The scaled value of the envelope’s result in radians is added to the new list:

```
angles=new ArrayList<>();
angles.add(wynik);
```

Eventually, the robot moves the left arm with the calculated angle and estimated speed.

Speed is estimated from range 0–1 to keep the natural movement effect (0.46):

```
motion.angleInterpolationWithSpeed(names,angles,pFractionMaxSpeed);
```

Next, the value from angles list is erased:

```
angles.clear();
```

Listing 3: Part of the code that is responsible for calculating and scaling the envelope's value.

```

public void informuj(int[] tab) {

    int max=tab.length;
    int min=0;
    int nrOkna=0;
    int liczbaOkien = (max-199)/199+1; //liczba okien
    double suma=0; //suma potęg wartości próbek w poszczególnym oknie
    while(min<=max-199)
    {
        nrOkna++;
        suma=0;
        int i=min;
        for(;i<min+199 ;i++)
        {

            suma=suma+(tab[i])*(tab[i]);

        }
        wyn =20*Math.log10(Math.sqrt(suma)/100);

        if (wyn<=65)
        {
            wyn2=-1.2; //scale(wyn,60,75,-1.2,1.2);
        }
        else {
            wyn2=scale(wyn,65,75,-1.2,1.2);

        }
        min=min+199;
        hc.moveHand(-wyn2);
    }
}

```

The application recalculates new data from sound signal and calls the aforementioned method. The auditory and visual feedback mode uses two described feedbacks at the same time. The metronome mode allows the robot hand's movement up and down from the lowest to the highest location with the speed estimated from 0 to 1, where 0 means no movement and 1 means fast movement: two full movements of the hand per second.

Conclusions

The main advantage of this application is the possibility of using a humanoid robot in the process of therapy using the “echo” method, extended by the visual feedback — precisely by the robot's hand movement in tact of the speaking person. This robot, in case of specific actions,

can replace a specialist, for example, the speech therapist, while accompanying the patient and leading his treatment. The therapy indicated above is based on the fact that the patient, a person stuttering, performs various types of exercises such as reading, conversing, or running a monologue.

These activities should be carried out as frequently as possible and should last for at least 20 min every day.

The process of speaking belongs to social activities. Usually, a speech therapist should be a part of the therapy, which helps solving the problem of speech disfluency. Thanks to the presented application, the specialist can be replaced by the NAO robot during the mechanically repeated activities.

To make the described therapy more attractive, in the development version of the application, it will be possible to use the function of asking earlier prepared questions

Listing 4: Part of the code that is responsible for calculating and scaling the envelope's value.

```

public class HandControl{
    float pFractionMaxSpeed = 0.46f;
    String robotUrl = "tcp://nao.local:9559";
    String[] args=new String[1];
    Application application;
    ALMotion motion ;
    List<String> names;
    List<Double> angles ;
    public HandControl(String url) throws Exception
    {
        args[0]="appc";
        robotUrl = url;
        Application application = new Application(args,robotUrl);
        application.start();
        motion= new ALMotion(application.session());
        motion.wakeup();
        names = new ArrayList<>();
        names.add("LShoulderPitch");
        angles =new ArrayList<>();

    }
    public void moveHand(double wynik)
    {
        try {

            angles.add(wynik);
            motion.angleInterpolationWithSpeed( names,angles,pFractionMaxSpeed );

            angles.clear();

        } catch (Exception ex) {
            Logger.getLogger(HandControl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

directly by the robot, so the patient would get the impression that he participates in a real conversation, and that would allow him to break through the social barriers associated with shyness, resulting from the possessed speech defects.

The application, which is the essence of this publication, provides the possibility of adjusting the time delay, further allowing the individualization of the therapy (first longer delay and then smaller delay with progress).

Another advantage of the described program is the ability to remotely connect to the robot, which eliminates harmful feedback and external noise.

Finally, it should be noted that the application will be tested on a group of people with speech disfluency and compared to other methods, for example, the stationary

“echo” method. Its further development will mainly rely on the preparation of additional programmable exercises, particularly making the conversation with the robot possible.

There are also many other therapeutic applications of NAO robot.

Humanoid robot NAO is used for therapy on selected laterality disorders in children [22].

One of the most important is using the humanoid for interaction with children suffering from autism spectrum disorder (ASD).

Thanks to the possibility of relatively easy application development for NAO, there have been several papers reporting research of children playing games with robots. The game-based approach using NAO is useful for teaching children how to express their feelings in

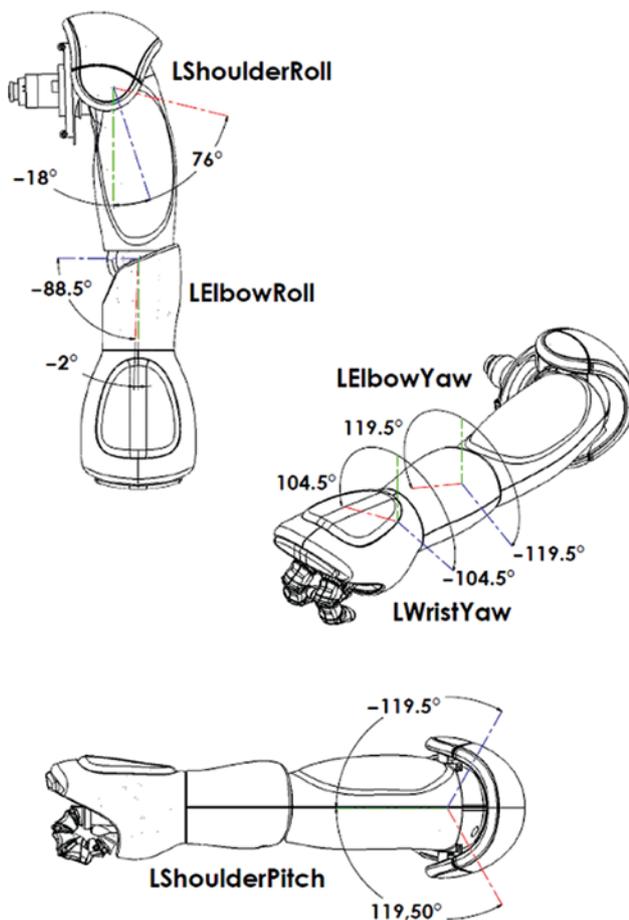


Figure 5: Hand's joints of robot and range of movement [21].

appropriate situations [23]. There have been special programs developed to help children with ASD in the areas of socialization, communication, and playful behavior through robot-based intervention [24, 25]. It is believed that, thanks to some skills of NAO such as blinking eyes and human-like gestures, some children may be encouraged to draw their attention and interact with the robot [26].

Using the humanoid robot NAO causes an increased concentration of the child's attention on the exercises and willingness to repeat these exercises [27].

There are also investigations trying to find a link between stuttering (actually speech fluency) and ASD [28]. On the contrary, we have some experience in modeling attention and consciousness in ASD [29–32].

This is the reason why a deeper investigation of ASD with the application of NAO robot will be in our future interest and research to be conducted at the Department of Neuroinformatics.

Author contributions: The authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Research funding: None declared.

Employment or leadership: None declared.

Honorarium: None declared.

Competing interests: The funding organization(s) played no role in the study design; in the collection, analysis, and interpretation of data; in the writing of the report; or in the decision to submit the report for publication.

References

1. Adamczyk B. Anwendung des Apparates für die Erzeugung von künstlichem Wiederhall bei der Behandlung des Stotterns. *Folia Phoniatri* 1959;11:216–8.
2. Adamczyk B. 30-Years of the “echo-method”. In: *Stuttering. Proc. of IX Scientific Conference of the Polish Logopaedic Society, Lublin, 1987:7–14* (in Polish).
3. Adamczyk B. Speech correction method and a device by telephone to this method. Home Patent 47696, 1963.
4. Van Borsel J, Reunes G, Van den Bergh N. Delayed auditory feedback in the treatment of stuttering: clients as consumers. *Int J Language Commun Disord* 2003;38:119–29.
5. Radford N, Tanguma J, Gonzalez M, Nericcio MA, Newman D. A case study of mediated learning, delayed auditory feedback, and motor repatterning to reduce stuttering. *Percept Motor Skills* 2005;101:63–71.
6. Gallop RF, Runyan CM. Long-term effectiveness of the SpeechEasy fluency-enhancement device. *J Fluency Disord* 2012;37:334–43.
7. Botterill W. Developing the therapeutic relationship: from ‘expert’ professional to ‘expert’ person who stutter. *J Fluency Disord* 2011;36:158–73.
8. Packman A, Meredith G. Technology and the evolution of clinical methods for stuttering. *J Fluency Disord* 2011;36:76–85.
9. Van Borsel J, Drummond D, Medeiros de Britto Pereira M. Delayed auditory feedback and acquired neurogenic stuttering. *J Neurolinguist* 2010;23:479–87.
10. Foundas AL, Mock JR, Corey DM, Golob EJ, Comture EG. The SpeechEasy device in stuttering and nonstuttering adults: fluency effects while speaking and reading. *Brain Language* 2013;126:141–50.
11. Adamczyk B, Kuniszuk-Józkowiak W, Smółka E. Influence of echo and reverberation on the speech process. *Folia Phoniatri* 1979;31:70–81.
12. Adamczyk B, Kuniszuk-Józkowiak W, Sadowska E. Correction effect in chorus speaking by stuttering people. In: *XVIth International Congress of Logopedics and Phoniatrics, Interlaken, 1976:2–6*.
13. Choe YK, Jung HT, Baird J, Grupen RA. Multidisciplinary stroke rehabilitation delivered by a humanoid robot: interaction between speech and physical therapies. *Aphasiology* 2013;27:252–70.

14. Kuniszyk-Jóźkowiak W, Dzieńkowski M, Smółka E, Suszyński W. Computer diagnosis and therapy of stuttering. *Structures-Waves-Human Health* 2003;XII:133–44.
15. Dzieńkowski M, Kuniszyk-Jóźkowiak W, Smółka E, Suszyński W. Computer speech echo corrector. *Ann UMCS Inf* 2004:315–322.
16. Kuniszyk-Jóźkowiak W, Smółka E, Adamczyk B. Effect of visual and tactile reverberation on speech fluency of stutterers. *Folia Phoniatri Logopaed* 1997;49:26–34.
17. <http://www.roboshop.pl/nao.html>. Accessed: 7 June 2016.
18. <https://www.aldebaran.com/en/more-about>. Accessed: 7 June 2016.
19. Hunt C. *TCP/IP-Network Administration. Oficyna Wydawnicza READ ME*, 1996:24.
20. <http://home.agh.edu.pl/~opal/sieci/instrukcje/stream.pdf>. Accessed: 7 June 2016.
21. Gardecki A, Kawala-Janik A. The application extending the capabilities of the human-robot interaction of the humanoid robot NAO. *Prace Inst Elektrotech* 2016;63:EN 45–54.
22. http://doc.aldebaran.com/2-1/family/robots/joints_robot.html. Accessed: 7 June 2016.
23. Miskam MA, Masnin NF, Jamhuri MH, Shamsuddin S, Omar AR, Yussof H. Encouraging children with autism to improve social and communication skills through the game-based approach. *Procedia Comput Sci* 2014;42:93–8.
24. Ismail LI, Shamsudin S, Yussof H, Hanapiah FA, Zahari NI. Robot-based intervention program for autistic children with humanoid robot NAO: initial response in stereotyped behavior. *Procedia Eng* 2012;41:1441–7.
25. Shamsuddin S, Yussof H, Ismail LI, Mohamed S, Hanapiah FA, Zahari NI. Initial response in HRI—a case study on evaluation of child with autism spectrum disorders interacting with a humanoid robot NAO. *Procedia Eng* 2012;41:1448–55.
26. Shamsuddin S, Yussof H, Ismail LI, Mohamed S, Hanapiah FA, Zahari NI. Humanoid robot NAO interacting with autistic children of moderately impaired intelligence to augment communication skills. *Procedia Eng* 2012;41:1533–8.
27. Fosnot SM, Jun S. Prosodic characteristics in children with stuttering or autism during reading and imitation. In: *Proceedings of the 14th International Congress of Phonetic Sciences*. Berkeley, CA: University of California, 1999:1925–8.
28. Duch W, Nowak W, Meller J, Osiński G, Dobosz K, Mikołajewski D, et al. Consciousness and attention in autism spectrum disorders. In: *Proceedings of Cracow Grid Workshop, 2011*;2010:202–11.
29. Duch W, Nowak W, Meller J, Osiński G, Dobosz K, Mikołajewski D, et al. Computational approach to understanding autism spectrum disorders. *Computer Science* 2012;13:47–61.
30. Dobosz K, Mikołajewski D, Wójcik GM, Duch W. Simple cyclic movements as a distinct autism feature—computational approach. *Comput Sci* 2013;14:475–89.
31. Wójcik GM, Ważny M. Bray-Curtis metrics as measure of liquid state machine separation ability in function of connections density. *Procedia Comput Sci* 2015;51:2979–83.
32. Kawala-Janik A, Gardecki A, Podpora M, Kolańska-Płuska J, Grochowicz B. Implementation of the LES systems and NAO humanoid robots in children. In: *Międzynarodowa Konferencja z Podstaw Elektrotechniki i Teorii Obwodów IC-SPETO, Gliwice-Ustron, 20–23 maj 2015*.