Microprocessors and Microsystems 69 (2019) 179–187

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Speech-recognition cloud harvesting for improving the navigation of cyber-physical wheelchairs for disabled persons

Andrej Koložvari[a], Radovan Stojanović[b], Anton Zupan[c], Eugene Semenkin[d], Vladimir Stanovov[d], Davorin Kofjač[a], Andrej Škraba[a,*]

[a] Cybernetics & Decision Support Systems Laboratory, Faculty of Organizational Sciences, University of Maribor, Kidričeva cesta 55a, Kranj 4000, Slovenia
[b] Faculty of Electrical Engineering, University of Montenegro, Džordža Vašingtona bb, Podgorica 81000, Montenegro
[c] University Rehabilitation Institute, Republic of Slovenia, Soča, Linhartova 51, Ljubljana 1000, Slovenia
[d] Reshetnev Siberian State University of Science and Technology, Institute of Computer Science and Telecommunications, Krasnoyarskiy Rabochiy, 31, Krasnoyarsk 660037, Russia

ARTICLE INFO

Article history:
Received 4 January 2019
Revised 6 May 2019
Accepted 11 June 2019
Available online 12 June 2019

Keywords:
Cyber-physical systems
Internet of Things
Speech recognition
Mobile robots

ABSTRACT

A cyber-physical system for a speech-controlled wheelchair has been proposed. It is based on cloud-harvesting principles, which result in significant improvement in command error rate (CER). The overall methodology and developed cloud-harvesting algorithm have been presented, discussed, and tested. The combination of IBM Watson and Google Cloud Speech APIs gave significantly better results than the use of solitary speech recognition APIs. The proposed approach shows the potential for and usability of speech-controlled wheelchairs, as well as in similar applications.

1. Introduction

The potential of cloud technologies is clearly evident, and the possibility to efficiently use cloud-based speech recognition is a new challenge. Several speech recognition services are currently active, including the Google Web Speech API [1] and IBM Watson Speech-to-Text Service [2]. Thus, there is a practical need to examine whether such services could be effectively "harvested" to obtain better results in terms of the command error rate (CER), which is considered the primary parameter in speech control and defines the accuracy rate of the interpretation of spoken commands [3–6]. In particular, these speech-recognition technologies are useful in the development of various cyber-physical systems that explore cloud services in order to achieve better performance and usability. One of the areas in which these technologies can be used is in the assistance of persons with disabilities.

In previous research, a speech-controlled cloud-based wheelchair platform for disabled persons was developed [7,8]. A single cloud service for speech recognition was used, giving good performance, but the result was still not at a level for the technology to be widely used. In the present research, we will take previous work further, considering the application of several speech-recognition cloud services to achieve better performance of speech-controlled wheelchairs. The concept of cloud harvesting will be introduced. The methodology can be applied to any cyber-physical system requesting this type of control mechanism. The theoretical considerations will be elaborated with details of practical implementation. The comparison and verification will be presented through the experimental testing in real conditions.

The research in speech-controlled wheelchairs is ongoing [9] in order to develop affordable and accurate solutions. Word error rate (WER) in speech-recognition systems is typically the main concern. Our paper contributes to this field of study by proposing a novel approach to engage several cloud-based speech-recognition systems in order to improve WER and, more importantly in our case, CER.

The presented system introduces a paradigm of developing speech-recognition systems wherein the cloud speech-recognition harvesting principle is applied. Necessary novel developments regarding the realization of this paradigm are usage of a Node. JS server and JavaScript to develop a complete control system, including PID algorithm and fuzzy logic for navigation control. By using Node. JS and JavaScript, we were able to easily combine two cloud speech-recognition APIs. Such an approach accelerates the development of a prototype with the potential for use in an end-product. The main advantage of the proposed approach compared to a stand-alone embedded system is its openness and tight inte-

* Corresponding author.
E-mail address: andrej.skraba@um.si (A. Škraba).

https://doi.org/10.1016/j.micpro.2019.06.006
0141-9331/

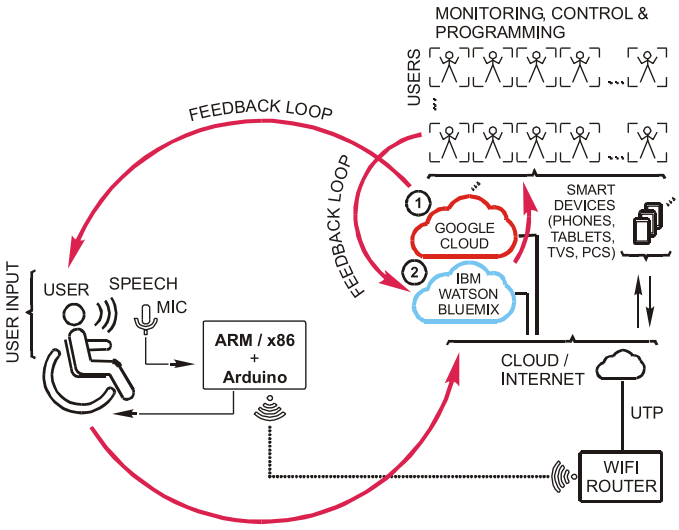

Fig. 1. User interaction with cyber-physical speech-controlled wheelchair applying cloud information systems in an intelligent learning loop.

gration with the internet and its users', with a constantly increasing vocabulary that could be corrected by users.

2. The wheelchair as a cyber-physical system

Cyber-physical system is defined as a "mechanism that is monitored or controlled by computer algorithms tightly integrated with the internet and its users." Fig. 1 shows the model of user interaction with a cyber-physical speech-controlled wheelchair applying cloud information systems in an intelligent learning loop. The speech input is provided by a microphone connected to an ARM [7,8] or x86 PC and then fed into the cloud-based voice-recognition services, Google (1) and IBM Watson (2).

The critical difference between the conventional standalone speech-controlled system and the one presented here is in the feedback loops. These loops are not present in the conventional recognition system; here, they provide better system accuracy and adaptability. Such an approach represents a new way of designing and operating cloud-based speech-recognition systems, increasingly used worldwide. The innovative combination of several cloud services for speech-recognition tasks is more useful, wherein the feedback loop between the user and the cloud system makes it possible to improve the speech-recognition function by correcting errors and producing suggestions; this enables the development of new cyber-physical systems with improved performance in terms of CER. The innovation of including several cloud speech-recognition systems together, which provides the possibility of developing new algorithms for speech control by harvesting several competing cloud platforms, builds on our previous research [7,8,10] that was based on a one-cloud service approach. In [7] and [8] the prototype solution was developed, first as the physical model [7] and later as the full-sized prototype [8]. Both solutions [7,8] were based on the single-cloud harvesting principle. In [10] we preliminarily analyzed the possibility of using several cloud-based speech-recognition platforms. A key addition to our previous research in the present paper is the detailed description of the developed system, thorough analysis of wheelchair navigation experiments with users and analytical as well as empirical confirmation of the appropriateness of the proposed approach.

3. System architecture

The wheelchair control system contains multiple elements, as shown in Fig. 2. The mobile part-the wheels together with the

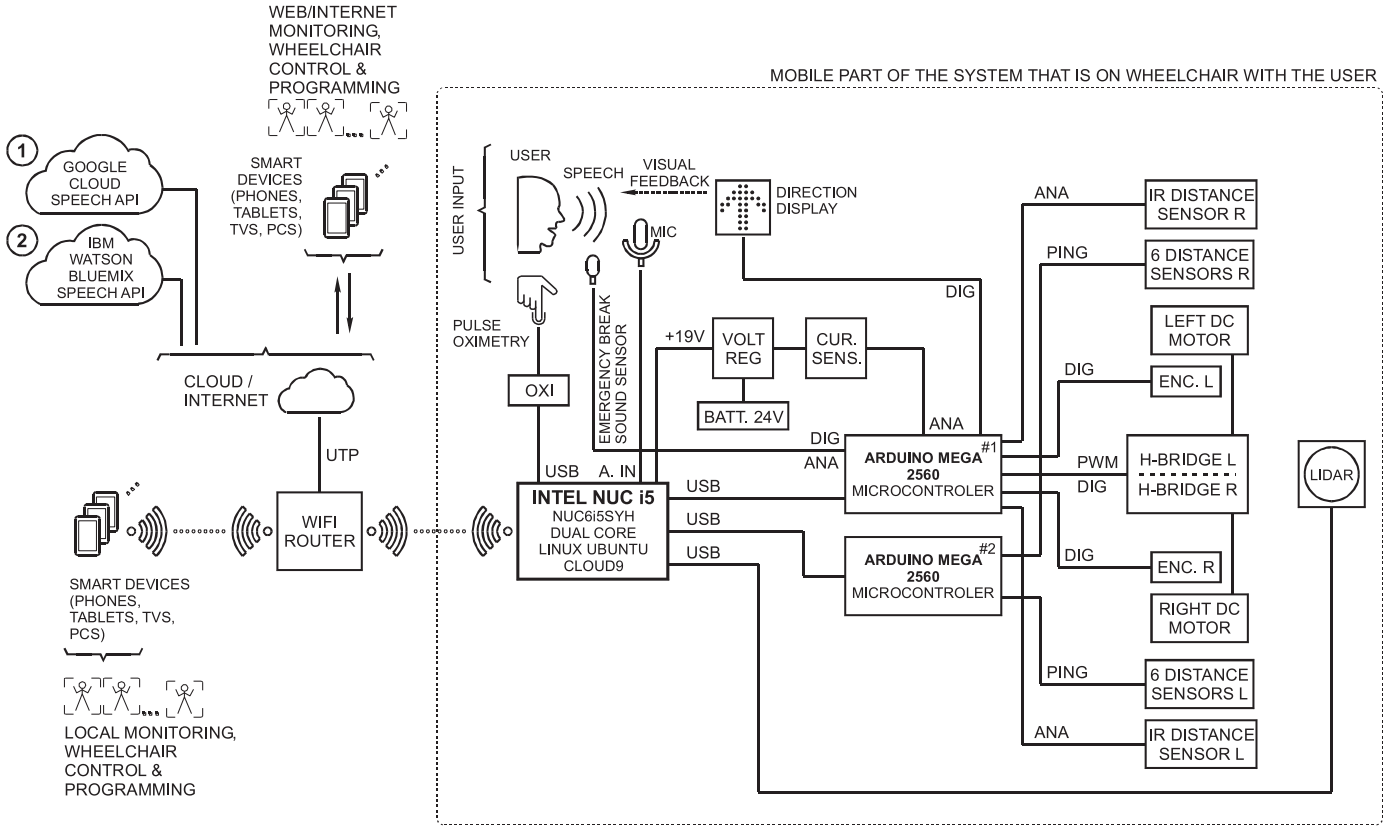

Fig. 2. Cyber-physical wheelchair schematics using Google and IBM Watson speech cloud APIs.

user-is on the right side of Fig. 2 (within the dashed border). The left side of Fig. 2 shows the "cloud" part of the system with the possibility of remote usage, control, and programming.

The novel part of the system is the parallel inclusion of two cloud speech APIs: (1) the Google cloud-based speech API and (2) the IBM Watson speech API. Both APIs will be explained, as they will be used in parallel to improve CER.

The core computational element of the system is small-form-factor Intel i5 NUC (NUC6i5SYH) personal computer with i5-6260 processor running at 1.8Ghz with 4Gb RAM and 40 Gb SSD, which is connected to two Arduino Mega 2560 microcontrollers, the front-facing LIDAR, microphone and noninvasive pulse oximetry (OXI) device, which is placed on the user's finger [11–13]. With the inclusion of the pulse oximeter, we wanted to show that biomedical signals could easily be acquired by a developed platform and used for monitoring the user's condition. The NUC runs the Linux Ubuntu Operating System, installed with local cloud9 IDE [14] in order to provide easier programming and control. The NUC is powered by a voltage regulator, providing +19 V from a 24 V battery pack. The same power is used to drive the wheelchair DC motors. The power consumption is monitored by the current sensor. The user provides the input via speech and gets visual feedback from the direction display, which allows better user interaction and control. One of the Arduino units (#1) reads the encoders, infrared sensors, current sensor, noise sensor (used for emergency break, triggered if a user produces a loud vocal sound or blows into it), drives the LED matrix of direction display, switches the solenoid brake, and finally controls two H-bridge drivers for the left and right motors; the other Arduino unit (#2) obtains the data from ultrasonic sensors and transfers it via USB to the computer. The same Arduino (#2) uses the uploaded firmware code to trigger the ultrasonic sensors and waits for the response, transforming the echo time to distance in centimeters. To avoid interference with the sensors, a unique triggering protocol is used so the sensors on different sides of the wheelchair are triggered. The first Arduino (#1) Mega 2560 contains a standard Firmata [15] library, with the only parameter changed being the PWM frequency, which was set to 30 kHz to eliminate the PWM noise in the DC motors. This Arduino (#1) was controlled by a Node. JS [16] server with a Johnny-Five [17] library. The Node. JS server on NUC with installed Firmata library [15] contained the entire control algorithm, including the data reading from the second Arduino. The position encoders for each wheel consist of 360 2 mm neodymium magnets in three rows and three Hall sensors. The encoders were used not only to acquire the rotation frequency, but also the rotation direction for each wheel.

In addition to the PID controller, which generates PWM values to drive the motors, the system also includes a fuzzy controller to read the sensor values and change the desired rotation frequency for each wheel in case of obstacles. The fuzzy controller includes 55 fuzzy variables, including 4 output variables to adjust the desired frequency (reduction coefficients), a single variable showing the number of the control command, 12 variables for ultrasonic sensors, 2 variables for infrared sensors, and 36 variables for LIDAR (reading 36 sectors). The rule set/base contains 264 rules, grouped into sets for every command. For example, when the given command is "move forward," the fuzzy controller only considers two ultrasonic sensors, two infrared sensors, and LIDAR data from 300 to 60°. To obtain the LIDAR readings, we divided the whole range (360°) into 36 overlapping sectors, each 20° wide. The LIDAR range is 3 m. To obtain the value from each sector, we used a procedure similar to the median calculation. In particular, we sorted the array of 20 values for each sector in ascending order and took the seventh-smallest value. This procedure was shown to be more robust than the usual median calculation, as LIDAR tends to return noisy points whose values are much larger than the actual distance to the object.

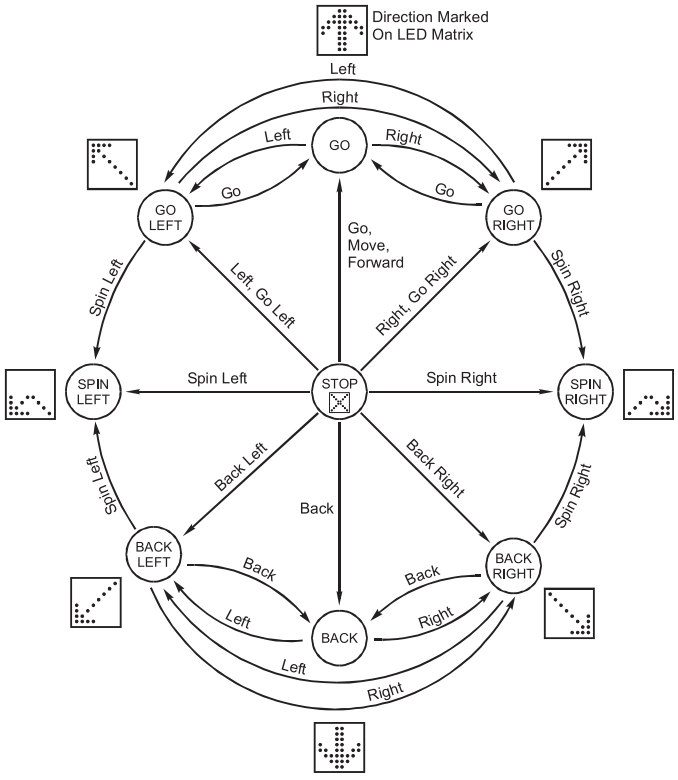

Fig. 3. Transition between states with direction indicators.

Fig. 3 shows the finite automaton transition between states in wheelchair speech control. From the central STOP position, the user controls the wheelchair in all directions. The transitions between the states depend on the previously issued command. The finite automaton defining the transition between wheelchair states is defined by a set of states, $Q = \{Q_S, Q_{GO}, Q_B, Q_{GL}, Q_{GR}, Q_{BL}, Q_{BR}, Q_{SL}, Q_{SR}\}$, and an alphabet set of input commands, $\Sigma = \{s, go, b, gl, gr, bl, br, sl, sr, l, r\}$, with initial state $q_0 = Q_S$.

"Left and "Go left" are two different command combinations for the wheelchair to go forward and turn left. Only the "Left" command, for example, could be used in two different cases: a) if the wheelchair is going forward the command "Left" will result in the wheelchair going forward-left, and b) if the wheelchair is going backward, the command "Left" will result in the wheelchair going backward-left. The command "Right" functions similarly.

Table 1 shows the transition between states. The last two columns show different transitions when commands "Left" or

Table 1
Transition table of cyber-physical wheelchair states.

Q	Σ										
	S	*go*	*b*	*gl*	*gr*	*bl*	*br*	*sl*	*sr*	*l*	*r*
S	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*GL*	*GR*
GO	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*GL*	*GR*
B	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*BL*	*BR*
GL	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*GL*	*GR*
GR	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*GL*	*GR*
BL	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*BL*	*BR*
BR	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*BL*	*BR*
SL	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*GL*	*GR*
SR	*S*	*GO*	*B*	*GL*	*GR*	*BL*	*BR*	*SL*	*SR*	*GL*	*GR*

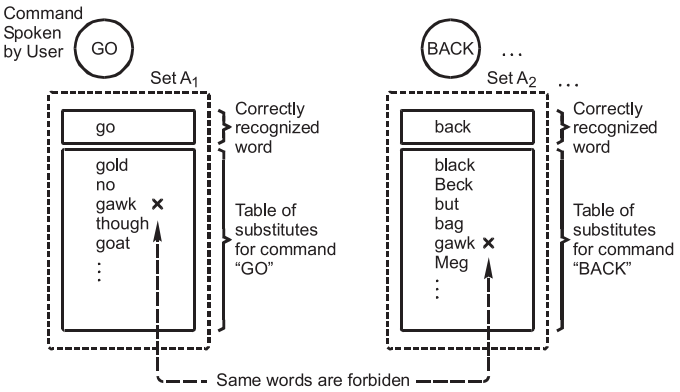

Fig. 4. Substitute tables with examples of forbidden-word duplication in different sets.

"Right" are used. The state to which the system moves depends on the starting position. For example, if one is moving forward in the state GO, the command "Left" will result in the state GL-that is, "Go left" which will move the user forward-left. If one is moving backward in state B, and the same command "Left" will be issued, the wheelchair will transit to the state BL, i.e. "Back Left," which is convenient when navigating.

The entire control system was developed in JavaScript/ECMAScript [18], from server-side code with Node. JS [16] (for interaction with microcontrollers) to client-side GUI, which was developed with plain HTML and JavaScript/ECMAScript [18]. This approach using only one programming language for development enabled us to efficiently and rapidly integrate two different cloud APIs within the system.

4. Methodology to improve the system's performance with parallel cloud harvesting

Recently, several different approaches have been developed [19–23] to apply and improve speech recognition in IoT systems. To improve the control of a speech-controlled cyber-physical wheelchair, we assumed only the predefined basic set of commands as $W = \{$go, stop, left, right, back, spin left, spin right, go left, go right, back left, back right$\}$. Therefore, we defined CER as the ratio between the number of false interpretations and the number of issued commands in total:

$$CER = \frac{C_f}{\mathrm{No}} \tag{1}$$

where C_f is the number of the falsely interpreted commands and N_o is the number of all ordered (spoken) commands.

One possible way to obtain a lower CER is to form the substitution table of final recognized words, as well as of interim results. The substitute table principle is shown in Fig. 4. For example, for the command "Go," several other words could be considered as proper results, such as {gold, no, ~~gawk~~, though, goat}. For the command "Back," the set of acceptable results would be {blak, Beck, but, bag, ~~gawk~~, meg}.

There should not be a common element in substitute tables represented as sets if we compare all possible pairs; this case is represented by the dashed line and *x* at the word "gawk," which is present in two substitute tables at the same time (Fig. 4). The sets should be formed experimentally from several users. When forming the vocabulary, the correctness of the recognition should be determined. The obtained sets should always be tested for consistency. Therefore, the n substitution tables (for n = wheelchair direction commands), which could be defined as sets $A_1,\ A_2, A_3, \ldots, A_n,$ should be pairwise disjoint:

$$A_i \cap A_j \equiv \emptyset; i \neq j \tag{2}$$

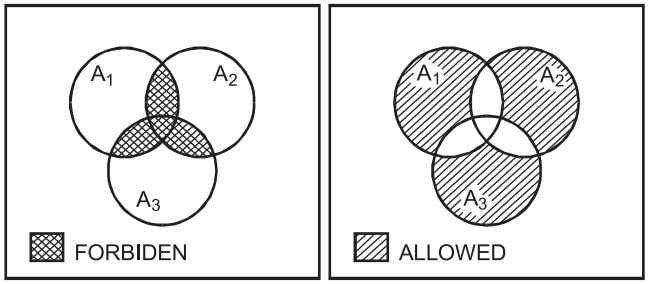

Fig. 5. Forbidden intersection for the case of three substitute tables sets, A_1, A_2 and A_3, (left) and allowed situation with area (right).

Algorithm 1 (Speech-to-command cloud harvesting) algorithm).

A1: **get** user speech input
A2: **seed** speech to the speech-recognition cloud field APIs
A3: **harvest** set of interim transcripts and timestamps from cloud field
A4: **if** $C_w > C_t$ add interim transcript to the Cloud_i command subset
A5: create unique union set of words for particular command from Cloud_i command subset
A6: **check** for pairwise disjoint condition for all unique union sets: $A_i \cap A_j \equiv \emptyset;\ i \neq j$
A7: **if** condition not met erase word pair
A8: **order** checked unique union set by interim transcript timestamp
A9: **execute** command with lowest timestamp

Table 2
Possible output combinations for two cloud APIs.

Google API	IBM Watson API	Combined output
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 5 illustrates the idea of a pairwise disjoint set for the case of three substitute tables sets: A_1, A_2 and A_3. On the left side of Fig. 5, the forbidden intersection is shown; on the right side, the allowed area is represented. In the case of more sets, this idea would be expanded in a similar manner, according to Eq. (2).

According to the type of data provided by several speech APIs and considering substitution tables and timestamps of provided results, Algorithm 1 has been developed (see above Algorithm 1 pseudo-coding). C_w and C_t represent the confidence threshold set by the user. Interim transcript timestamps determine which cloud service will be used in a particular case. Even if no new word is put into the vocabulary, timestamps are used with the sliding average window, which determines the order in a unique union set for a particular command. The algorithm needs the supervision in the initial phase. The automatic generation could improve its performance [24–30].

To avoid errors in recognition, certain thresholds should be included in the confidence value. A particular word would then be added to the union only if the confidence value is greater than he critical confidence threshold $C_w > C_t$.

Table 2 shows possible output combinations for the two cloud APIs, in our case Google and IBM Watson. In the first two columns, "0″ represents an error when recognizing a command and "1″ represents a correctly recognized command. In the last column, the output is represented, where "0″ represents an error in overall recognition and "1″ correct overall recognition.

To execute the correct output command, only one of the APIs' output should be correct, meaning that the situation would occur when the Google API gives correct output and the IBM Watson API false output, and vice versa. It would be expressed as functional

dependence, where a and b represent the API outputs:

$$f_2 = (a \wedge \neg b) \vee (\neg a \wedge b) \quad (3)$$

i.e. XOR operator. Referring to Table 2, this means that lines #2 and #3 should be present. If both the outputs are wrong, there are two options: a) If the spoken word is not within the table of substitutes, no command is executed, and b) if the output is incorrect (for example, the user says "Back" but the system understands "Left"), the related movement is falsely enacted (so the wheelchair goes left instead of back), but the user can then use the emergency break by blowing in a second microphone that is used only for the emergency break. Afterward, the system waits for the new command to be issued by the user.

If the two outputs are both correct, the system acts upon the first correctly recognized command. Here the cloud recognition with the lowest latency has priority.

The following situation could also arise: if the user says "Left" and one of the cloud speech recognition systems understands it as "Left" and the other as "Right," the last-recognized would be executed, according to the timestamp of the recognition, meaning the false command would be executed. In this case, as before, the emergency break could be used. In general, the last command according to the timestamp is executed.

In the case of three APIs, the function takes the following form:

$$f_3 = (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) \quad (4)$$

In the case of more APIs, we would have a similar expression. The application of more APIs in parallel could also be justified if the conditions stated by Eq. (3) or Eq. (4) could not be fulfilled. In this case, the reliability of command execution and the system latency would be improved.

For example, even if we have three cloud APIs with perfect speech recognition, each producing CER=0, the latency will be different. In addition to the increased accuracy due to the parallel system structure, we will be able to check whether the speech recognition results of three branches match.

According to the presented example of the cloud result combinations, the CER when harvesting multiple cloud platforms (CER_m) is defined as:

$$CER_m = \prod_{i=1}^{n} CER_i \quad (5)$$

where CER_i is the command error rate for the i th cloud platform and n is the number of harvested platforms, all under condition $CER_i > 0$. This indicates the rationale of using the cloud-harvesting algorithms. Ideally, $CER = 0$ is possible via the use of a diversity of cloud services.

5. Experiments and tests

Experiments with two groups (16 students from the University of Maribor and 4 patients in a clinical environment) were performed. Members of the student group were 7 females and 9 males, aged from 20 to 55 years, while the patient group had 1 female and 3 male participants, aged from 35 to 55. The experiments with the student groups were preliminarily performed to test the platform's functionality. There were no native English-speakers among the participants.

5.1. Task description

The task was divided into two phases. In the first phase, each participant was asked to issue 12 direction commands 10 times, 120 commands altogether. The CER was simultaneously measured. This was done without moving the wheelchair over the test track, as an introductory phase of user-speech-recognition system interaction. In the second phase, the participants performed a test drive on a test track, as shown in Fig. 6. Each participant performed three test drives in a row. The test track was marked with five cones, determining the path to be followed. The maximum distance between cones was 5 m. A participant started from behind the first cone, marked by the red dashed rectangle, and then moved between two cones that were 1.2 m apart.

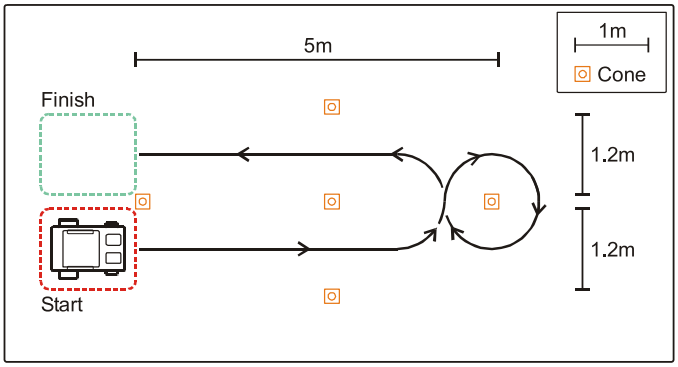

Fig. 6. Test track set-up with five cones and the path that was followed by the participants.

The participant should make a bow turn around the right-most cone and return on the right side between the upper two middle cones to the finish line. The path includes left and right turns, as well as driving between two obstacles. The experimental set-up is compact and suitable for maneuverability tests.

The experimental set-up — the polygon with test track — was used with the group of students ($N = 16$) and patients ($N = 4$) at the University Rehabilitation Institute, Soča, Republic of Slovenia. The test track proved to be easily set up, both in experimental and clinical environments. Here it is important that the test track is not too big and that a proper variety of commands is used, for example, turning left as well as right when driving on the test track. Moreover, the tasks should not be too time-consuming or burdensome for the test subjects.

5.2. Measurements and field testing

During the experiments, the following data was acquired:

(1) CER
(2) Time to perform the test drive
(3) Energy consumption of the wheelchair for test drive
(4) Number of issued commands
(5) Latency

The experiments were conducted from 1 July 2016 to 31 July 2016 in two geolocations (46.251119, 14.349448) and (46.066700, 14.524123).

Fig. 7 shows the patients testing the wheelchair at University Rehabilitation Institute, Soča, Republic of Slovenia. At No. 1, the microphone is marked. Its function is to input the spoken commands and includes the sound sensor for the emergency brake. No. 2 represents the LED matrix display, showing the direction recognized by the wheelchair and acting as important visual feedback to the user. No. 3 presents one of the rear right-side ultrasonic distance sensors. No. 4 illustrates the oximetry sensor that provides the heart-rate monitoring and SpO_2 (oxygen saturation data) of the patient. This sensor is essential in monitoring the health status of the patient and adds elements of telemedicine to the system [31–32]. No. 5 marks the cones on the test track. No. 6 is the middle right-side ultrasonic sensor, while No. 7 is the front ultrasonic sensor. No. 8 is the infrared left front ultrasonic sensor, and No. 9 is the LIDAR mounting frame. No. 10 is the NUC computer, and No. 11 is

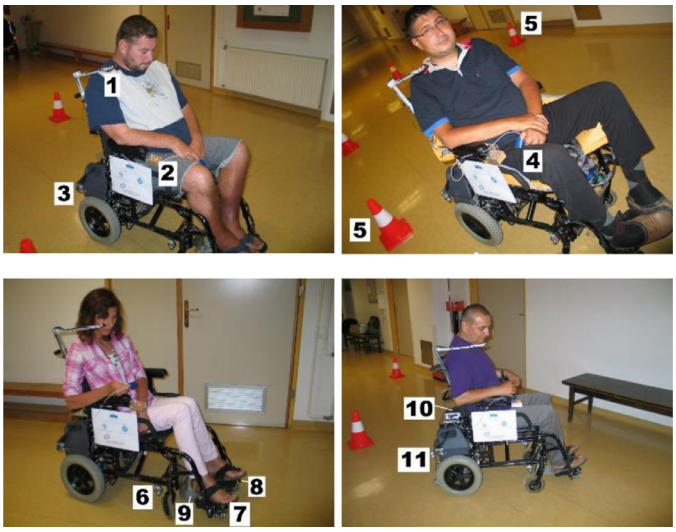

Fig. 7. Patients testing the cyber-physical speech-controlled wheelchair on the test track with enumerated locations of main system components.

Table 3
Average number of issued commands needed to complete a drive on the test track, time needed, and energy consumption in 1st, 2nd, and 3rd attempts.

Run	N	AVG [#c.]	SD	N	AVG [min]	SD	N	AVG [W·s]	SD
1st	14	33.0	14.12	13	3.70	0.93	14	4875.79	1577.27
2nd	14	21.7	9.16	13	3.03	0.79	14	3742.50	1999.96
3rd	14	19.6	8.40	13	2.77	0.74	14	3606.64	1627.24

the rear right-side sensor. Note that there is no joystick control on the wheelchair.

6. Results

The experiments were performed with 20 subjects issuing the commands from the set $W=\{$go, stop, left, right, back, spin left, spin right, go left, go right, back left, back right, break$\}$. Each subject issued each command 10 times: 120 commands were issued per participant, and 2400 commands altogether were issued. This was done without the wheelchair actually moving and was only done for initial testing of the speech-recognition accuracy. The obtained CER values for different subjects are shown in Fig. 8 (left). The CER values for different subjects ranged from 0.533 to 0.025. This means that a particular person's speech recognition had at worst 53.3% wrong interpretations, while at best, only 2.5% wrong interpretations. Experimentally, we determined the distribution for the set of 12 commands specified by W (Fig. 8, right). The CER values for different commands ranged from 0.240 to 0.060. The bars in Fig. 8 are ordered according to the CER values from largest to smallest. Distribution is not symmetrical, indicating that the control commands could also be selected according to the lowest CER; for example, the command "Stop" has a lower CER than "Brake". Therefore, we declared an additional "Stop" word, its "synonym," and included it in the synonym table to provide better CER in the case of the most critical command (to stop the wheelchair). In general, this means that we should select the word with the lowest possible CER to issue this particular command. It is notable that the "Stop" command, as one of the most critical commands, has the lowest CER value. This is likely connected to the features of human language.

Table 3 shows: (a) the average number of issued commands to complete a test drive on the test track, (b) the time needed, and (c) energy consumption in the first, second, and third attempts. The number of issued commands by the user is not predetermined. It is dependent on the skill of the user in navigating the wheelchair according to the path in Fig. 6. The theoretical minimal number of issued commands to navigate according to the path in Fig. 6 would be five ("Go," "Left," "Right," "Left," and "Stop"). The number of actually issued commands by users is higher because of the fact that participants were using this system for the first time and needed to learn how to navigate it. There were also some additional commands issued because some commands were not interpreted correctly. The number of gathered data was $N=14$ for the number of issued commands and the average energy consumption, and $N=13$ for average time. The number is not 20, as the number of participants, as some measurements were not recorded in the early phase of system development.

For example, to determine the average number of issued commands needed to complete the test drive on the test track in Fig. 6, we counted the number of issued commands to complete the test track for $N=14$ participants and determined the average and standard deviation. The results indicated that there was a statistically significant difference between the numbers of issued commands in the case of first and third attempts. This number significantly decreased (t-test, $N=14$, $p=0.05$, M1 = 33.00, SD1 = 14.12, M3 = 19.6, SD3 = 8.40, df = 21, t-stat = 3.05807) from 33 command to 19.6 commands. This means that, for example, the participant had to issue an average of 33 spoken commands to complete the test track task upon the first attempt and only 19.6 commands at the final (third) attempt. The time needed to complete the test track also significantly decreased ($p=0.05$, t-test, $N=13$, M1 = 3.70, SD1 = 0.93, M3 = 2.77, SD3 = 0.74, df = 23, t-stat = 2.80391) from the first to third attempts. The time needed to complete the test track decreased from 3.7 min to 2.77 min, which means the participants on average completed the test track almost one minute earlier in the final (third) attempt. A similar situation was observed in terms of energy consumption ($p=0.05$, t-test $N=14$, M1 = 4875.79, SD3 = 1577.27, M3 = 3606.64, SD3 = 1627.24, df = 26, t-stat = 2.06641). Here, the energy consumption decreased from 4875.79 W•s (Watt seconds) down to 3606.64 W•s per test drive. The results indicated the learning curve in the three repetitions that were performed.

In the first attempt, the number of commands is the highest since users have used the system for the first time and need to learn how to interact with the systems by speech in order to navigate. In the second attempt the users have improved their performance. In third attempt the number of issued commands is the lowest since the users have better learned how to use the system. With more attempts it would be possible to use only five commands to complete the test track. With brief training, significantly better results could be obtained in terms of time, energy consumption, and the efficiency of spoken commands.

Fig. 9 shows the learning curve in the three attempts to complete the test track. The gradient is higher between the first and the second test drives. Nevertheless, the third attempt also provided an improvement in all three parameters. Thus, training is crucial for wheelchair control by the user. With proper training, a minimal number of commands would need to be issued to follow the desired path.

After they had completed the three test drives, we asked the subjects about the quality of the system's speech recognition, the execution of commands, and navigation task difficulties. A negative correlation was indicated between the subjective perception of how well the system recognized speech and CER for 120 preliminary issued commands (CER120) ($r=-0.27$). This means that measured CER for a particular subject and his/her subjective perception of how well the system recognized speech were correlated. A similar negative correlation was indicated between the perception of how well the wheelchair executed a command and CER120 ($r=-0.28$). A negative correlation was also indicated between the

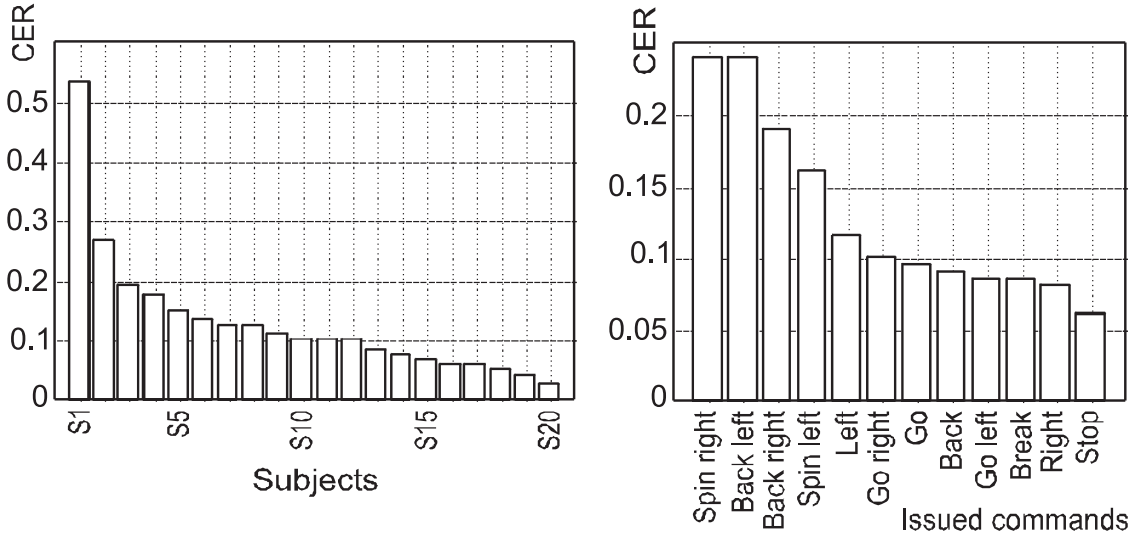

Fig. 8. CER for twelve commands for controlling wheelchair, ordered from highest to lowest (with lowest indicating better results).

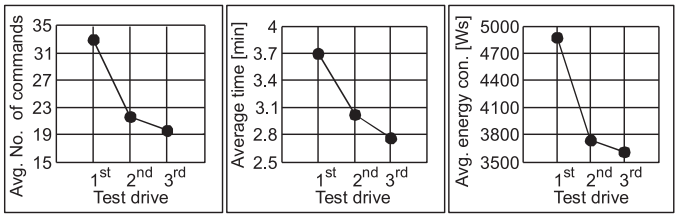

Fig. 9. Average number of issued commands in 1st, 2nd, and 3rd test drives (left), average time needed (center), and average energy consumption (right).

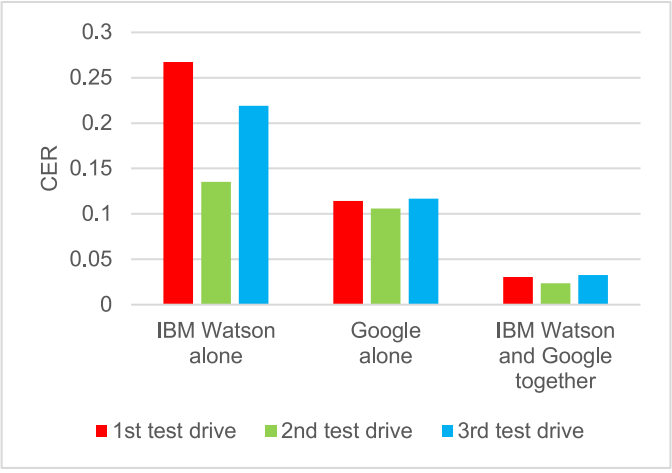

Fig. 10. CER improvement when IBM Watson and Google APIs are combined (lower is better).

perception of the navigation task difficulty and power consumption (r=−0.25). If the participant perceived the navigational task to be easy, the power consumption for driving through the test track was lower. This indicates that the participants understood the questions from the survey and generally had a proper notion of the system accuracy, as well as self-perception of the test-drive task execution.

We also gathered latency measurements for all three test runs. Average latency for the Google API was 1.037 s with SD = 0.262, while IBM Watson API latency was 3.552 s with SD = 4.208 ([ANOVA F(1, 52) = 11.73041, p = 0.001, N1 = N2 = 9]). Again, the number was lower than 20 due to prototype development and some measurements not being recorded in early phases of the study. The latency of the two cloud speech-recognition systems significantly differed. The latency measurement set-up is described in detail by Škraba et al. [8].

Table 4 shows the number of issued commands in the first, second, and third attempts, as well as the CER of Google's API (CERg), the CER of Watson's API (CERw), the combined CER (CERgw), and the improvement of Google API and Watson API in percentages, indicated in the last two columns. Altogether, 1016 commands were issued by 20 participants. When the combination of the two cloud-based speech-recognition systems was used, a statistically significant improvement in CER of both cloud speech-recognition systems (Google [ANOVA F(1, 4) = 369.93, p = 4.31E-05] and IBM Watson [ANOVA F(1, 4) = 21.23, p = 0.00997]) was obtained.

Fig. 10 shows the improvement in CER over three performed test drives. The combination of IBM Watson and Google APIs provided significantly better performance than a solitary speech-recognition API application.

Results indicate that it is possible to develop a speech-recognition system that would surpass the accuracy of the IBM Watson and Google speech-recognition systems. This indicates a potential future realization of similar systems based on several concurrent cloud services.

Table 4
CER of Google's API (CERg), IBM Watson's API (CERw), combined CER (CERgw), and improvement of Google API and Watson API in percent (last two columns).

Run	No. of issued comm.	N	CER of Google API alone (CERg)	CER of IBM Watson API alone (CERw)	CER of Google & IBM Watson combined (CERgw)	Google CER improvement	Watson CER improvement
1st	490	20	0.11	0.27	0.03	8%	24%
2nd	340	20	0.11	0.14	0.02	8%	11%
3rd	274	20	0.12	0.22	0.03	8%	19%

7. Conclusion

An approach for a speech-controlled cyber-physical wheelchair for disabled persons has been developed. It is based on cloud harvesting and, as such, presents a new paradigm of complex cyber-physical systems development. The main part of the system resides in the cloud with the primary characteristic of being open and connecting the internet with the users. Within this approach, we have tested two different cloud APIs and noted the differences in terms of speech-recognition accuracy. A proposed methodology based on a combination of several cloud speech-recognition APIs has been successfully tested by different users in a clinical environment using a specially designed test track. The quantitative tests demonstrated significant improvements of CER that fully correspond to the theoretical model. In the studied case, by the proposed methodology, a combination of IBM Watson and Google APIs provided significantly better performance than a solitary speech-recognition API, with important impact for further development of such systems. It was also shown that there are significant differences between users in CER values and commands (for example, the “Stop” command had one of the lowest CERs).

The diversity of the cloud APIs is a mandatory condition for the effectiveness of the proposed methodology. Therefore, it is of benefit that various providers of cloud APIs are developing independent solutions, including IBM’s Watson, Google Cloud Speech API, Amazon’s Alexa, and Apple’s Siri.

Additionally, the difference between the cloud services and their competitiveness enable participants to use the demonstrated methodology of parallel cloud harvesting to develop systems with superior technical performance. The present research and trial represent development in this direction.

Conflict of Interest

There is no conflict of interest.

Acknowledgments

This work was supported by the Slovenian Research Agency (ARRS) within SI-MNE bilateral project “Development of Speech-Controlled Wheelchair for Disabled Persons as Cyber-Physical System,” Proj. No.: BI-ME/16-17-022, SI-RF bilateral project “Efficient Control of Cyber-Physical Systems & Internet of Things by the Application of Evolutionary and Biologically Inspired Algorithms,” Proj. No.: BI-RU/16-18-040, SI-RF bilateral project “Numerical-analytical examination of biologically inspired algorithms for control of Cyber-physical Systems” Proj. No.: BI-RU/19-20-034, SI-ME bilateral project BI-ME/18-20-009 “Development of a cyber-physical system for stress monitoring in endangered individuals and groups”, Russian Federation Presidential Scholarship No. 16-in-689 and research program “Decision Support Systems in Electronic Commerce,” program No.: UNI-MB-0586-P5-0018. Authors would like to express gratitude to Ms. Nataša Bartol, Mr. Boštjan Klenovšek, Mr. Klemen Kramžar and Mr. Matjaž Bartol for their helpfulness and good will to participate in the clinical testing of the new technology.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.micpro.2019.06.006.

References

[1] Google.com, Web Speech API Demonstration. 2019 (accessed 11 May 2019) https://www.google.com/chrome/demos/speech.html.
[2] IBM Watson, Speech to text demo. 2019 (accessed: 11 May 2019) https://speech-to-text-demo.ng.bluemix.net/.
[3] Y. Oualil, D. Klakow, G. Szaszák, A. Srinivasamurthy, H. Helmke, P. Motlicek, A context-aware speech recognition and understanding system for air traffic control domain, in: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Okinawa, 2017, pp. 404–408.
[4] K. Spiel, S. Bertel, M. Heron, Navigation and immersion of blind players in text-based games, Comput. Games J. 3 (2) (2014) 132–154.
[5] R. Zmarzlak, Missile tracking satellites explore new missions, MilsatMagazine (2013) 18–20 June 2013.
[6] Feeder Level Microgrid Demonstration in Los Alamos, Japan-U.S. New Mexico Smart Grid Collaborative Demonstration Project, NEC Corporation, Feb. 2015 A-6-1 – A-6-4.
[7] A. Škraba, A. Kološvari, D. Kofjač, R. Stojanović, Prototype of speech controlled cloud based wheelchair platform for disabled persons, in: 3rd Mediterranean Conference on Embedded Computing (MECO), 2014, Budva, 2014, pp. 162–165.
[8] A. Škraba, R. Stojanović, A. Zupan, A. Kološvari, D. Kofjač, Speech-controlled cloud-based wheelchair platform for disabled persons, Microprocess. Microsyst. 39 (8) (2015) 819–828.
[9] S. Teller, Intelligent wheelchair project at MIT. 2019 (accessed 11 May 2019) http://rvsn.csail.mit.edu/wheelchair/.
[10] A. Škraba, V. Stanovov, E. Semenkin, A. Kološvari, D. Kofjač, Development of algorithm for combination of cloud services for speech control of cyber-physical systems, Int. J. Inf. Technol. Secur. 10 (1) (2018) 73–82 Iss.
[11] J. Pan, J. McElhannon, Future edge cloud and edge computing for internet of things applications, IEEE Internet Things J. 5 (1) (2018) 439–449.
[12] M.S. Mahmud, H. Wang, A.M. Esfar-E-Alam, H. Fang, A wireless health monitoring system using mobile phone accessories, IEEE Internet Things J. 4 (6) (2017) 2009–2018.
[13] I. Bisio, A. Delfino, F. Lavagetto, A. Sciarrone, Enabling IoT for in-home rehabilitation: accelerometer signals classification methods for activity and movement recognition, IEEE Internet Things J. 4 (1) (2017) 135–146.
[14] Amazon Web Services, Cloud9, 2019 (accessed 11 May 2019) https://github.com/c9/core.
[15] The Firmata.js Authors, Firmata NPM. https://www.npmjs.com/package/firmata, (accessed 11 May 2019).
[16] Node.js Foundation, Node.js. 2019 (accessed 11 May 2019) https://nodejs.org.
[17] The Johnny-Five Contributors, Johnny-Five: tThe JavaScript Robotics & IoT Platform. 2019. (accessed 11 May 2019) http://johnny-five.io/.
[18] Ecma-international.org, Standard ECMA-262. 2019 (accessed 11 May 2019) https://www.ecma-international.org/publications/standards/Ecma-262.htm.
[19] M.S. Hossain, G. Muhammad, Emotion-aware connected healthcare big data towards 5G, IEEE Internet Things J. 5 (August(4)) (2018) 2399–2406.
[20] A. Buzo, H. Cucu, C. Burileanu, M. Pasca, V. Popescu, Word error rate improvement and complexity reduction in Automatic Speech Recognition by analyzing acoustic model uncertainty and confusion, in: 6th Conference on Speech Technology and Human-Computer Dialogue (SpeD), Brasov, 2011, pp. 1–8.
[21] M. Carpuat, D. Wu, Improving statistical machine translation using word sense disambiguation, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, 2007, pp. 61–72.
[22] L. Mangu, E. Brill, A. Stolcke, Finding consensus in speech recognition: word error minimization and other applications of confusion networks, Comput. Speech Lang. 14 (4) (2000) 373–400.
[23] K. Zechner, A. Waibel, Minimizing word error rate in textual summaries of spoken language, in: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL 2000), Stroudsburg, PA, USA, Association for Computational Linguistics, 2000, pp. 186–193.
[24] J. Drexler, J. Glass, Analysis of audio-visual features for unsupervised speech recognition, in: GLU 2017 International Workshop on Grounding Language Understanding, Stockholm, Sweden, 2017, pp. 57–61.
[25] K. Gupta, D. Gupta, An analysis on LPC, RASTA and MFCC techniques in Automatic Speech recognition system, in: 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016, pp. 493–497.
[26] K. Ochi, N. Ono, S. Miyabe, S. Makino, Multi-talker speech recognition based on blind source separation with ad hoc microphone array using smartphones and cloud storage, in: Proc. INTERSPEECH 2016, 2016, pp. 3369–3373.
[27] R. Yazdani, A. Segura, J.M. Arnau, A. Gonzalez, An ultra low-power hardware accel-erator for automatic speech recognition, in: 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Taipei, 2016, pp. 1–12.
[28] M.R. Talib, M.K. Hanif, Z. Nabi, M.U. Sarwar, N. Ayub, Text mining of judicial system’s corpora via clause elements, Int. J. Inf. Technol. Secur. 3 (2017) 31–42.
[29] A.S. Sharma, R. Bhalley, ASR — A real-time speech recognition on portable de-vices, in: 2016 2nd International Conference on Advances in Computing, (Fall), Bareilly, Communication, & Automation (ICACCA), 2016, pp. 1–4.
[30] K. Nakadai, T. Mizumoto, K. Nakamura, Robot-audition-based human-machine interface for a car, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 6129–6136.
[31] T.N. Gia, V.K. Sarker, I. Tcarenko, A.M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, Energy efficient wearable sensor node for IoT-based fall detection systems", Microprocess. Microsyst. 56 (2018) 34-4, doi:10.1016/j.micpro.2017.10.014.
[32] D. Kofjač, R. Stojanović, A. Kološvari, A. Škraba, Designing a low-cost real-time group heart rate monitoring system, Microprocess. Microsyst. 63 (2018) 75–84 75-84, doi:10.1016/j.micpro.2018.08.010.

Andrej Kološvari received his B.Sc. and M.Sc. in the field of Organizational Sciences – Informatics from the University of Maribor in 1986 and 2005, respectively. Mr. Kološvari has extensive industrial experience in the field of measurement, monitoring and control from Iskra Kibernetika, Inc. He was a developer of hardware and software systems at the Institute for Teleinformatics in Kranj. He is currently a Ph.D. student at the University of Maribor, Faculty of Organizational Sciences working on his thesis in the field of cyber-physical systems and Internet of Things in rehabilitation processes.

Radovan Stojanović obtained his Dipl. Ing. degree from University of Montenegro and Ph.D. from the University of Patras, Greece in electrical engineering and computer engineering. Currently, he is a full professor at the University of Montenegro, where he leads the Applied Electronics Centre. He has been a leader of numerous EU, NATO, bilateral and national projects and an author/coauthor of more than 200 publications. Prof. Stojanović is a member of the Board of Montenegrin Academy of Science for Natural and Technical Sciences, Think Tank Team of the Ministry of Science of Montenegro, Montenegrin representative in H2020-ICT Committee as well as a President of the Montenegrin Association for New Technologies (MANT). He is the founder and chairman of the Mediterranean conference on embedded systems (MECO) and Montenegrin Director of EuroMicro. In Montenegro, he established the Centre for Applied Electronics, the Centre for Biomedical Engineering (BioEMIS), the Centre for the Simulation of Disasters (GEPSUS) and the Mediterranean Excellence in Embedded Computing (MECOnet).

Anton Zupan MD, Ph.D., graduated in Medicine at the University of Ljubljana (Slovenia) in 1980. He received MSc degree in 1988 and Ph.D. degree in 1992 both from the University of Ljubljana in the field of medicine. He is a physician, specialist in physical and rehabilitation medicine and specialist in pediatrics. He works at the University Rehabilitation Institute in Ljubljana as head of the Rehabilitation Engineering Department and head of the unit for the rehabilitation of patients with neuromuscular diseases. Since 1994, he has been an Associate Professor of physical and rehabilitation medicine in the Faculty of Medicine of the University of Ljubljana. Dr. Zupan is an experienced clinician and scientist. He has led and participated in several national and international projects as well as conducting several clinical trials investigating rehabilitation programs.

Eugene Semenkin received his Master's in Applied Mathematics from Kemerovo State University (Kemerovo, USSR) in 1982, his Ph.D. in Computer Science from Leningrad State University (Leningrad, USSR) in 1989 and his DSc in Engineering and Habilitation from the Siberian State Aerospace University (Krasnoyarsk, Russia) in 1997. Since 1997, he has been a professor of systems analysis at the Institute of Computer Science and Telecommunications of the Siberian State Aerospace University (SibSAU). His areas of research include the modelling and optimization of complex systems, computational intelligence and data mining. Prof. Semenkin has been awarded the Tsiolkovsky Badge of Honor by the Russian Federal Space Agency and the Reshetnev medal by the Russian Federation of Cosmonautics. He is a member of IEEE Computational Intelligence Society and corresponding member of Russian Engineering Academy. Prof. Semenkin is a co-founder and chair of the International Workshop on Mathematical Models and their Applications. He established the Siberian Institute of Applied System Analysis at SibSAU.

Vladimir Stanovov received his B.S. and M.S. degrees in system analysis and control from Reshetnev Siberian State Aerospace University (SibSAU), Krasnoyarsk, Russia, in 2012 and 2014, respectively. He holds a Ph.D. in system analysis, control and information processing from SIbSAU (2016). His research interests include genetic fuzzy systems, self-configured evolutionary algorithms and machine learning. He is currently a senior research fellow at the Siberian State University of Science and Technology (former SibSAU). Mr. Stanovov received the Best Student Paper Award from the 4th International Congress on Advanced Applied Informatics in 2015 as well as Krasnoyarsk Governor Price for outstanding achievements in science (2016). In the 2015/2016 academic year, he was awarded the Ph.D. Scholarship of the President of the Russian Federation.

Davorin Kofjač received his BSc and Ph.D. from the University of Maribor in the field of Management of Information Systems in 2003 and 2007, respectively. He has published several papers in international conferences and journals and has been involved in many national research projects. Currently, he is working as a researcher at the same university in the Cybernetics & Decision Support Systems Laboratory. His research interests include modeling and simulation, artificial intelligence and operational research. Dr. Kofjač has received the Best Paper award at the Conference on Computing Anticipatory Systems CASYS 2005 and Best Paper award on the conference Computer Supported Education CSEDU in 2010. He was a recipient of Trimo Research Award for his doctoral dissertation in 2008. He was recognized as a "Fellow of the IIAS" in 2017.

Andrej Škraba obtained his BSc, MSc and Ph.D. in the field of Organizational Sciences – Informatics from the University of Maribor in 1995, 1998, and 2000, respectively. He works as a full professor and researcher in the Cybernetics & Decision Support Systems Laboratory at the University of Maribor, Faculty of Organizational Sciences. His research interests cover systems theory, modeling and simulation, cyber-physical systems, the internet of things and decision processes. Prof. Škraba has received a Bronze Medal of University of Maribor, for successful research and pedagogical work in the field of Systems Modeling and Simulation in 2003. He is a member of the System Dynamics Society (SDS) and the Slovenian Society for Simulation and Modelling (SLOSIM).