**N E W S**
OF THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
**SERIES OF GEOLOGY AND TECHNICAL SCIENCES**
ISSN 2224-5278
Volume 6, Number 438 (2019), 215 – 222 **https://doi.org/10.32014/2019.2518-170X.173**

UDC 004.9:519
IRSTI 20.53.00

**Maksat Kalimoldayev, Maxat Akhmetzhanov, Murat Kunelbayev, Talgat Sundetov**

Institute Information and Computational Technologies CS MES RK, Almaty, Kazakhstan.
E-mail: maks714@mail.ru, murat7508@yandex.ru

# INFORMATION SYSTEMS OF INTEGRATED MACHINE LEARNING MODULES ON THE EXAMPLE OF A VERBAL ROBOT

**Abstract.** In this work, information systems of integrated machine learning modules have been performed using the verbal robot as an example. Hardware components have been developed, logical components that have been assembled and /or developed to implement an automated verbal robot system, module tracking has been also performed that can track human faces in real time through the OpenCV library and automated services on Jetson TX1 SoC for maneuvering a mobile robot chassis.

**Key words:** information systems, integrated modules, machine learning, verbal robot.

**Introduction.** An automated platform to study the role of the verbal robot is shown in [1]. In [1], a humanoid robot was developed by French sculptor Gael Langevin using the InMoov platform as part of a public project initiated in 2012.

The main software architecture is the implementation of Automatic-Deliberative (AD) Architecture [2]. The main component of the AD architecture is skill [3]. Skill is a minimal module that allows the robot to perform an action, such as moving through the environment, reading products from a laser sensor or communicating with a person.

In essence, skill is a process that conducts computational operations and shares the results of these operations with other skills. For example, imagine a skill that is responsible for detecting obstacles using laser readings. In this case, the main skill operations: reading laser data, deciding whether there is an obstacle or not and making this information available to other skills. The partitioning mechanisms used by skills are events (a communication mechanism that follows the publisher/subscriber paradigm described by Gamma et al. in [4]), or a shared memory system. ROS [4] is an open source, meta-operating system for robots.

It provides services similar to those provided by the operating system (OS), including hardware abstraction, low-level device control, implementation of commonly used functionalities, interprocess communication, and package management. In addition, it also provides tools and libraries for obtaining, building, writing, and managing code in a multi-computer environment.

The main concept of ROS that applies to this article is nodes and themes. The first is the minimal ROS architecture unit structure. These are processes that perform the calculation. Essentially every skill is implemented as a ROS node. The latter, topics, are a communication system that allows the exchange of information between nodes. They are, in fact, the implementation of Announcement events.

**Method of research.**

***Hardware Components.*** The central processor consists of the Arduino Mega ADK https://store.arduino.cc/arduinomega-adk-rev3) board. This microcontroller is responsible for collecting commands from various software modules running on the PC discuss later and transmitting them to the servos. Two cameras were placed in the robot's eyes to reproduce the vision system. Two speakers were used for sound reproduction; they were attached to the amplifier Board and placed in the robot's ears. An external microphone was used for the auditory system to limit the effects of noise produced by the servos on the perceived audio. The robot assembly includes 28 servos with different speeds and spins distributed across the body and providing a total of 28 degrees of freedom (DOFs). All arm servos have been modified to allow robot joints to perform voluntary rotations not allowed in the original design. The arm chain (omoplate, shoulder, biceps, elbow, forearm, and wrist) consists of six articulated inverse kinematics (IK) of use.

***Software components.*** The logical components that were assembled and/or developed to implement the automated verbal robot system are illustrated in figure 47. In the created architecture, which was inherited from the InMoov project, the lowest layer is represented by physical robot sensors and head drives.

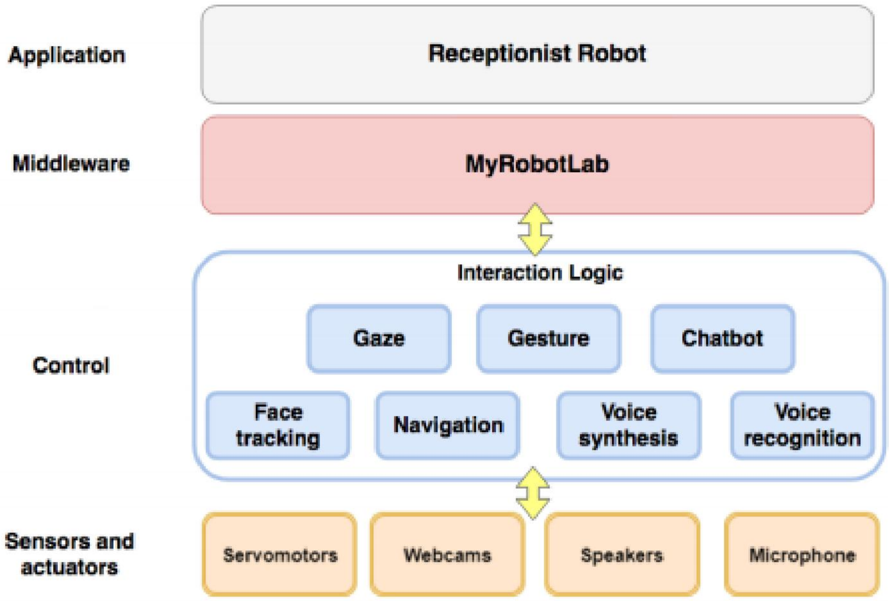


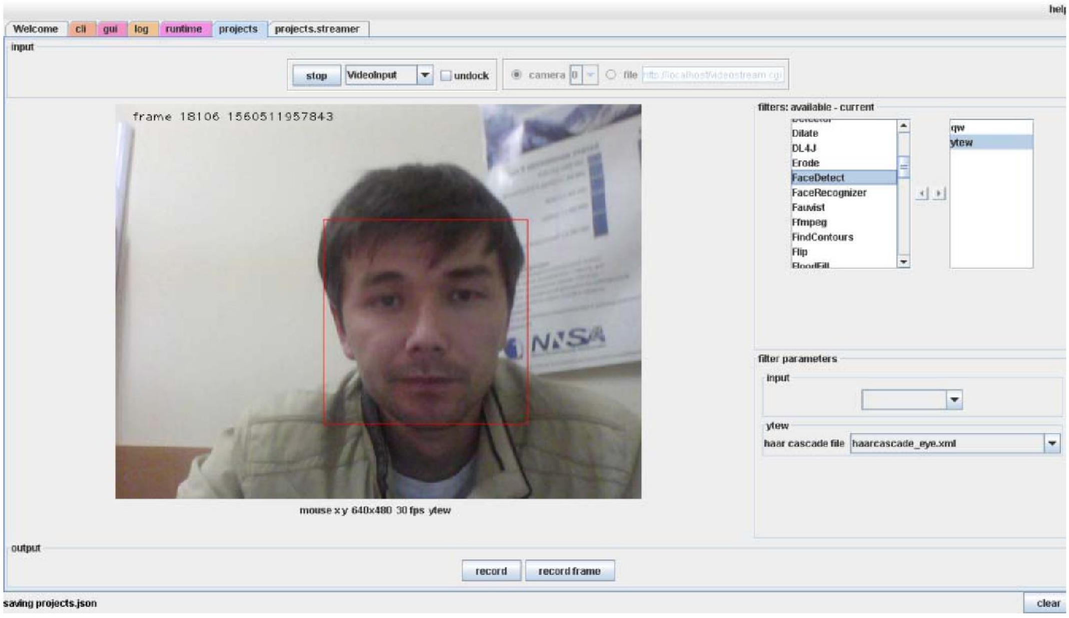
Figure 1 – The control layer consists of eight main modules that are used to control the directional functions of the robot

**Face tracking:** The module processes the video stream received from the cameras to detect the presence of human faces and their positions in the field of view of the robot. It is based on MyRobotLab (http://www.myrobotlab.org) A tracking module that can track human faces in real time using OpenCV (https://opencv.org) library. When a face is traced, the top of the robot is adapted to hold a specific face in the center of its field of vision. In the case of using a technique in which the robot is expected to be placed in public and crowded places, not all detected people may want to start a conversation. Therefore, in order to limit the number of unwanted activations, two events, i.e. found face and lost face were added to the original module. Therefore, in order to limit the number of unwanted activations, two events, i.e. found face and lost face were added to the original tracking module.

The found event is triggered when a human face is detected in a given number of sequential structures; likewise, a lost face event is triggered when no human face is detected in a predetermined number of structures. The data produced by this module is sent to look closely at the module.

**Gaze:** This module is responsible for the upper part of the guiding robot and eye movements during human user interaction. For example, in greetings and farewell phases, the robot's gaze is focused on the user's face. In the configuration studied, in which the robot gives directions by indicating destinations on a map, the gaze is directed to the map.

**Chat bot:** This module represents the brain of the system and produces responses to the text based on the received textual stimuli. A publicly accessible natural language processing language chatterbot that uses an XML schema called AIML (Artificial Intelligence Markup Language) to enable conversation customization With AIML it is possible to define the keywords/phrases that the robot needs to capture and understand (related to the greeting/farewell phases as well as the destinations) and should provide answers



frame 18106 1560511957843

mouse xy 640x480 30 fps ytew

haar cascade file haarcascade_eye.xml

saving projects.json

Figure 2 – Found face in the program MyRobotLab

(greeting/farewell expressions and directions). In addition, when a keyword/phrase relative to a destination is defined, the AIML language can be used to activate robot hand gestures to provide directions by sending the requested information to the Navigation module. Purple clouds represent examples of possible user input, while gray clouds are examples of possible robot responses.

**Voice recognition:** To allow the robot to communicate by voice with the user, two software tools are needed: the Text to Speech (TTS) tool to speak to the user and the Automatic Speech Recognition (ASR) tool to hear and understand what the teacher is saying. The first tool, TTS, is a technology that converts written information into spoken words, that is, TTS says any text that it receives as input. Conversely, ASR converts any human utterance captured by a robot microphone into written text that can be understood by a computer.

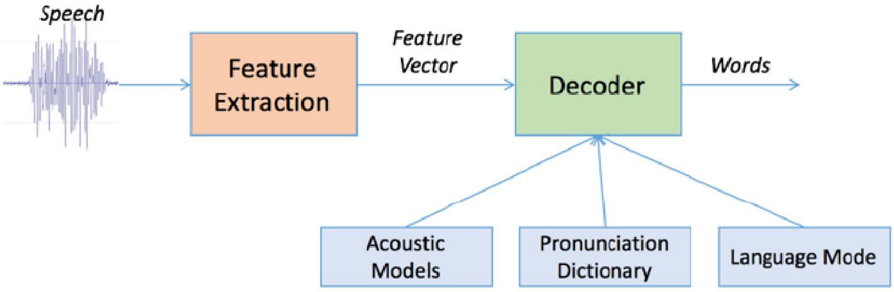


Figure 3 – Speech recognition engine

In [5], commercial TTS and ASR tools were used. TTS provides Application Interfaces (APIs) for both TTS and ASR. These APIs are wrapped in the form of two skills: Skill ETTS (Emotional Text for Speech) and Skill ASR [6]. They are wrapped in the form of skills, so they allow other skills to send the utterance into the ETTS skill and recover what the user said from the ASR Skill, simply using the communication mechanisms. This module receives voice commands from the microphone, converts them into text using Google's WebKit speech recognition API and sends the result to the Chatbot module.

**Voice synthesis:** This module allows the robot to talk. It receives text messages from the Chatbot module, converts them into audio files through MaryTTS (http://mary.dfki.de) speech synthesis engine and sends them to speakers. In addition, when a message is received, it triggers a moveMouth event that causes the robot's mouth to move, synchronizing with the spoken words.

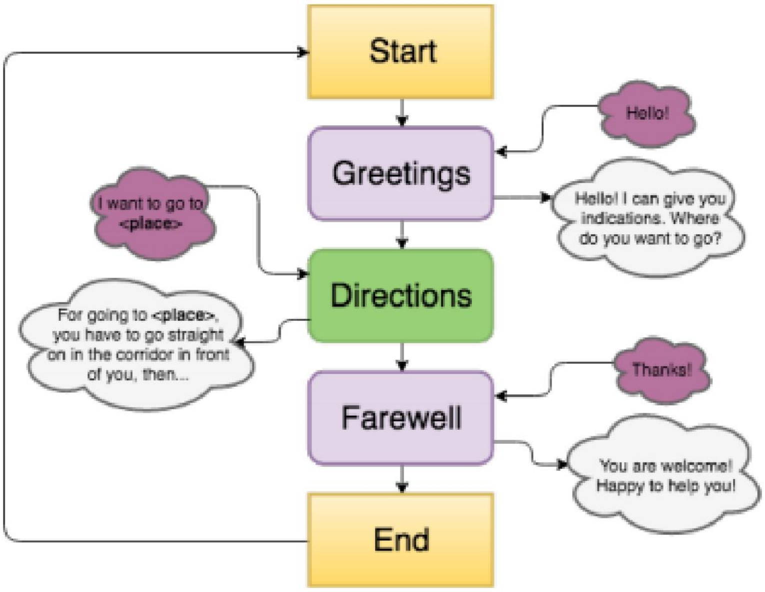

Start
Greetings
Directions
Farewell
End
Hello!
Hello! I can give you indications. Where do you want to go?
I want to go to <place>
For going to <place>, you have to go straight on in the corridor in front of you, then...
Thanks!
You are welcome! Happy to help you!

Figure 4 – Voice synthesis circuit block

**Navigation:** This is the main module that has been developed in this work and combined into MyRobotLab to provide users with directions for the desired destination. SLAM - updates the map of an unknown environment, while tracking the location of the agent in it [1]. Fig. 1 shows a simplified version of a common SLAM pipeline that operates as follows:

1. Inertial Measurement Unit, or IMU, consists of a gyroscope to measure angular velocity and accelerometers to measure acceleration in three axes. The IMU produces six data points (angular velocities in three different axes and acceleration in three axes) at a high rate and feeds the propagation stage with data.

2. The main task of the Propagation Unit is to combine the IMU data points and produce a new location. Since the IMU data is obtained at fixed intervals, combining the acceleration twice over time, we can obtain the agent movement during the last interval. However, since IMU hardware typically has bias and errors, we cannot fully rely on Propagation data so that positions do not gradually produce drift out of the actual path. To fix the drift problem, we use a camera to capture structures along the path at a fixed interest rate, typically 60 meters per second. Structures captured by the camera can be fed with a Feature Extraction Unit, which extracts useful angular singularities and produces a description for each feature. The extracted features can then feed the Mapping Unit to expand the map as the Agent explores. Note that we mean by map a collection of 3D points in space, each 3D point would correspond to one or more characteristic points found in the feature extraction unit.

3. In addition, the detected features would be sent to the Update Unit, which compares the features with the map. If the detected features already exist in the map, the Update unit can then retrieve the agent's current position from the known map points. With this new provision, the Update Unit can correct the drift introduced by the Propagation Unit. In addition, the Update unit updates the map with recently detected feature points.

In this implementation, we use our own SLAM system, which uses a stereo camera to image at a level of 60 meters per second, with each structure measuring 640 by 480 pixels. In addition, the IMU produces 200 Hz IMU updates (three axes of angular velocity and three axes of acceleration).

**Gesture:** This module was created as part of this work. Its role is to force the robot to execute a gesture sequence suitable for the particular directional modality that is being considered (in the air or a map indicating gestures) and a specific selected destination.

**Interaction logic:** This module controls all the previous ones based on the human robot interaction flow and the directional modality in use. As an example, while the robot says the use of Voice synthesis module, the voice recognition module should be stopped to avoid misconduct.

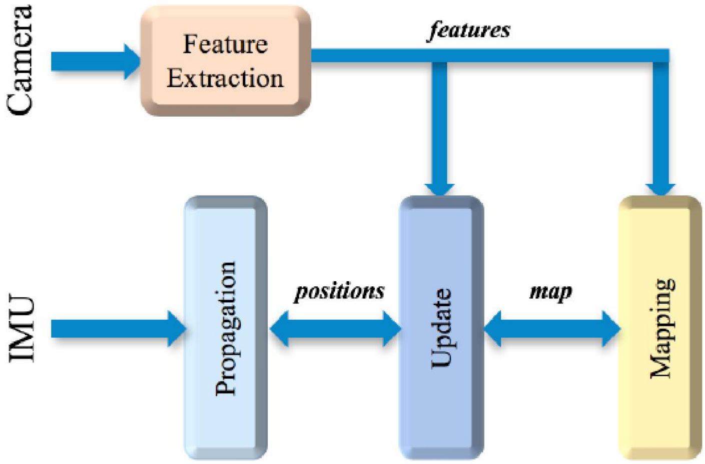


Figure 5 – Visual inertial execution of SLAM

The execution of the above modules is organized by the middleware layer, which is represented by MyRobotLab service and acts as an intermediary between the application layer and the robot functionality.

The stack is completed by an application layer that actually implements the reception logic illustrated in Figure 6, thus forcing the robot to interact with human users and give directions in a natural way. When the system starts, it initializes the MyRobotLab modules and waits for external stimuli to start the interaction. The stimuli can be either a detected face or a voice command given by the user. In the first case, the registrar's robot starts interacting with the phrase of congratulations: "Hello! I can give you signs! Where do you want to go? " Subsequently, the user can continue the interaction as shown in figure 5. If no answer is detected, the robot's profit is in the waiting phase. In the second case, the user starts interacting with congratulations to the robot or asking him about this destination. The interaction continues, as illustrated in figure 6.

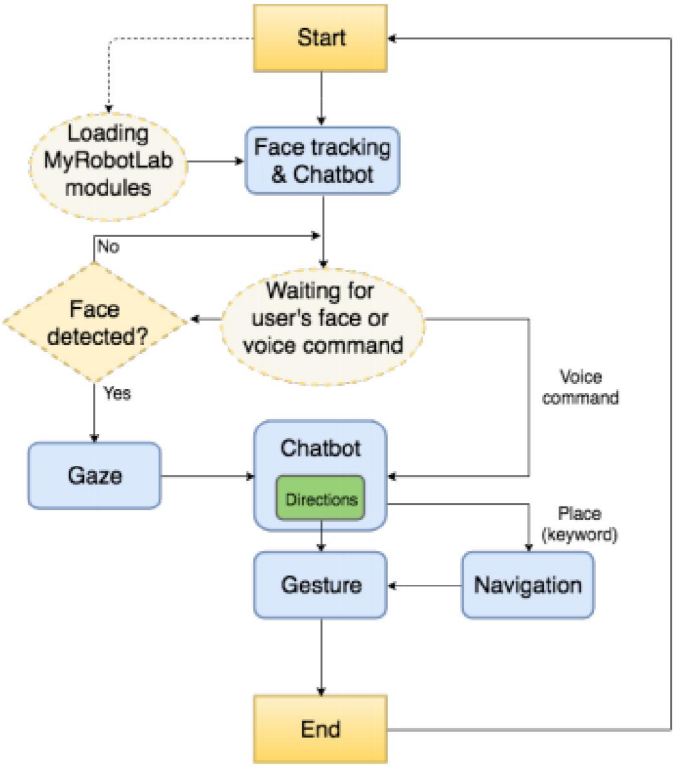


Figure 6 – Applied Logic of an Automated Verbal Robot

**Research results.** *Automated system on the Jetson TX1.* In this section, we present how we implement the aforementioned automated services on the Jetson TX1 SoC to maneuver the mobile robot chassis. We will install the hardware, system architecture, as well as the performance and power consumption of such an implementation.

*Hardware Installation.* For the robot and the implementation of services over it, we first need to install the hardware. The robot consists of two parts: the perception and decision units implemented on TX1 and the execution unit, which is the robot chassis. The robot chassis receives commands from TX1 and executes these commands accordingly.

This module generates stereo VGA resolution images at 60 meters per second along with IMU updates at 200 Hz. This source data is fed to the SLAM pipeline to produce accurate location updates and fed to the CNN pipeline to perform object recognition. In addition, the TX1 Board is connected to the main chassis via serial communication. Thus, after going through the stages of perception and decision, TX1 sends commands to the main chassis to move it. For example, after the SLAM pipeline creates an environment map, the solution pipeline may order the robot to move from location to location B, and commands are sent via the serial interface. For speech recognition, for commands we set to continuously perform audio playback to speech recognition. In addition, a 2,200 mAh battery is used to power the TX1 Board.

*System architecture.* In order to closely integrate these services on the installation of hardware, the next task is to design a system architecture. Figure 7 presents the system architecture that we implement on TX1. In the frontend, we have three sensor threads to produce the initial data: the camera thread produces images at a level of as much as 60 Hz, the IMU thread produces inertial updates at a rate of 200 Hz, and the microphone thread produces an audio signal at a rate of 8 kHz. The IMU image and data then enter the SLAM pipeline to update the position at a rate of 200 Hz. Meanwhile, when the robot moves, the SLAM pipeline also expands the environment map. Position updates, along with an updated map, are then sent to the navigation thread to decide how the robot makes its next move. Image data is also included in the object recognition pipeline to extract the labels of the objects the robot encounters. Object labels are then filed into a reaction unit, which contains a series of rules for what to do next when a specific label is detected. For example, a rule might be that every time a human face is detected, the robot should greet the person. The audio data passes through the speech recognition pipeline to extract commands, and then commands feed the command unit. A command unit stores a series of predefined commands, and if an incoming command matches one on the predefined command interface, then the action is triggered. For example, we execute the command "stop", every time the robot hears the word "stop", it stops all its ongoing actions.

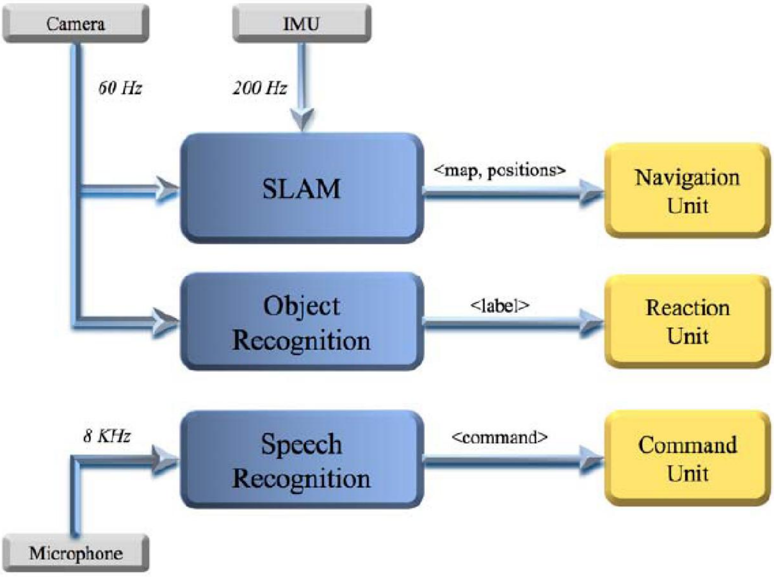


Figure 7 – System integration

In its own process, this architecture provides a very good separation of different tasks with each task. For different tasks to use the main heterogeneous computing resources should be fully connected to high efficiency and energy efficiency. For example, feature extraction operations used in frontend SLAM, as well as CNN computations, show very good data parallelism, so it might be advantageous to offload these tasks to the GPU, and free some CPU resources for another computation, or for energy efficiency. Therefore, in our implementation, the SLAM frontend is offloaded to the GPU, the SLAM backend is executed on the central processor; most object recognition is offloaded to the GPU; speech recognition task is performed on the central processor. We explore how this setup behaves on the Jetson TX1 SoC in the following subsections.

**Conclusion.** The conducted theoretical researches proved a new constructive and technological scheme of verbal robot, including pattern recognition, speech, navigation, localization. Using research methods of complex systems of machine learning, it is structured into blocks integrated into common platforms, the blocks are divided into elements, and the equations of omnidirectional movement of the robot are compiled and solved.

The modeling of a humanoid robot with the creation of mechanisms for the movement of robot organs has been developed. Electrical circuits for power supply and control of servos, sensors and motors have been constructed. The robot parts have been modeled on a 3 D printer and the robot body has been created. A system has been proposed in which various complex machine learning modules will be integrated.

Most existing implementations use high-level central processing units that consume more than 100 watts for multiple services per robot system. Robots are mobile systems with strict constraints on the energy. Meeting the real-time requirements for mobile robots, we have to simultaneously enable all of these services at approximately 10 watts of power. In this paper, we present our studies of integrating localization, vision and speech recognition services on a mobile robot.

On an Nvidia Jetson TX1 SoC, with approximately 10 watts of power consumption, a system architecture was designed. Using the central processor for other tasks provided by SoC, using the GPU mainly for frontend SLAM tasks, we will be able to efficiently use heterogeneous computing resources. To offload computer vision and speech recognition tasks to the cloud we tried to achieve high energy efficiency. While still meeting real-time requirements, our research shows that if the cloud is deployed on a local network, offloading these tasks can easily double the robot's battery life.

**М. Калимолдаев, М. Ахметжанов,**
**М. Кунелбаев, Т. Сундетов**

ҚР БҒМ ҒК ақпараттық және есептеуіш технология институты,
Алматы, Қазақстан

## ИНТЕГРАЛДЫҚ МАШИНАНЫ ОҚЫТУ ҮШІН АҚПАРАТТЫҚ ЖҮЙЕЛЕРД МЫСАЛЫ ВЕРБАЛДЫ РОБОТЫҢ ЖАСАУ

**Аннотация.** Жұмыста мысал ретінде ауызша роботты қолдана отырып, автоматтандырылған оқу модульдерінің ақпараттық жүйелері орындалды. Автоматтандырылған ауызша робот жүйесін енгізу үшін аппараттық құралдар жасалды, логикалық компоненттер жасалды, сонымен қатар OpenCV кітапханасы арқылы нақты уақыт режимінде адамның бет-бейнесін бақылай алатын модульдерді бақылау және Jetson TX1 SoC үшін автоматтандырылған қызмет көрсету жүзеге асырылды. жылжымалы робот шассиіне маневр жасау.

**Түйін сөздер:** ақпараттық жүйелер, кіріктірілген модульдер, машиналық оқыту, ауызша робот.

**М. Калимолдаев, М. Ахметжанов, М. Кунелбаев, Т. Сундетов**

Институт информационных и вычислительных технологий КН МОН РК, Алматы, Казахстан

## ИНФОРМАЦИОННЫЕ СИСТЕМЫ ИНТЕГРИРОВАННЫХ МОДУЛЕЙ МАШИННОГО ОБУЧЕНИЯ НА ПРИМЕРЕ ВЕРБАЛЬНОГО РОБОТА

**Аннотация.** В данной работе информационные системы интегрированных модулей машинного обучения были выполнены на примере словесного робота. Были разработаны аппаратные компоненты, логические компоненты, которые были собраны и разработаны для реализации автоматизированной системы словесных роботов, также было выполнено отслеживание модулей, которые могут отслеживать человеческие лица в режиме реального времени через библиотеку OpenCV и автоматизированные сервисы на Jetson TX1 SoC для маневрирования шасси мобильного робота.

**Ключевые слова:** информационные системы, интегрированные модули, машинное обучение, словесный робот.

**Information about authors:**

Kalimoldayev Maksat, Institute Information and Computational Technologies CS MES RK, Almaty, Kazakhstan; maks714@mail.ru;

Akhmetzhanov Maxat, Institute Information and Computational Technologies CS MES RK, Almaty, Kazakhstan; https://orcid.org/0000-0001-7890-5422

Kunelbayev Murat, Institute Information and Computational Technologies CS MES RK, Almaty, Kazakhstan; murat7508@yandex.ru; http://orcid.org/0000-0002-5648-4476

Sundetov Talgat, Institute Information and Computational Technologies CS MES RK, Almaty, Kazakhstan;

### REFERENCES

[1] Barber R. Ph.D. Thesis. Universidad Carlos III de Madrid; Leganes, Spain: 2000. Desarrollo de una Arquitectura Para Robots Móviles Autónomos. Aplicacioón a un Sistema de Navegación Topoloógica.

[2] Rivas R., Corrales A., Barber R., Salichs MA. Robot Skill Abstraction for AD Architecture // Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles; Toulouse, France. 3–5 September 2007.

[3] Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional; Harlow, UK: 1995.

[4] Quigley M., Gerkey B., Conley K., Faust J., Foote T., Leibs J., Berger E., Wheeler R., Ng A. ROS: An Open-Source Robot Operating System // Proceedings of the Open-Source Software Workshop of the International Conference on Robotics and Automation (ICRA); Kobe, Japan. 12–17 May 2009.

[5] Nuance Communications LTD. [(accessed on 1 June 2013)]. Loquendo web page. Available online:www.loquendo.com/.

[6] Alonso-Martin F., Salichs M. Integration of a voice recognition system in a social robot // Cybern. Syst. 2011; 42: 215-245.