: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889, 2024-
5 ResearchBib IF: 8.848 / 2024

THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

VOLUME-4, ISSUE-8

# Using the Kalman Filter to Represent Probabilistic Models for Determining the Location of a Person in Collaborative Robot Working Area

**Dmytro Gurin [1], Vladyslav Yevsieiev [1], Amer Abu-Jassar [2], Svitlana Maksymova [1]**

[1] Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2] Faculty of Information Technology, Department of Computer Science, Ajloun National University, Ajloun, Jordan

**Abstract:** The article is devoted to the study of the use of the Kalman filter for the development of probabilistic models for determining the location of a person in collaborative robot working area. The article discusses in detail the mathematical representation of the Kalman filter, including formulas for estimating and predicting system states based on measurements with noise. A software implementation of the Kalman filter was carried out, including the development of an algorithm for processing data from a video camera, which ensures accurate tracking of a person. As part of the research, a series of experiments was conducted to evaluate the effectiveness of the algorithm in real conditions, which confirms its ability to improve the accuracy of location determination and increase the safety of collaborative robots in dynamic environments.

**Key words:** Industry 5.0, Collaborative Robot, Mobile Robots, Work Area, Computer Vision.

## Introduction

In the modern world, it is almost impossible to imagine production without the use of robots [1]-[11]. Moreover, robots, including humanoid robots, are already widely used in everyday life and the social sphere [12]-[24]. Both classical approaches [25]-[32] and other special methods [33]-[41] can be used here.

In modern robotic systems, especially in the context of collaborative robots, the accurate location of a person in the work area is critical to ensure safety and work efficiency [42]-[45]. Using the Kalman filter to represent probabilistic models is a powerful tool that allows you to solve this problem due to its advantages in processing and integrating data from different sources. The Kalman filter is a mathematical algorithm for estimating and predicting system states based on incomplete and noisy measurements, which is especially important in real-time environments where accuracy and speed of response are critical. The use of this filter in robotic systems helps not only to reduce the impact of measurement errors, but also to increase the reliability and efficiency of collaborative robots, which in turn contributes to their integration into complex and dynamic production environments. Understanding and optimizing the processes involving the Kalman filter is essential to designing systems that can effectively and safely interact with humans while providing high levels of performance and security.

## Related works

The creation of Human-Robot Interaction (HRI), one of the tasks of which is to determine the location of a person, is an extremely urgent task. Many scientists and researchers are engaged in its solution, therefore, many scientific works on this topic appear. Let us consider several recent such works.

The ability to recognize human partners is an important social skill to build personalized and long-term Human-Robot Interactions [46]. Authors in [46] propose a framework to allow robots to autonomously organize their sensory experience into a structured dataset suitable for person recognition during a multiparty interaction.

Paper [47] introduces a set of conventions and standard interfaces for HRI scenarios, designed to be used with the Robot Operating System [47]. It directly aims at promoting interoperability and reusability of core functionality between the many HRI-related software tools, from skeleton tracking, to face recognition, to natural language processing.

HRI is a required method of information interaction in the age of intelligence [48]. The study [48] proposes a novel gaze-point-driven interaction framework using only RGB cameras to provide a more convenient and less restricted way of interaction.

Müller, S., and co-authors in [49] present a modular detection and tracking system that models position and additional properties of persons in the surroundings of a mobile robot.

Face recognition became a key element in social cognition which is used in various applications including HRI, pedestrian identification, and surveillance systems [50]. Authors in [50] present robust face recognition and tracking framework in unconstrained settings.

Scientists in [51] propose an HRI system, referred to here as Robot-Facilitated Interaction System, by which an autonomous mobile robot transporting a patient facilitates interaction with an accompanying caretaker monitoring and nursing the patient.

Researchers in [52] introduce the Robot-Centric Group Detection and Tracking System, a new method that enables robots to detect and track groups of people from an ego-centric perspective using a crowd-aware, tracking-by-detection approach [53].

Thus, we see that the range of problems arising during the creation of the HRI is quite wide. Further in this article we will consider our system of determining the location of a person.

## Mathematical model of the Kalman filter for solving the problem of determining the location of a person in collaborative robot working area

The choice of the Kalman filter for the task of determining the location of a person in collaborative robot working area through a computer vision system is justified by its ability to effectively process and integrate data with noise. The Kalman filter is a powerful tool for estimating and predicting the state of a system based on incomplete and noisy measurements, which is especially important in real-time environments where accuracy and speed are critical. With its help, you can accurately determine the position of a person in a dynamic environment, taking into account unpredictable changes in the behavior and position of the object. The Kalman filter reduces the influence of measurement errors and improves the stability of the tracking system, which ensures a high level of safety and reliability of collaborative robots. The integration of this filter with computer vision provides accurate and timely determination of a person's location, which is critical for effective management of robots in complex and changing working environment conditions.

Here is a mathematical representation of the use of the Kalman filter to solve the problem of determining the location of a person in collaborative robot work area. Let us use the following parameters:

– the State parameter in the Bayesian filter is a vector ($x_k$), which contains all the information about the system that is necessary to predict its future state based on the current state and new observations. In the context of determining the location of a person in collaborative robot working area based on streaming video, the state vector can include the following parameters: Position ($x$, $y$) where in our case $x$ - the coordinate of the person's position in the horizontal direction (pixels on the image) and $y$ - the coordinate of the person's position in the vertical direction (pixels); Velocity ($v_x$, $v_y$) where $v_x$ is the speed of a person in the horizontal direction and $v_y$ is the speed of a person in the vertical direction. In this research, velocity describes how much a position changes over time and can be used to predict a person's future positions; Acceleration ($a_x$, $a_y$), where $a_x$ is the acceleration of a person in the horizontal direction and $a_y$ is the acceleration of a person in the vertical direction. This parameter makes it possible to simulate dynamic changes in movement, such as acceleration or deceleration; Orientation and Angular Velocity ($\theta$, $\omega$), where $\theta$ is the angular orientation of a person (for example, turning around a vertical axis) and $\omega$ is the angular velocity, which describes the rate of change in orientation. These parameters make it possible to take into account how a person turns or orients himself in space. We also need to take into account the Additional Features parameter (*Size*, *LightingConditions*), where *Size* includes information about the dimensions of a person, such as width and height (in pixels) and *LightingConditions* contains information (data at a given time) about lighting conditions, which can affect video quality and detection accuracy. According to the parameters listed above, the mathematical representation $\overline{x_k}$ , where *k* is the moment of time, has the following form:

$$\overline{x_k} = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \\ a_{x,k} \\ a_{y,k} \\ \theta_л \\ \omega_k \\ \ldots \end{bmatrix} \qquad (1)$$

where each element of the vector represents a specific aspect of the system state. The number and type of parameters in the state vector may vary depending on the specific task and system characteristics.

Then the Process Model is the model of the evolution of the state over time, will have the following form. Let $\overline{x_k}$ is the state vector at the moment of time *k*, then the evolution model can be described by the following equation:

$$\overline{x_k} = \overline{F}\,\overline{x_{k-1}} + B\overline{u_k} \qquad (2)$$

$\overline{F}$ - State Transition Matrix, describes how the state vector changes over time. This matrix defines how previous state values affect the new state value. It uses the previous state vector $\overline{x_{k-1}}$ to calculate the predicted state $\overline{x_k}$ ;

$\overline{x_{k-1}}$ - the state vector at the previous moment in time *k-1*. It includes all parameters that describe the state of the system at a previous point in time. It is used as a basis for predicting a new state using a transition matrix $\overline{F}$

*B* - Control Matrix describes how the control vector $\overline{u_k}$ affects the change in the state of the system. It transforms the control vector into changes that are taken into account when calculating the new state.

$\overline{u_k}$ - Control Vector contains external control inputs or commands that may affect the state of the system. These can be, for example, control signals or pulses that directly affect the dynamics of the system.

After receiving new observations, the state vector is updated taking into account the new data, it can be mathematically represented as follows:

$$\hat{x}_k = \hat{x}_{\bar{k}} + K_k(z_k - H\hat{x}_{\bar{k}}) \qquad (3)$$

$\hat{x}_k$ - the updated state vector at the moment of time *k*. Provides an opportunity to assess the state of the system, which takes into account new observations. It is the final result of updating the predicted state $\hat{x}_{\bar{k}}$, taking into account new data $z_k$;

$\hat{x}_{\bar{k}}$ - the predicted state vector at the moment of time *k* (previously calculated). This is a preliminary estimate of the state of the system before receiving new observations, which is obtained from the state evolution model;

$K_k$ - the Kalman coefficient (Kalman Gain). A matrix that defines how strongly new observations should affect state updates. It is calculated so as to minimize the mean squared error of the state estimate;

$z_k$ - the observation vector at the moment of time $k$. These are actual measured data or observations from the system, such as a person's coordinates obtained from a video camera;

$H$ - Observation Matrix describes how a state vector $\hat{x}_k$ is transformed into an observation vector $z_k$, and defines which components of the state are visible or measurable through observations;

$(z_k - H\hat{x}_{\bar{k}})$ - the difference between actual observations and predicted observations. It is the residual (or innovation) that shows how much the predicted state $\hat{x}_{\bar{k}}$ differs from the actual observations $z_k$, reflecting the discrepancy between the forecast and the actual data.

The mathematical formula for representing the Kalman coefficient ($K_k$) from expression 3 has the next form:

$$K_k = P_{\bar{k}} H^T (H P_{\bar{k}} H^T + R)^{-1} \tag{4}$$

$P_{\bar{k}}$ – covariance matrix that describes the uncertainty in the predicted state estimate. The smaller this matrix, the more accurate the forecast. It affects how much weight should be given to new observations compared to the predicted state;

$H$ - Observation Matrix describes how a state vector is transformed into an observation vector, and defines which components of the state are visible or measurable through observations;

$H P_{\bar{k}} H^T$ - the covariance matrix of the observation describes the uncertainty or variance of the predicted observations. It is an estimate of how much predicted observations may vary. It takes into account how predicted state errors are reflected in observations;

$R$ - the covariance matrix of the observation noise describes the uncertainty or noise in the observations. This determines the accuracy of measurements and indicates the level of error in observations;

$(H P_{\bar{k}} H^T + R)^{-1}$ - the inverse matrix, which is a combination of the observation covariance matrix and the observation noise matrix, is essentially part of the calculation of the Kalman coefficient used to correct the state estimate. It shows how much the predicted observations can be adjusted based on their uncertainty and the uncertainty of the observations.

It should be noted that within the framework of these studies, the term "covariance" in statistics and probability theory describes how two random variables change together. It is a measure of how a change in one variable is related to a change in another. Covariance is used to estimate the relationship between two variables.

**Software implementation of the Kalman filter for representing probabilistic models for determining the location of a person in collaborative robot working area**

The choice of the Python language for the implementation of the Kalman filter in the task of determining the location of a person in collaborative robot working area through a computer vision system is justified by several good reasons. Python is known for its ease of learning and use, thanks to its clear syntax and code readability, which greatly facilitates the development of complex algorithms such as the Kalman filter. In addition, Python offers powerful libraries for image processing, such as OpenCV, as well as machine learning tools, such as TensorFlow and PyTorch, allowing the efficient integration of the Kalman filter with computer vision. The language also has an active community of developers, which provides access to a large number of resources, documentation and support, which is essential for rapid development and debugging of software solutions. Python's high performance in video and image processing thanks to its libraries allows the implementation of probabilistic models in real time, which is critical for accurate tracking of a person's position in a dynamic environment. Given these factors, Python is the optimal choice for developing a system that integrates the Kalman filter with computer vision to provide accuracy and efficiency in positioning in robotic systems.

We will give an example of the implementation of some functions in the developed Kalman filter implementation program for presenting probabilistic models for determining the location of a person in collaborative robot working area.

```
def initialize_kalman():
    kalman = cv2.KalmanFilter(4, 2)
    kalman.statePre = np.array([0, 0, 0, 0], dtype=np.float32)
    kalman.statePost = np.array([0, 0, 0, 0], dtype=np.float32)

    kalman.transitionMatrix = np.array([[1, 0, 1, 0],
                                        [0, 1, 0, 1],
                                        [0, 0, 1, 0],
                                        [0, 0, 0, 1]], dtype=np.float32)
    kalman.measurementMatrix = np.array([[1, 0, 0, 0],
                                         [0, 1, 0, 0]], dtype=np.float32)
    kalman.processNoiseCov = np.array([[1e-2, 0, 0, 0],
                                       [0, 1e-2, 0, 0],
                                       [0, 0, 1e-2, 0],
                                       [0, 0, 0, 1e-2]], dtype=np.float32)
    kalman.measurementNoiseCov = np.array([[1e-1, 0],
                                           [0, 1e-1]], dtype=np.float32)
    kalman.errorCovPost = np.eye(4, dtype=np.float32)
    return kalman
```

This piece of code is responsible for initializing the Kalman filter object in the OpenCV library. It adjusts the filter parameters, including the transition, observation, process noise, and observation noise matrices, to provide efficient tracking and prediction of the system state. The Kalman filter will be used to correct and predict the location of the object in real time.

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
```

This piece of code is used to detect faces in an image. It creates a face detector object using a Haar classifier and applies it to a grayscale image to find regions likely to contain faces with specified scale and minimum size parameters.

```
kalman.correct(measurement)
prediction = kalman.predict()
```

This piece of code uses a Kalman filter to correct and predict the state of the system. The function kalman.correct(measurement) updates the filter state estimate based on the new measurements, and kalman.predict() calculates the next predicted state of the system, taking these corrections into account.

The developed user interface of the program for presenting probabilistic models for determining the location of a person in collaborative robot working area based on the Kalman filter through the computer vision system is shown in Figure 1.

: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889, 2024-
5 ResearchBib IF: 8.848 / 2024
THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY
VOLUME-4, ISSUE-8

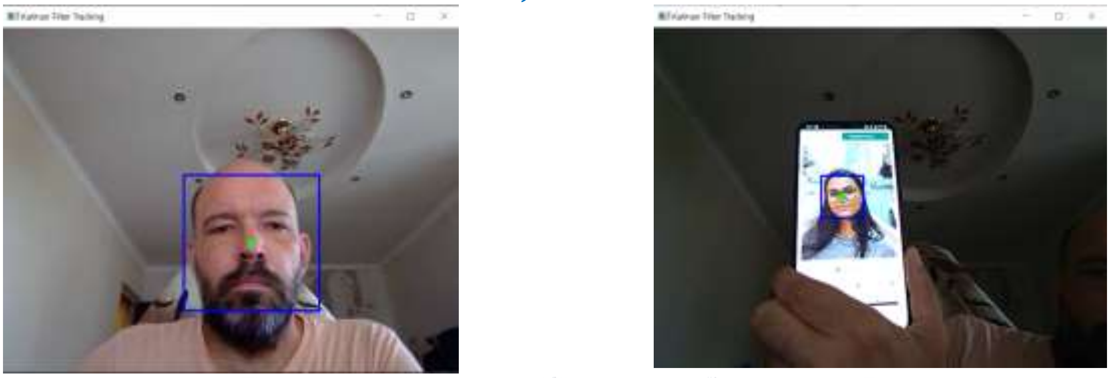

**Figure 1:** The developed interface of the program for presenting probabilistic models for determining the location of a person in collaborative robot working area based on the Kalman filter

On the basis of the developed program, a number of experiments were conducted to investigate the location of a person in collaborative robot working area based on the Kalman filter. For the purity of the experiment, we note that the hardware consisted of the following elements: CPU Intel Core i7-6650U, 3.4 GHz; RAM – 16Mb; HDD - 512Gb; GPU Intel Iris Graphics 540. The obtained results are presented in Table 1, and for the convenience of visualizations for data analysis are presented in the form of graphs in Figure 2.

**Table 1:** Data obtained during the testing of determining the location of a person in collaborative robot working area using the Kalman filter

| Test case | Measured location (X, Y) | Predicted location (X, Y) | Divergence (X, Y) | Velocity ($v_x$, $v_y$) | Accuracy (mm) | Forecast error (m) |
|---|---|---|---|---|---|---|
| Test 1 | (100, 150) | (102, 148) | (2, -2) | (1.2, -1.1) | 5.0 | 0.05 |
| Test 2 | (200, 250) | (198, 252) | (-2, 2) | (-1.0, 1.3) | 4.8 | 0.04 |
| Test 3 | (300, 350) | (305, 340) | (5, -10) | (2.0, -1.5) | 6.0 | 0.07 |
| Test 4 | (400, 450) | (398, 455) | (-2, 5) | (-1.1, 1.0) | 5.5 | 0.06 |
| Test 5 | (500, 550) | (505, 545) | (5, -5) | (1.5, -1.2) | 5.2 | 0.05 |

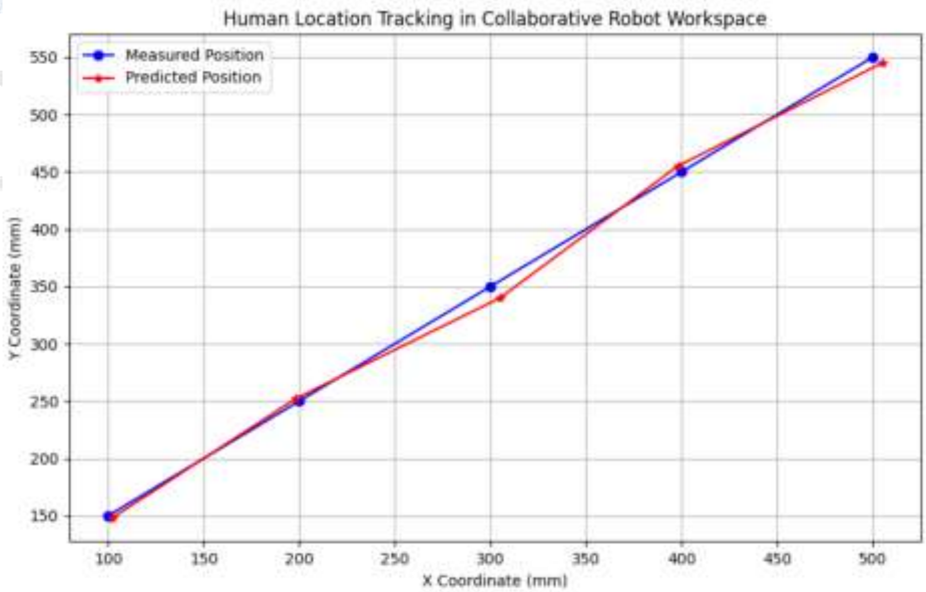


**Figure 2:** Comparison graph of the obtained experimental results during the testing of determining the location of a person in collaborative robot working area using the Kalman filter.

: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889, 2024-
5 ResearchBib IF: 8.848 / 2024
THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY
VOLUME-4, ISSUE-8

Based on the test results, it is possible to make the following conclusions that the results obtained during the testing of determining the location of a person in collaborative robot working area using the Kalman filter indicate the effectiveness of using this algorithm. The measured location is close to the predicted location, with small differences, confirming the accuracy of the filter. The differences between measurements and predictions vary but remain within an acceptable range, indicating the adequacy of the Kalman filter settings for the given task. The accuracy and prediction error rates demonstrate that the filter provides stability and reliability in determining human positions. The data also confirms that the Kalman filter effectively reduces the impact of noise and measurement errors, which is critical for the safety and efficiency of collaborative robots in dynamic environments.

## Conclusion

Using a Kalman filter to locate a human in a collaborative robot's work area has shown significant benefits in terms of improving tracking accuracy and reliability. The mathematical model of the Kalman filter discussed in the article demonstrated its ability to effectively process and integrate data from a video camera, taking into account noise and unpredictable changes in the environment. Implementation of the program in Python provided practical verification of theoretical calculations and allowed to evaluate the efficiency of the filter in real conditions. The results of the tests confirmed that the Kalman filter reduces the discrepancies between the measured and predicted positions, which indicates the high accuracy of the model. The identified forecast errors were minimal, and the location accuracy was within acceptable limits. These results indicate that the Kalman filter is an effective tool for providing reliable human tracking in a dynamic work area environment, which is critical for the safety and functioning of collaborative robots.

## References:

1. Stetsenko, K., & et al. (2023). Exploring BEAM Robotics for Adaptive and Energy-Efficient Solutions. Multidisciplinary Journal of Science and Technology, 3(4), 193-199.
2. Maksymova, S., & et al. (2024). Gripping Device Development: Some Aspects. Journal of Universal Science Research, 2(1), 150–158.
3. Basiuk, V., & et al. (2023). Mobile Robot Position Determining Using Odometry Method. Multidisciplinary Journal of Science and Technology, 3(3), 227-234.
4. Невлюдов, І. Ш., & et al. (2024). BEAM робототехніка. Кривий Ріг : Видавець Чернявський Д. О., 276 с., ISBN 978-617-8045-79-1
5. Abu-Jassar, A. T., Al-Sharo, Y. M., Lyashenko, V., & Sotnik, S. (2021). Some Features of Classifiers Implementation for Object Recognition in Specialized Computer systems. TEM Journal: Technology, Education, Management, Informatics, 10(4), 1645-1654.
6. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural networks as a tool for pattern recognition of fasteners. International Journal of Engineering Trends and Technology, 69(10), 151-160.
7. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.
8. Lyashenko, V., Abu-Jassar, A. T., Yevsieiev, V., & Maksymova, S. (2023). Automated Monitoring and Visualization System in Production. International Research Journal of Multidisciplinary Technovation, 5(6), 9-18.
9. Abu-Jassar, A. T., Attar, H., Lyashenko, V., Amer, A., Sotnik, S., & Solyman, A. (2023). Access control to robotic systems based on biometric: the generalized model and its practical implementation. International Journal of Intelligent Engineering and Systems, 16(5), 313-328.
10. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2024). Remote Monitoring System of Patient Status in Social IoT Environments Using Amazon Web Services (AWS) Technologies and Smart Health Care. International Journal of Crowd Science.

11. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2024). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. International Journal of Crowd Science.

12. Yevsieiev, V., & et al. (2024). Humanoid Robot Gripping Device 3D Model Development Siemens NX Unigraphics. Multidisciplinary Journal of Science and Technology, 4(7), 189-196.

13. Yasser, A. S., & et al. (2023). A Robo-hand prototype design gripping device within the framework of sustainable development. Indian Journal of Engineering, 2023, 20, e37ije1673.

14. Maksymova, S., & et al. (2024). The Bipedal Robot a Kinematic Diagram Development. Journal of Universal Science Research, 2(1), 6–17.

15. Yevsieiev, V., & et al. (2024). Humanoid Robot Movement Simulation in ROS. Multidisciplinary Journal of Science and Technology, 4(7), 146-154.

16. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.

17. Ahmad, M. A., Sinelnikova, T., Lyashenko, V., & Mustafa, S. K. (2020). Features of the construction and control of the navigation system of a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(4), 1445-1449.

18. Maksymova, S., Matarneh, R., & Lyashenko, V. V. (2017). Software for Voice Control Robot: Example of Implementation. Open Access Library Journal, 4, e3848.

19. Lyashenko, V., & et al.. (2021). Semantic Model Workspace Industrial Robot. International Journal of Academic Engineering Research, 5(9), 40-48.

20. Sotnik, S., & et al.. (2022). Analysis of Existing Infliences in Formation of Mobile Robots Trajectory. International Journal of Academic Information Systems Research, 6(1), 13-20.

21. Sotnik, S., & et al.. (2022). Modern Industrial Robotics Industry. International Journal of Academic Engineering Research, 6(1),. 37-46.

22. Lyashenko, V., & et al.. (2021). Modern Walking Robots: A Brief Overview. International Journal of Recent Technology and Applied Science, 3(2), 32-39.

23. Sotnik, S., & et al.. (2022). Overview of Innovative Walking Robots. International Journal of Academic Engineering Research, 6(4), 3-7.

24. Sotnik, S., & et al.. (2022). Agricultural Robotic Platforms. International Journal of Academic Engineering Research, 6(4), 14-21.

25. Гиренко, А. В., Ляшенко, В. В., Машталир, В. П., & Путятин, Е. П. (1996). Методы корреляционного обнаружения объектов. Харьков: АО “БизнесИнформ, 112.

26. Matarneh, R., & et al. (2017). Building robot voice control training methodology using artificial neural net. International Journal of Civil Engineering and Technology, 8(10), 523–532.

27. Maksymova, S., & et al. (2017). Software for voice control robot: example of implementation. Open Access Library Journal, 4, 1-12.

28. Putyatin, Y. P., & et al.. (2016) The Pre-Processing of Images Technique for the Material Samples in the Study of Natural Polymer Composites. American Journal of Engineering Research, 5(8), 221-226.

29. Lyashenko, V., Matarneh, R., & Kobylin, O. (2016). Contrast modification as a tool to study the structure of blood components. Journal of Environmental Science, Computer Science and Engineering & Technology, 5(3), 150-160.

30. Lyubchenko, V., & et al.. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.

31. Lyashenko, V. V., Matarneh, R., Kobylin, O., & Putyatin, Y. P. (2016). Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology. International Journal, 4(1), 85-94.

32. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. International Journal of Science and Research (IJSR), 3(11), 2870-2877.

33. Mustafa, S. K., Yevsieiev, V., Nevliudov, I., & Lyashenko, V. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. SSRG International Journal of Engineering Trends and Technology, 70(1), 139-145.

34. Vasiurenko, O., Lyashenko, V., Baranova, V., & Deineko, Z. (2020). Spatial-Temporal Analysis the Dynamics of Changes on the Foreign Exchange Market: an Empirical Estimates from Ukraine. Journal of Asian Multicultural Research for Economy and Management Study, 1(2), 1-6.

35. Mustafa, S. K., Ayaz, A. M., Baranova, V., Deineko, Z., Lyashenko, V., & Oyouni, A. A. A. (2020). Using wavelet analysis to assess the impact of COVID-19 on changes in the price of basic energy resources. International Journal of Emerging Trends in Engineering Research, 8(7), 2907-2912.

36. Matarneh, R., Tvoroshenko, I., & Lyashenko, V. (2019). Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space. International Journal of Recent Technology and Engineering, 8(4), 1687-1693.

37. Babker, A. M., Abd Elgadir, A. A., Tvoroshenko, I., & Lyashenko, V. (2019). Information technologies of the processing of the spaces of the states of a complex biophysical object in the intellectual medical system health. International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3221-3227.

38. Khan, A., Joshi, S., Ahmad, M. A., & Lyashenko, V. (2015). Some effect of Chemical treatment by Ferric Nitrate salts on the structure and morphology of Coir Fibre Composites. Advances in Materials Physics and Chemistry, 5(1), 39-45.

39. Kuzemin, O., & Lyashenko, V. Microsituation Concept in GMES Decision Support Systems/A. Kuzemin, V. Lyashenko. Intelligent Data Processing in Global Monitoring for Environment and Security (pp. 217–238).–2011.–P, 217-238.

40. Kuzemin, A., & Lyashenko, V. (2008). Conceptual Foundations of Construction of the Models and Procedures for Prediction of the Avalanche-dangerous Situations Initiation. International Journal INFORMATION THEORIES & APPLICATIONS, 15(2), 153-158.

41. Zeleniy, O., Rudenko, D., Lyubchenko, V., & Lyashenko, V. (2022). Image Processing as an Analysis Tool in Medical Research. Image, 6(9), 135-141.

42. Samoilenko, H., & et al. (2024). Review for Collective Problem-Solving by a Group of Robots. Journal of Universal Science Research, 2(6), 7-16.

43. Nevliudov, I., & et al. (2023). Mobile Robot Navigation System Based on Ultrasonic Sensors. In 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), IEEE, 1, 247-251.

44. Nevliudov, I. S., & et al. (2023). Conveyor Belt Object Identification: Mathematical, Algorithmic, and Software Support. Appl. Math. Inf. Sci. 17(6), 1073-1088.

45. Nevliudov, I., & et al. (2023). A Small-Sized Robot Prototype Development Using 3D Printing. In XXXI International Conference CAD In Machinery Design Implementation and Educational Issues, 12.

46. Gonzalez, J., & et al. (2021). Towards a cognitive framework for multimodal person recognition in multiparty hri. In Proceedings of the 9th International Conference on Human-Agent Interaction, 412-416.

47. Mohamed, Y., & Lemaignan, S. (2021). Ros for human-robot interaction. In 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 3020-3027.

48. Li, W., & et al. (2021). A novel gaze-point-driven HRI framework for single-person. In Collaborative Computing: Networking, Applications and Worksharing: 17th EAI International Conference, CollaborateCom 2021, Virtual Event, October 16-18, 2021, Proceedings, Part I 17, Springer International Publishing, 661-677.

49. Müller, S., & et al. (2020). A multi-modal person perception framework for socially interactive mobile service robots. Sensors, 20(3), 722.

50. Khalifa, A., & et al. (2022). Face recognition and tracking framework for human–robot interaction. Applied Sciences, 12(11), 5568.

51. Lee, S., & et al. (2023). Robot-Facilitated Human–Robot Interaction with Integrated Tracking, Re-identification and Gesture Recognition. In International Conference on Intelligent Autonomous Systems, Cham: Springer Nature Switzerland, 257-275.

52. Taylor, A., & Riek, L. D. (2022). Regroup: A robot-centric group detection and tracking system. In 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 412-421.

53. Yevsieiev, V., & et al. (2024). Building a traffic route taking into account obstacles based on the A-star algorithm using the python language. Technical Science Research In Uzbekistan, 2(3), 103-112.