Organizing Interactions Between a Robotic Manipulator and a Technological Equipment Control Panel

N V Kim\(^1\), V N Zhidkov\(^1\), V V Polyansky\(^1\)

\(^1\)Moscow Aviation Institute (National research university)

Abstract. Algorithms for the organization of standalone operation of a robotic manipulator have been considered in this Article. The controls on the control panel of a man-made equipment piece have been the manipulated object. A robotic manipulator fitted with a control system, and a machine vision system, have been the subsystems under study. The robotic manipulator interacts with the objects to switch on and off a toggle switch, turn a rotary switch, press on a pushbutton, etc. Manipulating the objects in question involves assessing of their positions and orientations, which is achieved through the machine vision system. Coordinated operation of the subsystems is required to plan and to implement control of the manipulator, while taking into account positions of the objects relative to each other. A configuration of interactions between the manipulator and the control panel, a method for organizing coordinated functioning of the machine vision system when manipulating controls, which includes assessing positions of the object of interest relative to the robotic manipulator, and an algorithm for planning control based on artificial neural network technology have been presented in this Article.

1. Introduction

Robotics is currently proliferating [1-6]; industrial, household, social [2, 3], informational, recreational and other type robots are gaining currency; robots that are capable of interacting with humans actively and safely in standalone or semi-standalone mode, replacing them in various fields, are in the highlight now [7-9]. These robots are becoming ubiquitous in socially important areas of life, as well as in industries. In many cases, these robots are fitted with a number of functions [10-14], namely

- Environment, object, face etc recognition;
- Communicating with humans by voice, via visual or tactile channels;
- Planning and implementing autonomous activity;
- Etc.

2. Task formulation

In some robot designs, significant extensions to the functional, which mean improved usage efficiency, may involve organizing coordinated operation of robotic manipulators (RM) and using onboard machine vision systems (MVS). In particular, the RM arrangement in question, allows the RM doubling as a control panel operator of complex technological systems, e.g., switching toggle switches on and off, turning a rotary switch, etc.

Manipulating these objects involves their visual detection (search and localization), recognition, position assessment (relative to the robot’s coordinate system), and orientation [15-18]. In many cases, a necessity to solve such problems arises whenever the RM position is unclear relative to the object’s position and its target delivery point – stationary or traveling. Procedures related to detection,
recognition, and localization of an object, as well as the target delivery point, are implemented through the MVS.

Here, complex positioning problems (the inverse kinematics problems) have to be solved in real time, through, e.g., numeric methods for each specific manipulation task [19], which can be problematic due to limited productivity of onboard computers.

Thus, the operations of the MVS and the RM have to be coordinated to accomplish the whole object of interest manipulation process, and a computationally efficient algorithm for solving the inverse problem has to be developed.

3. Manipulator working on a control panel

Let us consider a RM that handles a control panel (CP) of a technological equipment piece, see Fig 1 below, with three controls types installed on it, namely, pushbuttons, toggle switches, and a rotary switch/key.

A collaborative robot (cobot) similar to CR5 cobot manufactured by SPA Android Technics has been chosen as the RM basic design, and has the characteristics listed below:

– Number of degrees of freedom: 6;
– Load capacity: up to 1 kg;
– Work area on the horizontal plane: up to 0.5 m²;
– Work areas on the vertical plane: up to 0.4 m²;
– Manipulator weight with gripper installed: not exceeding 10 kg.

The gripper is fitted to the output link, while the RM basis is stationary and is definitively positioned relative to the CP located within the RM service area. The CP is located on the plane $X_N Y_N$ of the coordinate system $N X_N Y_N Z_N$ of the operator cabin.

The gripper fitted to the manipulator has been also chosen from the product line manufactured by Android Technics, and is a three-fingered gripping tool with independent phalanx motion in each finger. A three-fingered gripper has been chosen as being best suited to manipulating objects on the CP – toggle switches, pushbuttons, and a rotary switch or a similar switch/key.

![Figure 1. The manipulator and the control panel: a general layout.](image)

Using the RM to manipulate the CP a two-stage control process:

1. Shifting the output link of the RM to a chosen work point above a manipulation object selected;
   2. Switching the manipulation object to a target position by using the gripper.

The coordinates of work points and the tolerance values have been found from early analysis of the controls designs:
- $H_z = 20 \text{ mm}$ is the span between the point $M$ on the RM output link (the gripper installation axis) and the CP plane;
- $|\Delta z| \leq 2 \text{ mm}$ is the gripper positioning error on the axis $Z$;
- $|\Delta x|, |\Delta y| \leq 1 \text{ mm}$ is the linear error of the gripper axis positioning relative to a control axis;
- $|\Delta \alpha| \leq 10^0$ is the angular error of the gripper axis positioning relative to a control axis.

4. Finding $O V_k$ and assessing a manipulated object relative to the robotic manipulator

The MVS of the robot may use various image processing and analysis technologies [13-15]. E.g., artificial neural network systems, correlation algorithms, characteristic point detection-based algorithms, etc. are typically used to detect and to recognize various objects of interest [13, 15].

Suppose the RM installed inside the operator cabin has been given a task to throw a switch. The CP, the RM, and the camera that form the MVS are stationary, and the CP is within the camera’s view. The problem (not considered here) of finding a desired object of interest can be solved either through identifying each object from its coordinates that have been input into a prepared database, or by visual marking of each object with its respective designated codes.

Coordinates systems of the objects of interest, the RM, and the operator cabin should be used to describe the computational model of an object observation.

On Fig 2 below, the point $N$ designates the zero point of the cabin coordinate system, the point $OM$ shows the position of an object of interest, $R$ is the zero point of the RM coordinate system, $C$ is the zero point of the onboard observation system (the observation cam), and $M$ is the position of the manipulator gripper.

![Figure 2. The vector configuration of observation of an object of interest.](image-url)
identify the relative coordinates of the camera and the gripper relative to the bound coordinate system of the robot respectively.

All it takes to solve target problems related to manipulating objects is to

- Find the vector $x^M_{OM}$ that identifies the position of the gripper ($M$) relative to an object of interest ($OM$);
- Solve the inverse problem that implies finding required controls for the RM. A required shift of the RM is identified by the vector $x^M_{OM}$ while taking into account that the vector $x^R_{OM}$ is known at each time moment.

The vector $x^M_{OM}$ is found while taking into account the specific functioning conditions:

1. If the coordinates of an object $x^N_{OM}$ and the robot $x^N_{RM}$ in $NX,Y,Z$ are known to an accuracy, then

$$x^R_{OM} = C^R_N (x^N_{OM} - x^N_{RM}),$$

where $C^R_N$ is the transit matrix from the stationary coordinate system to the robot’s coordinate system, in which case, the manipulator can be controlled blind, that is, with no MVS used.

2. If the coordinate identification accuracy of a manipulated object and the robot is insufficient for solving the target problem, then the onboard observation system should be used to identify the coordinates of the object relative to the camera $x^C_{OM}$. Then, the vector is found from the formula below

$$x^R_{OM} = C^R_N x^C_{OM} + x^R_C.$$

The coordinate vector components $x^C_{OM}$ are linked with the pixel coordinates on the observation plane ($u_{OM}, v_{OM}$) through the ratio

$$u_{OM} = -f \frac{x^C_{COM}}{x^C_{COM}} + u_c,$$

$$u_{OM} = -f \frac{z^C_{COM}}{x^C_{COM}} + v_c,$$

where $u_c, v_c$ are the coordinates (in pixels) of the onboard equipment center relative to the zero point of the onboard equipment coordinates; and $f$ is the focal length of the onboard equipment.

Distances to manipulated objects are found either through the base method or by using a structured light camera [15].

5. Controlling the robotic manipulator

The RM is controlled by means of an artificial neural network (ANN). Using ANNs allows solving inverse kinematics problems for the RM whenever there is a configuration for which it is hard to find analytical solutions from the MVS, and whenever numeric solutions require considerable computational resource. Many various neural network-based solutions for the control of RMs have been proposed lately [18-21]. Here, a classical teacher-trained multilayer perceptron is used. In this case, a teaching sample should include examples that contain the input vector and a required output vector. The standard back propagation algorithm has been used [17].

To simplify control, we assume that at stage one, the RM on the assembly base is rotated around the same base axes and in the same plane as the RM gripper axes, as well as the point above a selected object of interest. The next control stage (stage two) is to shift the RM through the vertical plane only.

It is the control task at stage two to select a vector of generalized coordinates for the manipulator $Q = [q_1, q_2, q_3]$ to shift the manipulator gripper from the designated point (O) to the terminal point (*). Matlab suite has been used to simulate the training process and the implementation of the neural
network algorithm to assess the applicability of the neural network-based solution for the inverse
kinetics problem (IKP) to solving the problem under study.

Developing a neural network algorithm for solving the IKP for the robotic manipulator includes a
number of stages, namely.

Stage 1. Building a training sample.

The method of choice for building a training sample is based on enumeration of generalized
coordinates of the manipulator by solving the direct kinematics problem for the coordinates of the
appropriate terminal points of the manipulator, followed by selection of optimal solutions from the
resulting set of solutions to be used to train the controlling ANN:

1. Set certain generalized coordinates of the manipulator \( Y_i = [q_1, q_2, q_3] \) within the working range of
the generalized coordinates;
2. Identify the position of the gripper (by solving the direct kinematics problem);
3. Save an example of solving the direct kinematics problem;
4. Change the generalized coordinates of the manipulator;
5. Reiterate pp 1-4 until the entire working area of the manipulator is covered;
6. Evenly, at a pace \( dR \), along the axes \( X \) and \( Y \), fill the working area of the manipulator with
points that are required to form training examples (creating teach points);
7. Set an allowable accuracy for the introduction of the gripper into a teach point (\( dS \)), and then
find an allowable vicinity for every tech point;
8. Choose a teach point \( (i) \) \( (i = 1..N) \),
where \( N \) is the number of teach points (examples of the training sample);
9. Out of the whole set examples found at pp 1-4, select those examples that shift the gripper into
the allowable vicinity of a chosen teach point \( (i) \);
10. Among the examples selected, choose the best one (the optimal) that shifts the gripper into the
allowable vicinity of the teach point \( (i) \) while causing a minimal change to the generalized coordinates
of the manipulator;
11. Save the above example into the training sample;
12. Go back to p 8 and reiterate the loop of building training sample examples pp 8-11;
13. The process is over as soon as all of the teach points have been enumerated \( (i) \).

By accomplishing the above algorithm, we obtain a set of examples for solving the direct
kinematics problem:
\[
[q_1, q_2, q_3] \implies [X, Y] \text{ at } (i = 1..N),
\]
where \( N \) is the number of training sample examples (which corresponds to the number of the teach
points) safeguarding an optimal shift of the manipulator to the point \([X, Y]\).

The examples obtained are be used in an inverse form to teach the neural network:
\[
[X, Y] \implies [q_1, q_2, q_3];
\]
where \([X, Y]\) is the input vector of the neural network, and \([q_1, q_2, q_3]\) is the required output vector
of the neural network.

Stage 2. Creating the neural network.

A three-layer ANN was used to solve the problem, which had two inputs into which the vector of
the required coordinates of the gripper was fed: \([X^*, Y^*]\), and three outputs, from which the
generalized coordinate vector was recorded \([q1^*, q2^*, q3^*]\), the generalized coordinates being required
to shift the gripper to the terminal (target) point (*) The numbers of neurons within the hidden layer
varied to achieve the best quality approximation of the desired solution surface.

Stage 3. Training and testing the neural network.

The ANN was trained by feeding vector pairs consecutively from the training sample into the
ANN. Matlab suite functions were used to conduct training.

In the meanwhile, the impact of the training sample parameters \( dS \) and \( dR \), as well as the number of
the neurons \( NH \) within the hidden layer of the ANN on the training accuracy and the time it would
take to accomplish training of the ANN were studied. A software implementation of the neural
network with various amounts of neurons within the hidden layer was used in experiments.
6. The implementation

Fig 3 and Fig 4 below show the curves that express the accuracy of shifting the robotic manipulator to a selected point, depending on a number of the above parameters.

**Figure 3.** The dependence of the manipulator error on numbers of neurons within the hidden layer.

**Figure 4.** The dependence of manipulator error on spans between training nodes.

The curves on Fig 3 above show that there exists an optimal value of the number of neurons within the hidden layer, depending on a specific organization of the manipulator and training parameters of the ANN. The fewer neurons there are, the higher is the manipulator error, which can be attributed to an insufficient approximation capacity of the ANN. Conversely, the manipulator error that occurs when there are too many neurons is attributable to “overtraining” of the ANN.

The curves on Fig 4 above illustrate that the manipulator error gets worse as spans between the training nodes $dS$, increase, and prove that there exists an optimal number of neurons within the hidden layer $MH$ for specific manipulator parameters and specific ANN training parameters.

Based on our findings, it is fair to conclude that the neural network algorithm yields a square mean error of 0.5 to 0.1 mm accurate when the robotic manipulator is being shifted to a selected point, which is acceptable for handling a chosen control panel.

7. Conclusion

Issues related to interactions between a robotic manipulator and a control panel of an equipment piece have been considered. A method for organizing coordinated function of a machine vision system and a robotic manipulator when handling controls has been presented, which includes assessing positions of objects of interest relative to the robotic manipulator; an algorithm for planning control, based on artificial neural network technology, has been proposed. The accuracy achieved meets all applicable standards for robotic manipulators handling control panels.

8. References

[1] Bajones M, Fischinger D, Weiss A, De La Puente P, Wolf A, Vincze M, Körtner T, Weninger M, Papoutsakis K, Michel D, Qammaz A, Panteleris P, Foukarakis M, Adami I, Ioannidi D, Leonidis A, Antonia M, Argyros A, Mayer P, Panek P, Eftring H and Frennert S 2019 Results of Field Trials with a Mobile Service Robot for Older Adults in 16 Private Households *J. Hum.-Robot Interact.* 9 2 Article 10

[2] Karpov V E, Malakhov D G, Moskovsky A D, Rovbo M A, Sorokoumov P S, Velichkovsky B M, Ushakov V L 2019 Architecture of a control system for a wheelchair: on the way to a multifunctional robot with neurobiological interfaces *Sovrem. technol. med.* 1

[3] Yuhang Che, Heather Culbertson, Chih-Wei Tang, Sudipto Aich, and Allison M Okamura 2018 Facilitating Human-Mobile Robot Communication via Haptic Feedback and Gesture Teleoperation *J. Hum.-Robot Interact.* 7 3 20

[4] Stankevich L A 2019 Cognitive systems of dialog communication *SAEC* 3
Alekseev I V and Mitrokhin M A 2019 Modern speech recognition methods for the construction of a voice control interface for special-purpose systems Proceedings of Higher Education Institutions Volga region. Engineering sciences 2(50)

Ivakin I A, Pechnikov A N, Potapov A A 2010 A system-informational approach to assessing the safety of spatial processes in human-machine systems Biotechnosphere 2

Minitaeva A M 2013 Organization of a human-machine interface taking into account the intellectualization of human interaction and a computer complex Software products and systems 3

Manko S V, Alexandrova R I and Diane Sekou Abdel Kader 2014 Human-machine interface for multi-agent robotic systems The Southern Federal University bulletin Engineering sciences 3(152)

Chernukhin Yu V, Butov P A 2014 Synthesis of braking quasi-fields of obstacles for the onboard system of autonomous planning of the trajectory of small-sized mobile robots IVD 2

Alekseev I A, Mitrokhin M A, Kolchugina E A 2018 Software tool for evaluating the effectiveness of speech recognition technologies Izvestiya VUZов The Volga region Technical sciences 3(47)

Ladygina I V 2017 Social and ethical problems of robotics Bulletin of VyatGU 7

He K, Zhang X, Ren S, Sun J 2016 Deep residual learning for image recognition In: Proceedings of the IEEE conference on computer vision and pattern recognition pp 770–778

Rosten E, Drummond T 2006 Machine learning for high-speed corner detection In: Proceedings of the European Conference on Computer Vision pp 430-443

Trofimenko Ya M, Vinogradova L N, Yershov E V 2019 Algorithms of image preprocessing in the face identification system in the video stream Bulletin of the Cherepovets State University 4(91)

Kim N V 2001 Image Processing and Analysis in Computer Vision Systems: Study Guide The MAI Publishing house (Moscow) p 164

Zenkevich S L and Yushchenko A S 2004 The Basics of Controlling Manipulation Robots: Textbook for Universities, 2nd ed. fix. and add The MSTU Publishing House named after N. E. Bauman (Moscow) p 480

Zhidkov V N 2013 Soft Computing in Integrated Systems of Aircraft: Study Guide The MAI Publishing house (Moscow) p 386

Oskin D A, Dyda A A and Konstantinova E A 2015 Neural network modelling of the kinematics problem for a manipulative robot Modern science-intensive technologies 12-2 254-257

Titov V V and Timofeev A V 2012 Neural network methods for planning and controlling the motion of complex mechatronic systems in extreme environments Materials of the conference Information Technologies in Management (ITU-2012) (Concern Central Research Institute Electropribor, St. Petersburg) pp 816–828

Rostov N V 2014 Analysis of Algorithms for Solving Inverse Kinematic Problems in Robot Motion Control Systems The SPbSPU Scientific and Technical Bulletin. Informatics. Telecommunications. Control 5(205) 93-99

Bol’shakov A A, Brovkova M B, Glazkov V P et al 2014 Artificial Intelligence Systems in Mechatronics: a Tutorial (The Saratov State Technical University named after Yu.A. Gagarin, EBS ASV, Saratov) p 252 ISBN 978-5-733-2690-7

Acknowledgements
This research was funded by the Russian Foundation for Basic Research (RFBR) as part of Research Project No 19-08-00613-a.