# Improvement of specialized industrial manipulator movement control

*Vladimir* Kuvshinnikov[1,*] and *Evgeny* Kovshov[1]

[1]JSC «NIKIMT-Atomstroy», Moscow, Russia

**Abstract.** This article deals with the specialized portal manipulator motion control. The key features of the system are analysed, management efficiency improvement task is considered and formulated. An approach to the manipulator trajectory formation is proposed, based on existing mobile and industrial robotics solutions including artificial neural networks.

## 1 Introduction

Among the wide variety of industrial manipulators there is a separate class of large-dimensional massive portal manipulators. When managing such manipulators, the movement strategy and trajectory selection have a great impact on performance. Specialized manipulators, in turn, have a set of additional features, which often impose a significant imprint on the entire control system (CS).

The object of the operator control is a specialized portal manipulator (PM), located above the cooling pond (CP) and designed to perform the following technological functions: search and detection of leaks in the welds of the cooling pond coating; defects areas cleansing; identified CP coating defects mending by patch method, using glue; applied patches leakage detection; patches reapplying in case of need.

Inspection and control of CP walls is exercised step by step, as CP bottom and walls are freed from the stored fuel assemblies (FA).

Plant specific features are conditioned by following technical parameters: large linear dimensions of plant and work area: more than 14 meters length, 6.5 meters width and 20 meters deep (fig. 1); large manipulator structure weight: up to 20 tons; operating temperature up to 60 Celsius degrees; hydrostatic pressure operating conditions: up to 2 atm.; high radiation background: specific absorption rate 30 000 Gy. per hour;
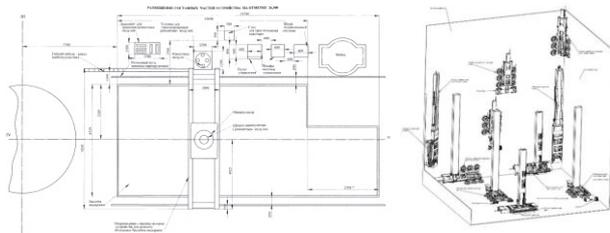


**Fig. 1.** Operating device position examples

work area scheme and dimensions changing within work space of the manipulator; replaceable highly specialized modules of the operating device (OD) of the manipulator; operator's camera surveillance restriction. The control is carried out by the operator from a specially equipped workplace using a personal computer with specialized software, as in others interactive non-destructive testing systems [1]. Portal manipulator control system (CS) is a distributed multicomponent system linked by local and controller area network. Control data exchange is performed via Ethernet, CAN, RS-485 bus interfaces and discrete sensor channels using Modbus TCP, CANOpen, FHPP and Modbus ASCII communication protocols.

During plant operating according to algorithm below shown on fig. 2, operator is responsible for OD strategic trajectory planning and motion regime. Optimal OD trajectory formation task is beyond operator strength due to human factor, man-machine interaction specifics [2] and operator view restrictions.
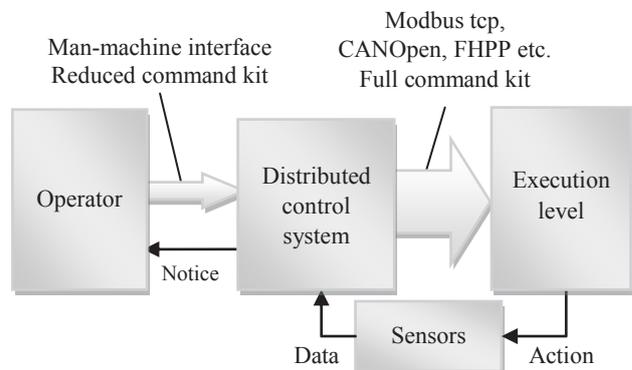


**Fig. 2.** Operator manipulator control scheme

Within the control process there will be trajectory deviations and overruns which results in extra motions to reach the end point. Described control scheme may cause following undesirable effects such as: motor winding overheat; the increased wear of PM mechanics;

---

*Corresponding author: Kuvshinnikov@ndtexpert.ru

excessive energy consumption; increased PM auxiliary time; increased OD service time.

Thus, it is required to solve a multicriteria optimization problem in order to reduce the negative effects of the PM operator management.

The problem is complicated by the fact that, unlike existing solutions [3], inadmissible manipulator configuration data is inaccessible in the considered CS. Periodic FS positioning change prevents unambiguous delimitation of work area, and OD is in no way sensitive to obstacles. In such conditions, the CS has the data obtained during manual control. Admissible manipulator configurations set is formed using executed movements data, from which optimized by the chosen criterion trajectory is selected. Such an approach is less effective than previously mentioned, but it allows to improve manipulator efficiency with no hardware changes, decreasing PM auxiliary time and service time of CP.

## 2 Structure and strategy

### 2.1. Structure of trajectory planning system

Trajectory planning is a multi-stage process of data processing, using current OD position data, work area shape and size data, and the endpoint data. It is proposed to use the structure based on three subsystems: neural network, trajectory generator and control signal generator to obtain the trajectory for auxiliary movement of the PM operating device. The developed architecture of the control system is shown in fig. 3.
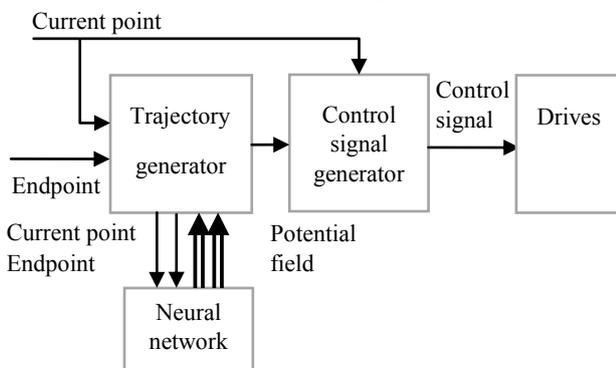
**Fig. 3.** OD auxiliary movement system architecture

There are solutions for the formation of trajectories and motion control of mobile robots based on similar architecture [4]. The differences between the presented solutions are based on the managed systems features.

On the one hand, as a control object, an industrial manipulator is simpler: it is not necessary to coordinate the trajectories of several simultaneously controlled objects, there is no need to dynamically adjust the trajectory due to volatile obstacles [5], and the simplicity of the working area makes the task of reaching an endpoint without getting into the local minima easier (fig. 4).
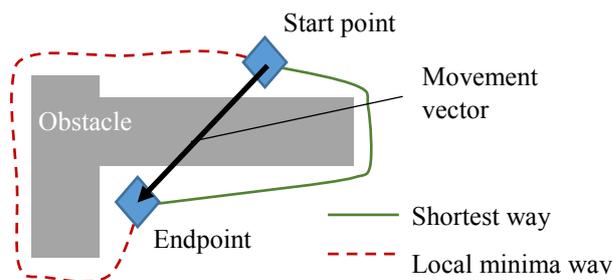
**Fig. 4.** Local minima trajectory example on surface

On the other hand, the task of PM control is complicated by the facts that it is necessary to form a three-dimensional trajectory; there is no sensory data to obtain information about the surrounding space during the OD movement; and working at greater depths under water requires the formation of a more complex potential transition cost field for each discrete configuration of the manipulator.

### 2.2 Movement planning stages

Determination of the manipulator OD position is achieved by absolute linear coordinates sensors data obtaining. The manipulator is equipped with sensors that provide information about the generalized X, Y and Z axes coordinates along the pool, across the pool and in the vertical plane, respectively. The work area of PM can be divided into several areas: the air area of free movement, the water area of free movement and the potentially inaccessible water area. The air area of free movement is above the water surface of the cooling pool. This area is distinguished by constant shape and the lowest travel costs due to the low viscous friction forces in the air compared to water. The water area of free movement is a vast volume located in a pool filled with water and limited by the speculative planes parallel to the walls and the bottom of the pool at some distance from them. This area is free from obstacles, but is completely filled with water, which forms a gradient of the transition cost along the horizontal plane depending on the depth of PM rod immersion. A potentially inaccessible water area is an area filled with water located between the free movement water area and the walls/bottom of the pool. The size of the most protruding technological structures, as well as the size of stored FA, determine the dimensions of this area. It is not possible to obtain sensory information about the shape and size of this zone for a number of reasons. It makes sense to use the recorded information on access to the inner surface of the pool, obtained during the manual control by operator, for trajectory formation.

Taking into account the mentioned peculiarities, the task of the movement planning should be decomposed into planning levels: strategic, tactical, detailed. This approach is used in mobile robotics as well [6] and is suitable for PM displacement planning. The shape of the work area either provides access to the part of the pool surface directly from the free movement area, or no access at all, which simplifies the potential paths to reach

any available point of the CP surface. The main objective of the strategic planning level is to form a set of three-dimensional direction vectors that define the sequence of target endpoints movements. Tactical level planning involves finding the shortest path through a neural map, generated considering the weighting functions in terms of the specified discretization parameter. The detailed planning level is designed to smooth the movement trajectory.

## 2.2 Trajectory formation strategy

PM kinematic scheme includes three controlled translational joints and one unmanaged rotary joint. Unmanaged rotary joint intended for mounting specific modules in different orientations. It affects the size of the OD and determines the operating point coordinates.

If we discretize the configuration space of the manipulator with resolution $N$, then the coordinate in each of the joints $j_1$, $j_2$ and $j_3$ can take discrete values from the range $\mathbf{q}_{xj}(xj \in \{1, \ldots, N\})$, where $q_{1j}$ and $q_{Nj}$ are the lower and the upper constructive linear motion constraints of the joints. Let us name the set of all OD configurations, conditioned by a set of available modules and orientations, as a set of discrete options $O$. Then the coordinate in the joint $j_4$ can take on a values of $\mathbf{q}_{o4}(o_4 \in \{1,\ldots,O\})$. Then we can represent a discrete model of the manipulator configuration space as a set of $N^3 \cdot O$ vectors

$$\mathrm{DC}=\{\mathbf{q}_a|1\ldots N^3 \cdot O\}, \qquad (1)$$

where $\mathbf{q}_a = [q_{x1} \ q_{x2} \ q_{x3} \ q_{o4}]^{\mathrm{T}}$ – discrete manipulator configuration $xj \in \{1, \ldots, N\}$ where $j = 1, \ldots, 3$; $o_4 \in \{1, \ldots, O\}$; $a$ – a one-dimensional index, taking a value of $a = N^2 \cdot O \cdot (x_1 - 1) + N \cdot O \cdot (x_2 - 1) + O \cdot (x_3 - 1) + o_4$.

Due to the fact that the configuration of the $j_4$ joint is manually changeable in a dedicated area, the only joints to be controlled automatically and used for trajectory formation are $j_1$, $j_2$ and $j_3$. At the same time, taking into account the difference in the CP linear dimensions, in practice we better choose separate discretization value for each joint $N_1$, $N_2$ and $N_3$.

The system has no information about the set of obstacles $B = \{B_1, B_2, \ldots, B_m\}$, making it difficult to determine a set of free from a collision configurations, $\mathrm{DC}_f = \{\mathbf{q}_a \in \mathrm{DC} \mid M(\mathbf{q}_a) \cap B = \varnothing\}$, denoting manipulator in configuration $\mathbf{q}_a$ as $M(\mathbf{q}_a)$ [7]. Instead, a set of safe configurations $\mathrm{DC}_s$ is defined as a known subset of $\mathrm{DC}_f \supseteq \mathrm{DC}_s$. A set of dangerous configurations consisting of the configurations that do not belong to $\mathrm{DC}_s$, including $B$ and $\mathrm{DC}_f \cap \mathrm{DC}_s$, is

$$\mathrm{DC}_d = B \cup (\mathrm{DC}_f \cap \mathrm{DC}_s); \qquad (2)$$

Secure configurations set consists of the configurations sets corresponding to air and water areas of free movement $(\mathrm{DC}_f \cap \mathrm{DC}_{fw})$, and configurations of potentially unavailable water area, which are considered to be safe during the operator manual actions monitoring $(\mathrm{DC}_m)$

$$\mathrm{DC}_s = \mathrm{DC}_{fa} \cup \mathrm{DC}_{fw} \cup \mathrm{DC}_m. \qquad (3)$$

Key trajectory selection strategy implies avoiding potentially dangerous manipulator configurations and only bringing the manipulator into safe configurations.

# 3 Mapping and trajectory generation

## 3.1 Three-dimensional area representation

To simplify the presentation, to reduce the amount of stored data and the access time, certainly safe areas are specified by coordinate ranges. A potentially inaccessible water area is represented as a set of corresponding PM configurations. At the same time, to decrease the amount of filtering area cells calculations, adjacent configurations are initially combined into three-dimensional groups. Initially all the configurations in potentially inaccessible water area are considered dangerous. The size of the configurations groups is selected as a multiple of the linear size of the base group. Sample size for a cubic configuration group:

$$N_g = n_g{}^3; n_g = l^i, \qquad (4)$$

where $N_g$ – cubic group full size, $n_g$ – cubic group linear size, $l$ – base group linear size, $i$ – group scale. At $i = 0$, a single configuration of the manipulator is considered. Configurations group is a three-dimensional convolution of the adjacent configurations, which stores information about the corresponding work area cell dangerousness. The convolution dangerousness is determined by the coefficient $d \in [0, \ldots, 1]$. Groups with $d = 0$ are completely safe and with $d = 1$ are completely dangerous. If the group contains only completely safe or completely dangerous groups of lower scale, it is also considered completely safe or dangerous correspondingly. The safety check of a cell comes down to sequential groups polling in descending order of scale to obtain sufficient or comprehensive information.

## 3.2 Trajectory generator

The manipulator OD motion trajectory formation is made by a sequential rise from the initial cell of the neural map to the final one in the direction of the maximum gradient. Initial cell is a cell, containing starting PM configuration. Final cell is a cell, containing endpoint configuration. The algorithm of trajectory generation is shown in fig. 5.
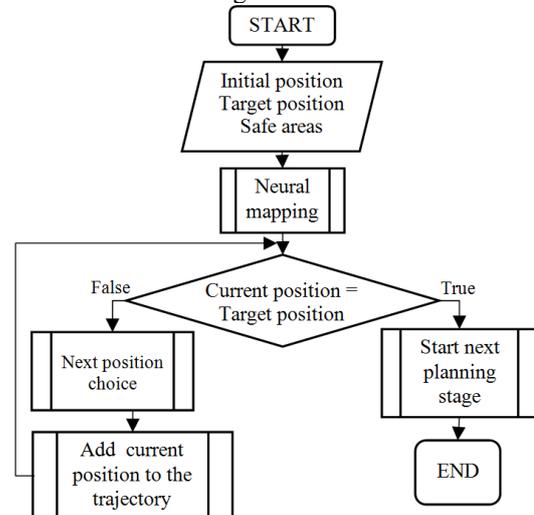


**Fig. 5** Algorithm of trajectory formation

## 3.3 Neural mapping

A special representation of the manipulator configurations set - a neural map – is used for trajectory formation. Each group of configurations stands for a cell of work area and corresponds to one neuron. The choice of the neural network type is determined by the problem: filtering the work area cells by the cost of transition from each cell to adjacent cells, considering the weight function.

The Hopfield network is one type of network with filtering ability. We use neural network to filter configuration groups. The scheme of the elementary network neuron is shown in fig. 6.
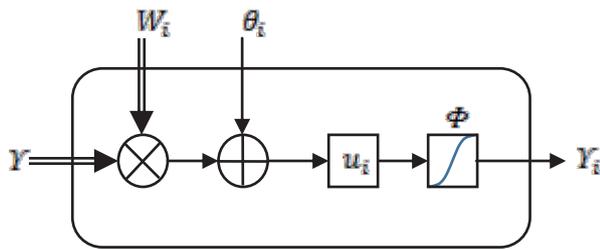


**Fig. 6** Elementary nonlinear processing neuron unit

Neuron $i$ is described by input vector $\mathbf{Y}$, weight vector $\mathbf{W}_i$, displacement vector $\boldsymbol{\theta}_i$, that form a sum network signal

$$u_i = W_i \times Y + \theta_i = \sum_{j=1}^{n} w_{ij} v_j + \theta_i . \quad (5)$$

The sum signal is processed by a nonlinear activation function $\Phi$ and formed the output signal of the neuron $\mathbf{Y}_i$. Hyperbolic tangent, sigmoidal function and others can be used as activation function.

Neural mapping requires determining the following parameters: the order of the considered configuration groups; the size of the considered area; the type of weight function, and the topology of the adjacent cells relationships. Considered configuration groups scale determines the linear dimensions of the working area cell, taken as a single position of the manipulator, which corresponds to one neuron. This parameter can be called the neural map resolution. The lower the scale, the more neurons will include a neural map, which will increase the trajectory sampling and allow better dangerous CP zone enveloping. Decreasing the scale increases the number of neural map neurons by a multiple of $l^3$, where $l$ is the base configuration group linear size. The coverage of the map is determined by the initial and target trajectory points relative location.

Working area shape is simple and there is no need to move far beyond the parallelepiped, which includes movement direction vector to reach the endpoint. The map includes a sufficient part of the free movement zone to plot the trajectory, with a reserve of height (coordinate $z$) to take the gradient of the linear movement cost into account also. The size of that reserve depends on the aquatic environment viscous friction.

Piecewise given function is chosen as a weight function of the $i$-th and $j$-th neuron connection

$$w_{ij} = \begin{cases} 0 & \rho(q_i, q_j) \cdot k_f(z, \phi) = 0 \\ f(\rho(q_i, q_j)) & 0 < \rho(q_i, q_j) \cdot k_f(z, \phi) < r \\ 0 & r < \rho(q_i, q_j) \cdot k_f(z, \phi) \end{cases} \quad (6)$$

where $\rho(\mathbf{q}_i, \mathbf{q}_j)$ – Euclidean distance between configurations, corresponding to i and j neurons; $k_f(z, \varphi)$ – function that counts how viscous friction forces affect the cost of transition at a given depth in a given direction; $z$ – chosen configuration's depth; $\varphi$ – the polar angle between the $x$-axis and the direction of travel from $q_i$ to $q_j$ measured in XOY plane; $f(\rho(\mathbf{q}_i, \mathbf{q}_j))$ – decreasing function of the form $f(x)=1/x$; $r$ – the radius of the neutron region i.e. how far each neuron affects other neurons through weight functions [8].

Neural mapping may base on different topologies of connections between neurons. Hexagon (fig. 7a) and rectangular (fig. 7b) are the most common connections topologies for the plane. It is convenient to choose a parallelepiped form connection (fig. 7c) for a three-dimensional task. Each neuron has 6, 18 or 26 adjacent neurons connection depending on the selected neural region radius. This is because the distances between the nodes are different: along the edge ($a$), along the face diagonals ($a\sqrt{2}$) and along the three-dimensional diagonal ($a\sqrt{3}$).

After the scaling with the weight coefficients vector $\mathbf{W}_i$ operation, the $i$-th neuron bias vector $\boldsymbol{\theta}_i$ is applied.



a)                                    b)
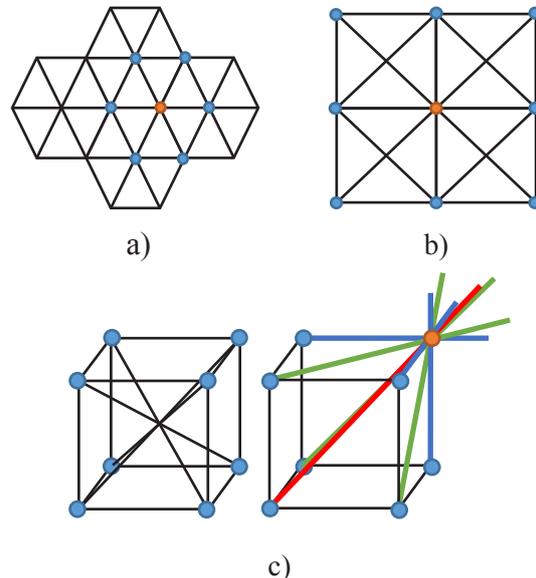


c)

**Fig. 7** Neuron connections topologies

The bias value is determined by the neural card input:

$$\theta_i = \begin{cases} +\infty & i-\text{th neuron is in target group} \\ -\infty & i-\text{th neuron is in group with } d > d_{max} \\ 0 & \text{other cases} \end{cases}, \quad (7)$$

where $d$ and $d_{max}$ – $i$-th neuron corresponding configurations group danger coefficient and maximum allowed coefficient value.

The bias vector makes the target neuron always maximally activated, and neurons corresponding to potentially dangerous groups of manipulator configurations – always maximally deactivated. In other cases, it is possible to set the bias value as a function of parameter *d*.

Other neurons, which are not directly defined by external data, can be activated in several ways. One of the ways is spreading the activation from target neuron, which corresponds to the target configuration. Activation of neurons is distributed in all directions using the simultaneous activation method until either the activation signal reaches the initial neuron, which corresponds to the initial position of the manipulator, or all neurons are activated. Another method involves consecutive neurons activation with rasters, for example, cross-activation [8]. Activation does not extend through neurons with a bias $\theta_i = -\infty$ (neurons of potentially dangerous configurations), which allows to reach the target neuron by just climbing in maximum gradient direction starting from the initial neuron.

## 4 Results and conclusions

The trajectory generation method with neural mapping is expanded from two-dimensional into three-dimensional representation. Modern computers processing power has multiplied compared to previously used computers, which has been successfully used for trajectory calculations in two dimensions. In addition, modern computers support parallel processing of information at many levels, starting with the hardware. Special software and hardware-software tools have been developed to distribute computing power for parallel processing, such as CUDA [9, 10] and OpenCL [11]. No wonder that these solutions were previously used for the volumetric calculations that are associated with learning [12, 13], and activation of neural networks [14]. Thus, the possibilities of multidirectional neural network activation simultaneous calculations [15] are significantly increased. For better trajectory formation representation a computer simulation with flexible parameters is proposed. Currently, the NVIDIA GPU Cloud (NGC) provides an easy access to the full catalogue of software for deep learning and high performance computing, optimized for the GPU [16].

## References

1. A. Kosach, E. Kovshov, ITM Web of Conferences, **10**, 01005 (2017)

2. E. Borisenko, P. Martynov, E. Kovshov, Collection of scientific papers on the materials of the international scientific-practical conference, **2** (2009)

3. V. Kuvshinnikov, E. Kovshov, "Rosatom" State Corporation innovation management unit. URL: http://innov-rosatom.ru

4. O. Darintsev, A. Migranov, Artificial intelligence, **3** (2009)

5. O. Darintsev, A. Migranov, Bulletin of RAS, **3** (2007)

6. A. Bolshakov, M. Stepanov, A. Stepanov, Bulletin of SSTU, **4** (2010)

7. L. Lobareva, M. Kozhevnikov, Bulletin of Polotsk state University, **11** (2017)

8. G. Michail, The Center for Advanced Computer Studies University of Southwestern Louisiana (1999).

9. CUDA Toolkit Documentation v. 9.1.85. URL: https://docs.nvidia.com/cuda/

10. NVIDIA Accelerated Computing. Deep learning. URL: https://developer.nvidia.com/deep-learning

11. The open standard for parallel programming of heterogeneous systems. URL: https://www.khronos.org/opencl/

12. E. Matkova, E. Kovshov, Informational technologies, **12** (2008)

13. J. Li, Q. Liu, 8th International Symposium on HEART, (2017)

14. S. Zoican, R. Zoican, D. Galatchi, Advanced Technologies, Systems and Services in Telecommunications (TELSIKS) (2017)

15. R. Glasius, A. Komoda, S. Gielen, Neural Networks, **8** (1995)

16. NVIDIA GPU Cloud. URL: https://www.nvidia.ru/gpu-cloud/