Response Time Estimation in Robotics Using Hierarchical Performance Modeling Approach

Ahmed Alsheikhy[1*], Muhannad Almutiry[1]

[1] Electrical Engineering Dept. Faculty of Engineering, Northern Border University, Arar, Saudi Arabia
* Corresponding author's e-mail: aalsheikhy@nbu.edu.sa

ABSTRACT

The Multi-disciplinary domain of robotics has become instrumental in advancements in many fields such as medical, military and industrial automation. They are used to ensure reliability, stability and precision. Particularly, they are preferred to be used in an environment where uncertainties and disturbances occur. Measuring or estimating a performance metric such as response time or throughput under severe circumstances is crucial since designers want to know the behavior of robots and their reaction when a constrained condition is applied. The present research applies the framework proposed earlier on a robotic manipulator to predict a performance metric, this paper considers only the execution time which is also known as the response time. In addition, a comparison is performed between predicted and actual values. All values herein are average ones. A simulation system in MATLAB is used to find out the differences between the predicted and actual values. Results demonstrate that the proposed framework is capable of finding the needed metric or metrics and determining/spotting bottleneck(s) inside any system under consideration as it has been verified by several experiments.

Keywords: robotics, response time, automation, framework.

INTRODUCTION

Thanks to the technological advances, use of robots has become very popular in many fields such as industrial, medical, military and space exploration. This has consequently created competitive requirements among various competitors. Robotics has reduced production time in a noticeable manner while still ensuring the Quality of Service (QoS). Robots can perform several tasks such as assembling, welding, painting, palletizing, sorting etc. [1]. In some scenarios, the use of robots is needed in an environment where human intervention may lead to an injury or a harm. A robotic arm can be considered as the most common configuration which is deployed in the industrial applications requiring stringent requirements of quality and precision. Owing to multi-dimensional and multi-disciplinary nature of robots, they offer fast, accurate and reliable operational performance. In recent decades, robotics has reshaped learning, prediction, perception and manipulation activities.

Various diversified application domains such as space jet, medical systems and e-commerce systems require performance modeling and timing analysis to evaluate their response time before using them and/or releasing them to the public [2]. The importance of the performance modeling method for the timing analysis is motivated by reducing time, cost and delays since there is no need to wait until the stages of component integration and final implementation. It also produces better results when comparing with 'fix-it-later' approach.

The performance metric considered in the present research is the response time which refers to the time required for completion of a task. Predicting amount of throughput, energy dissipation and accuracy can also be easily measured with some adjustments. To measure the execution time successfully, a Hierarchical Performance Modeling (HPM) is used in this research to follow an analytical modeling approach. In this approach, any job to be executed undergoes through four layers that can be seen as levels detailed below:

- System level.
- Task level.
- Module level.
- Operation level.

For more information about these levels, interested readers are referred to [2, 3]. Our aim here is on the timing analysis to ensure that any product functions well within the predefined time. HPM method predicts an average response time needed which is defined as the time interval between applying an input until the associated outputs are obtained [2, 4]. The timing analysis is highly required to compute the delays occurring between computation and communication processes with several software running on different platforms.

The contribution of this research is to explore HPM approach to predict the execution time in the optimal design reported in [5]. The rest of the paper is formed as follows: presenting the related work works for evaluating performance metrics on different robotic systems is done in Section II, followed by a detailed discussion on the proposed algorithm in Section III. Sections IV illustrates the simulation experiments that have been conducted to compute the performance metric. Finally, Section V concludes the paper.

RELATED WORK

Robotics community has presented several metrics to characterize the performance of the manipulators. Yan et al. in [6] proposed a collection of metrics to provide a comparison study between several algorithms to explore multi-robot collaborative systems. Their study identified parameters that can potentially influence robot performance such as throughput, delay etc. Simulation experiments were conducted in Robot Operating System (ROS) to uncouple the control software between several drivers and robot body itself. This algorithm did not consider the time needed to execute a task. It is very crucial to evaluate this metric since many robots are being used in critical fields where fast operations and real-time response actions are mandatorily required.

In [7], Quispe et al. provided guidelines to design a useful testing protocol to evaluate manipulation performance. Two different measures are provided to apply a fair comparison of grasping and manipulation approaches i.e. (i) Inclusion of inherent stochasticity for grasping benchmarks and (ii) Running experiments in a loop without human intervention which helped to capture robot's ability to repeatedly deal stochasticity inherent for execution of several tasks. They benchmarked several tasks into a taxonomy of manipulation along with their corresponding metrics. However, this work did not consider the response time. Readers can refer to [7] for more information about methodology to benchmark manipulation tasks.

Another research reported in [8] presents an algorithm based on the selection of the most informative node during the backtracking phase to evaluate the exploration complexity of different structure environments. The aim was to conduct a fair comparison among several related approaches. Furthermore, an evaluation index was developed to estimate overall performance of any exploration method as a single number. They focused on the distance travelled D, exploration time T, total number of created nodes in the tree N and the percentage of total area covered after the home step C.

In these studies, the response time of any performed task was not considered. In contrast, the algorithm applied in the present paper evaluates the response time by finding average value based on a framework which was developed and discussed in [2, 4]. The time considered here between the application of input and occurrence of output can also be referred as the average Execution Time (ET).

THE PROPOSED ALGORITHM

In order to estimate the ET, the framework developed in [2] has been used. This framework is suitable to be applied on any system due to several reasons such as:

- It is a generic approach which can be used and applied on any system regardless of its complexity.
- Several performance metrics can be evaluated like throughput, execution time and energy/power consumption. Furthermore, it can detect or spot any bottlenecks inside the system.
- It combines various system components related to hardware and software at different levels to estimate the desired performance metric.
- It considers the time delay required for communication between different components and the computation time.
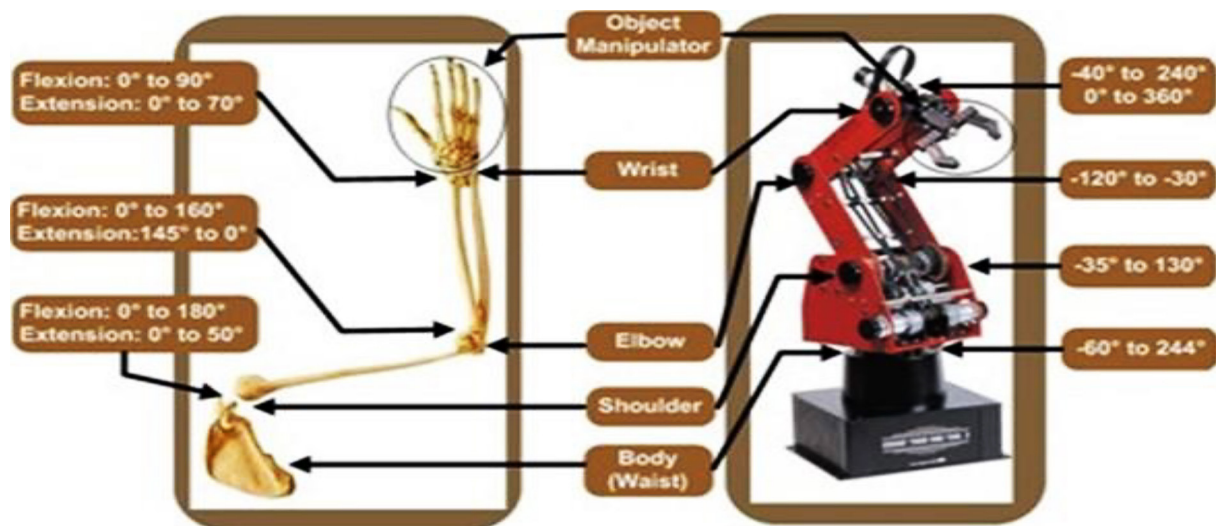
Fig. 1. AUTAREP robotic and human arms

The framework has been applied on a 6-DOF robotic manipulator of the AUTonomous Articulated Robotic Educational Platform (AUTAREP) reported in [9]. Figure 1 illustrates the comparison between human arm and the robotic manipulator.

Table 1 shows AUTAREP specifications for several parameters such as: Kinematics, physical, sensing and performance.

System designers are required to know the relationship between motion and forces to be able to model and develop an optimal robotic system. Modeling is considered as a vital and crucial factor in investigating this relationship. In this paper, we have considered on kinematic and dynamic models for deriving the equation to evaluate the average ET for the intended robotic system.

Kinematic Approach

Defining the time-based phases under system circumstances for several parameters such as location, orientation, acceleration and velocity is done by this approach. It engages derivation of forward and inverse kinematics. The former can be computed using Denavit-Hartenberg parameters (DH). The frame assignment of the robot joint is depicted in Figure 2.

The forward kinematics of AUTAREP manipulator is derived in [11] while the inverse kinematics model is presented in [10].

Dynamic Approach

This approach characterizes the relationship among system time-based parameters with applied forces and torques that cause the changes in the system under several circumstances. Euler-Lagrange formulation relying on energy-based technique has been used to derive the dynamic model of the robotic arm.

To estimate the average ET, the components of the robotic system have been distributed using the generic Finite State Machine (FSM) developed in

Table 1. AUTAREP parameters and their specifications [10]

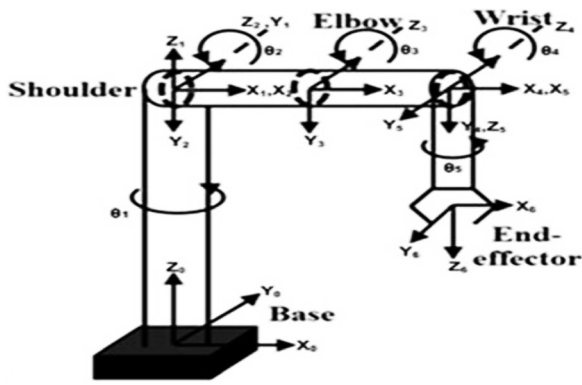| Criterion	Requirements	Explanation
Kinematics	No. of concourses	5
	No. of DOF	6
	ROM	Wrist pitch: 260°
		Wrist roll: 360°
		Elbow: 172°
		Shoulder: 90°
		Waist: 310°
Physical	Locomotion	Articulated links
	Actuation	6 DC servo motors
	Weight	33 Kg
	Measurements	Base-220x180(H) mm
		Arm length - 220+220 mm
Sensing	Perception	Camera (Logitech)
	Strength	FSR attached at gripper
	Location	Optical encoders
	Precision	1.5 mm
	Repeatability	1 mm
Performance	Movement speed	100 mm/s (max.)
	Payload	1 kg
	Action radius	580 mm (largest)
	Stall current	2.2 A (for each motor)
	Applied voltage	24 V

Fig. 2. Robot frame assignment

[2] and then HPM approach has been applied after converting the FSM into a Markovian model. From the framework point of view, any type of movement can be classified based on the Generic FSM (GFSM), developed in [2]. Figure 3 shows the block diagram for the developed framework. It is evident from the figure that the developed framework has two components which can be seen as Hierarchical (HGFSM) and HPM.

The description of various HPM levels is illustrated in Figure 4 with the system level residing at the top level. Figure 5 represents the high level structure of the HGFSM [20–25].
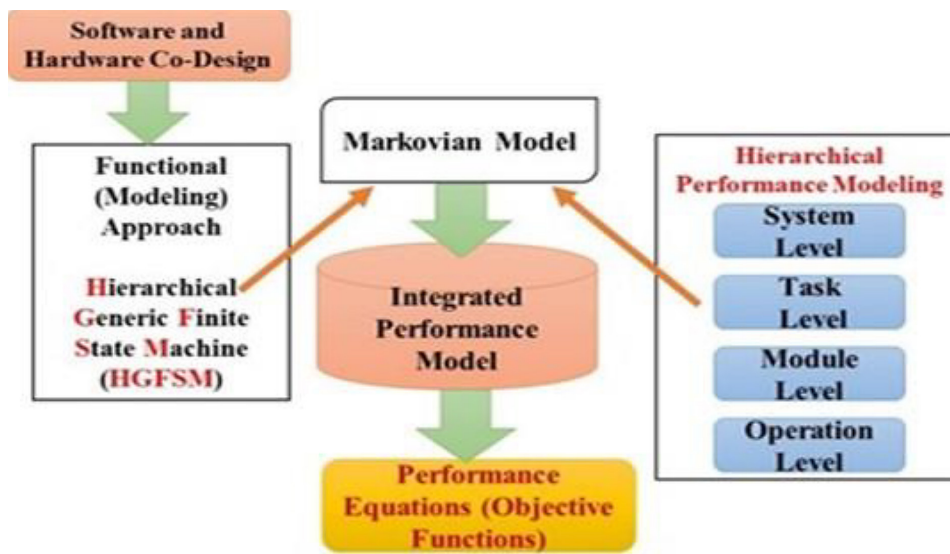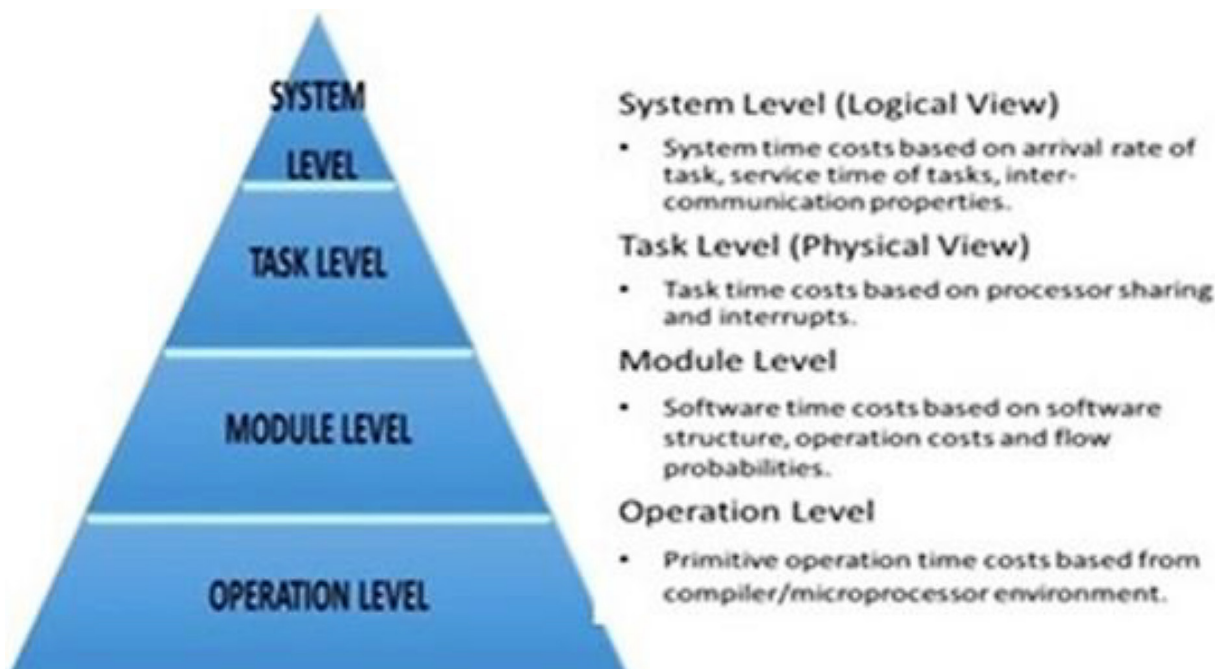

Fig. 3. Proposed framework
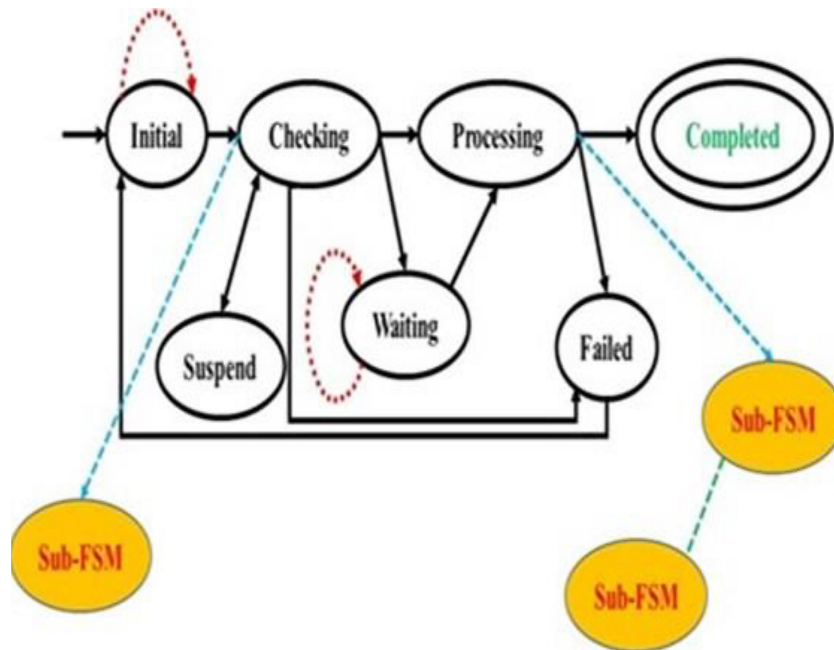

Fig. 4. HPM stack layers

Fig. 5. High level diagram of HGFSM

Fig. 6. Stack layers of HGFSM

The HGFSM can be illustrated as shown in Figure 6 for simplicity and easy of understanding.

From Figures 3 and 5, we can easily derive a relationship to compute the Average Response Execution Time (ARET) as follows:

$$ARET = (1 * C_{initial}) +$$
$$+[(1 + e_4) * (C_{check} + C_{test})] +$$
$$+[(e_9 - e_8) * (C_{wait} + C_{test})] + \quad (1)$$
$$+[(e_{11} + 1) * C_{decision}] +$$
$$+[e_{11} * (C_{exe} + C_{test})]$$

In the equation (1), every variable C is combined with its flow variable. Every variable refers to a cost when passing through a route or a track from a source node to a destination node in a Control Flow Graph (CFG). In order to identify the operations taking place in each state of the HGFSM, the structure of the system under consideration needs to be thoroughly analyzed. The details of obtaining the desired performance equations (i.e. objective functions) are reported in [2, 4].

RESULTS AND DISCUSSION

Simulation experiments have been conducted using MATLAB R2017b installed on a workstation which has Windows 10 Home Edition as the operating system. It is 64-bit based-processor system type and it runs on a 2.4 I7-8700T/8th Gen. CPU with 16GB RAM.

In order to estimate ARET, there is an equal probability of executing if-else branch, so $p = q = 0.5$. In each state in Figure 5, several activities/operations take place which are detailed in [2]. $C_{initial} = 1.2$ ms and $C_{check} = 3.8$ ms correspond to the initial state and checking state

respectively. The time required to execute if-statement is found to be very small and can be neglected. So, C_{test} = 0 ms. The Suspend state is not included in (1) because all tasks have been executed successfully and no task was routed to this state. C_{wait} has been taken as 0 ms since the system is considered to be critical as any failure leads to catastrophic results. The processing state includes both decision and execution steps. $C_{decision}$ is estimated to be 0.02 ms and C_{exe} is around 2.6 ms.

To find the value of all flow variables in (1), probabilistic relationships were used. Calculating the number of travels in every phase/state is crucial to determine the Estimated Average Response Time (*EART*), which is calculated as:

$$EART = \sum(V_i * C_i) \qquad (2)$$

where: *i* refers to the total number of phases/states in the HGFSM. i.e. $i = 1, 2,\ldots, 6$. [*V*] is calculated as follows:

$$[V] = [I - T]^{-1} \qquad (3)$$

where: *I* represents the identity matrix and *T* is the transition probability matrix corresponding to all states in Figure 5 and can be calculated as follows:

$$P_{ij} = K_{ij} / N \qquad (4)$$

where: K_{ij} represents the number of jobs or instructions initiating from the source phase towards the destination phase, *N* denotes the total number of jobs which are in the source phase. The probability transition matrix *T* can be depicted from HGFSM in Figure 6 as illustrated in Table 2.

Considering the proposed system in [5], every task is sent from its source to destination so P_{ij} = 1. By substituting this value in (3), we can find number of visits [*V*] which is used later to compute *EART* from (2). Hence, *EART* = 9.53 ms. Now, Actual Average Response Time (*AART*) is important to be determined so that the percentage of error between *AART* and *EART* can be calculated as:
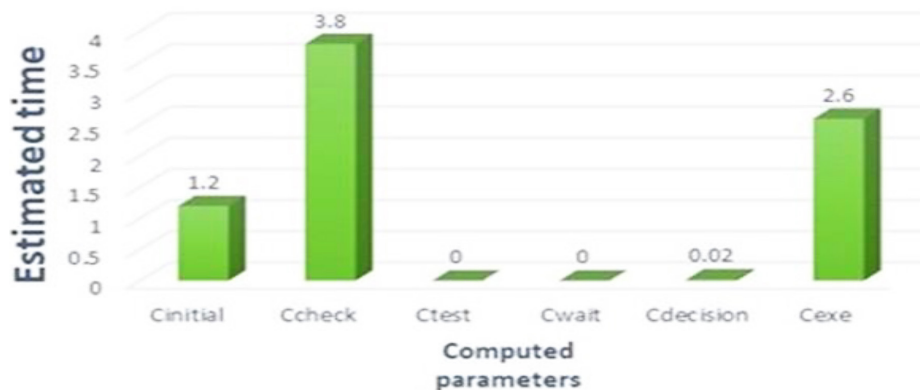
$$E\% = ABS\left(\frac{AART - EART}{AART}\right) * 100 \qquad (5)$$

Figure 7 displays the average value of *EART* for all calculated parameters after running the simulation for more than 1000 times.

AART is found to be 8.48 ms Thus, *E%* = 12.3%. Figure 8 shows the average values of *AART* and *EART*.

Table 2. Propability Transition Matrix

From-To	Initial₁	Checking₂	Waiting₃	Execution₄	Failed₅	Completed₆
Initial$_1$	0	$P_{12} = 1$	0	0	0	0
Checking$_2$	0	0	$P_{23} = k_{23}/N_2$	$P_{24} = k_{24}/N_2$	$P_{25} = k_{25}/N_2$	0
Waiting$_3$	0	0	$P_{33} = k_{33}/N_3$	$P_{34} = k_{34}/N_3$	0	0
Execution$_4$	0	0	0	0	$P_{45} = k_{45}/N_4$	$P_{46} = k_{46}/N_4$
Failed$_5$	0	$P_{52} = 1$	0	0	0	0
Completed$_6$	0	0	0	0	0	0

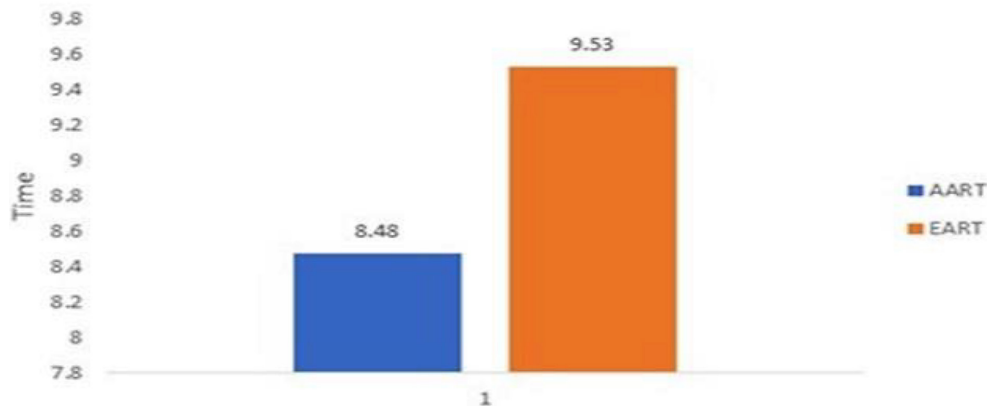**Fig. 7.** The average estimated values of time in ms

Fig. 8. Average time values of AART and EART

CONCLUSIONS

This research extends the earlier work of the authors by deriving objective functions to determine the estimated average response time. The proposed method is easy to use since it does not require any special tool and simply needs a pencil and a paper. The percentage of error between actual and estimated response time has been found to be around 12%. This figure seems to be acceptable as the error does not exceed 15%. In future, it is anticipated to minimize the EART using multiprocessors and a scheduling technique.

Acknowledgement

REFERENCES

1. Iqbal J., Islam R.U., Abbas S.Z., Khan A.A., Ajwad S.A. Automating industrial tasks through mechatronic systems–A review of robotics in industrial perspective Tehnički vjesnik. 2016;23:917–924.

2. Alsheikhy A., Han S., Ammar R. Hierarchical performance modeling of embedded systems in 2015 IEEE Symposium on Computers and Communication (ISCC), 2015, 936–942.

3. Smarkusky D., Ammar R., Antonios I., Sholl H. Hierarchical performance modeling for distributed system architectures in Proceedings ISCC 2000. Fifth IEEE Symposium on Computers and Communications, 2000, 659–664.

4. Alsheikhy A., Han S., Ammar R. Delay and power consumption estimation in embedded systems using hierarchical performance modeling in 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). 2015, 34–39.

5. Ajwad S.A., Iqbal J., Islam R.U., Alsheikhy A., Almeshal A., Mehmood A. Optimal and robust control of multi DOF robotic manipulator: Design and hardware realization Cybernetics and Systems. 2018;49:77–93.

6. Yan Z., Fabresse L., Laval J., Bouraqadi N. Metrics for performance benchmarking of multi-robot exploration in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, 3407–3414.

7. Quispe A., Ben-Amor H., Christensen H. A taxonomy of benchmark tasks for bimanual manipulators in Proc. Int. Symp. Robotics Research, 2015, 1–10.

8. El-Hussieny H., Assal S.F., Abdellatif M. Robotic exploration: new heuristic backtracking algorithm, performance evaluation and complexity metric International Journal of Advanced Robotic Systems. 2015;12:33.

9. Iqbal U., Samad A., Nissa Z., Iqbal J. Embedded control system for AUTAREP-A novel autonomous articulated robotic educational platform Tehnički vjesnik-Technical Gazette. 2014;21:1255–1261.

10. Manzoor S., Islam R.U., Khalid A., Samad A., Iqbal J. An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation Robotics and Computer-Integrated Manufacturing. 2014;30:351–362.

11. Iqbal J. Modern control laws for an articulated robotic arm Engineering, Technology & Applied Science Research. 2019;9:4057–4061.