

(RESEARCH ARTICLE)



Simulation of a robot arm that can supplant the dynamic activities of human arm

Okechukwu Stanley Ikwunze ¹ and Victor Ikedichi Okparaku ^{2,*}

¹ Department of Electrical/Electronic Engineering, Abia State Polytechnic, Aba, Abia State, Nigeria.

² Department of Science Laboratory Technology/Physics, Uma Ukpai Polytechnic, Asaga, Ohafia, Abia State, Nigeria.

GSC Advanced Engineering and Technology, 2022, 04(01), 072–078

Publication history: Received on 24 August 2022; revised on 27 September 2022; accepted on 30 September 2022

Article DOI: <https://doi.org/10.30574/gsaet.2022.4.1.0053>

Abstract

The research was tenaciously conducted to model and simulate a robot arm using MATLAB Simulink software. The research activities included modification of a robot arm that mimics the controlling performance of a real human arm in MATLAB and the controlling effects of each joint as per the angle of rotation of the robot arm. The controlling moment (forces) of the robot arm was mathematically modeled. Euler-Lagrange equation of motion was applied to determine vector of control moments (forces). During the simulation process, the first to achieve was to model equations of motion in MATLAB symbolically and then (further) to generate function blocks directly in Simulink followed by use of Sims cape toolbox to model the robot arm similar to 3D modeling. The robot arm end effector followed certain reference trajectory as computed torque chosen in feed forward plus PD (Proportional Derivative) controller in feedback division. Effective control of the robot arm was ensured by connecting the robot arm to a personal computer (PC) using a microcontroller. The complete control process was accomplished through two divisions of CUBLOC microcontroller program (to write code of interference between PC and the robot arm) and MATLAB Program (containing serial communication code and the graphical user interface). The stages employed in the activities of the complete system functions consist of kinematics, trajectory planning and a controller. The output was tested and it showed that the simulated robot arm performed the dynamic activities, particularly the control, of a real human arm in the software environment.

Keywords: Robot Arm; Dynamic Activities; Rotation Angle; Angular Velocity; Human Arm; Modeling and Simulation; Control

1. Introduction

Invention of robot is predominantly a huge area in science and engineering that will turn the world into an easy task execution system. The history of robot has its roots as far back as ancient myths and legends [9]. Its study is penetrating deeply into solving problem of doing hard tasks human finds difficult. A human arm which does the task is a combination of the scapula, humerus, ulna & radius, and carpus and the relative displacement of them determine the movement of an arm [3]. Today, robots are making efficient influence on various aspects of modern activities. In the future, robots will be as pervasive and personal as today's personal computers. The dream of creating machines that is skilled and intelligent has been part of humanity from the beginning and the dream is becoming part of our world's striking reality now [7].

However, robots in any unstructured environment may encounter objects which it has only very little *a priori* experience or knowledge. In such a case, successful grasping requires a sophisticated perception, planning and control [5]. The problem of grasping of unforeseen objects in unstructured environments remains problem unsolved. In addition to assuming full autonomy of robots, some researchers also had investigated the amount and type of human

* Corresponding author: Okparaku Victor Ikedichi
Department of Science Laboratory Technology/Physics, Uma Ukpai Polytechnic, Asaga, Ohafia, Abia State, Nigeria.

supervision required to enable a robot grasp unforeseen objects in an unstructured environment. The approach is of two stages: in the first stage, the human supervisor defines a bounding box around the target object by means of the man-machine interface. For the second stage, the robotic hand configuration, orientation, and location are determined by the robot's vision system which possibly accomplishes the grasping task based completely on a three dimensional (3D) vision-based object features as extracted by the robot's sensors [1]. The work on grasping studied here focuses on the automation of the second stage concerned with determining effective grasps posture. Marco and co presented graphic C++ with the application for three-dimension replication of industrial robots in sequence [6]. The program commenced particularly from robot forward kinematics which was reasonably considered and its functions implementation in the source code assisted in the ability to calculate the position and orientation of each robot joint and the position and orientation of the robot gripper. The OpenGL functions enabled the use to draw and replicate the three dimensional kinematic scheme of the robot.

The gripping of objects requires gripping force. The gripping force is a contact force per unit gripping area to clamp an object. There are various kinds of properties of object to be gripped such as the thin wall, soft and the fragility [4, 8]. The computer control system would be able to use the suitable gripping force to hold the objects and avoid deformations [2]. It is the postures belonging to a grasp type people usually choose unconsciously from the taxonomy to grab a particular object that are optimized for the task and for the object in a lifelong learning process.

This work is intended to simulation a robot arm (StanR2022) that will be able to supplant the dynamic, especially the trajectory control, of activities of human arm. The control will be specifically the controlling of angle of rotation of the joints resulting to defined angular velocity.

2. The Angle of Rotation (Joints Angle) Representation of the Simulated Robot Arm

This robot is made up of five (5) joints as represented by the diagram below in fig.1a while fig.1b simply identifies the joints in fig.1a for proper naming and understanding. By the labeling, joint 1, joint 2, joint 3 and joint 4 and joint 5 in fig.1a represent the base, shoulder, elbow, wrist and gripper respectively in fig.2.

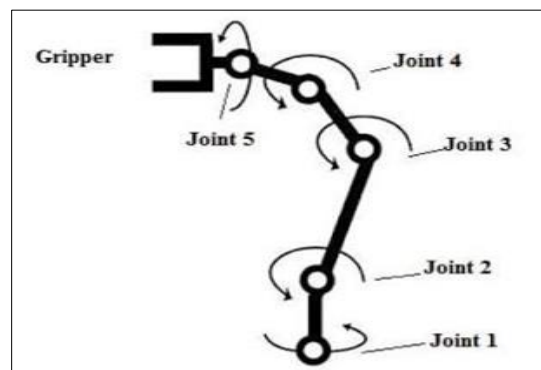


Figure 1a StanR2022 robot arm with 5 joints angle

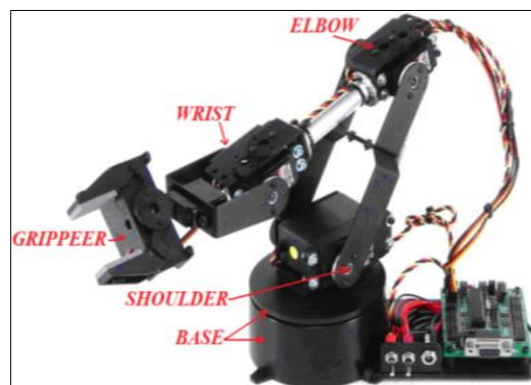


Figure 1b Angular joints representation of StanR2022

3. Output of Modeling (Mathematical) of Stimulated Arm

The controlling moment (forces) of the robot arm StanR2022 is mathematically modeled in Simulink Software. StanR2022 robot arm is considered as a kinematic chain with no branching. Modeling of the dynamical effect of robot arm consists of discovering inertia matrix gravitational forces vector and Coriolis and centrifugal forces matrix. The joints were not considered as ideal and therefore damping was accounted for. After finding all above mentioned parameters, it was then plugged in Euler-Lagrange equations of motion which have following general expression:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \left(\frac{\partial L}{\partial q} \right) = \tau, i = 1, 2, \dots, n \dots\dots\dots (1)$$

Letting L to be the Lagrangian which equals the difference between kinetic and potential energy of the system regarding the concerned robot arm.

$$L = E_k - E_p \dots\dots\dots (2)$$

Taking equation 2 into account, equation 1 can be rewritten into an absolute form of:

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{q}_Y} \right) - \left(\frac{\partial E_k}{\partial q_Y} \right) = Q_Y, Y = 1, 2, \dots, n \dots\dots\dots (3)$$

The matrix form of equation 3 becomes:

$$[a(q)]\ddot{q} + C(q, \dot{q}) = Q_c \dots\dots\dots (4)$$

The mathematical parameters can be denoted as follows: $a(q)$ signifies inertia matrix, $C(q, \dot{q})$ signifies vector with combined Coriolis, the centrifugal gravitational forces and Q_c signifies vector of control moments (forces).

4. The Robot Arm Simulation and Modeling Tools

The Simulation of this robot arm was done in MATLAB - Simulink software. Since the simulated robot arm (StanR2022) has 5DOF, setting up equations with block diagram representation was cumbersome. Then, to overcome the cumbersomeness, the following two solutions should be adopted: first solution is to model equations of motion in MATLAB symbolically and then (further) to generate function blocks directly in Simulink. Second solution is the using of Sims cape toolbox and modeling the robot arm similar to 3D modeling in Solid works. But in this research the equations of motion were calculated by Sims cape and there was no worry about the challenges and their remedies. In furtherance, the control law was implemented. Since the robot arm end effector has to follow certain reference trajectory, computed torque was chosen in feed forward plus PD (Proportional Derivative) controller in feedback branch. The feed forward torque was computed for every time instance of trajectory and combined with simple PD controller to generate input for robot. The computed torque compensated for the modeled robot arm dynamics like gravitational forces, inertia and centrifugal forces and nevertheless, does not for some additional external disturbances which were mitigated by applying PD controller.

In achieving a control of the robot arm, a personal computer (PC) was incorporated; and a strong connection was made between the robot and PC. This connection is called interface connection and it was done by using a microcontroller since its development is to enhance interference and interaction with electrical/electronic devices such as sensors and actuators, and high-tech gadgets to automate systems. The complete control process was accomplished in software environment. The software environment has two divisions comprising of CUBLOC microcontroller program and MATLAB Program. In CUBLOC program, a code was written to make the interfacing between PC and StanR2022 robot arm. The MATLAB program consists of the Serial Communication code and the graphical user interface (GUI). The complete system functions consist of three parts, namely: forward kinematics, trajectory planning and a controller.

Despite the programming trends and challenges in the development of the RSS (Robot Simulation Software) which are divided into two components such as the graphical user interface (GUI) and the control software, RSS and on-off line programming were importantly employed because of the essentiality to evaluating and predicting the behavior of a robot, and of its increasingly important role in the evolution of manufacturing automation.

5. Controlling of the Simulated Robot Arm

In order to control the simulated robot arm, control algorithm is applied and the equation (equation 5) containing state vector and the control vector are transferred to space-representation.

$$\dot{x}(t) = \begin{bmatrix} x_{2t} \\ -[a(x_1)]^{-1} c(x(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ [a(x_1)]^{-1} \end{bmatrix} u(t) \dots\dots\dots (5)$$

The reason of transference is to find Q_c (vector of control moments/forces) such that StanR2022 robot arm follows prescribed trajectory which is given in internal generalized coordinates. Feed forward and feedback combined control algorithm are applied in control of the simulated robot arm. In feed forward branch, it contains something similar to computed torque, but only with desired trajectory figuring in formula. This feed forward branch served for robot arm dynamics compensation, like inertia, gravity of robot itself, etc. In feedback branch there is PD (Proportional Derivative) controller that did the job of converging error to zero when external disturbances were detected. Block diagram of such system is given below in fig.2. These two branches are combined to form control input to the system.

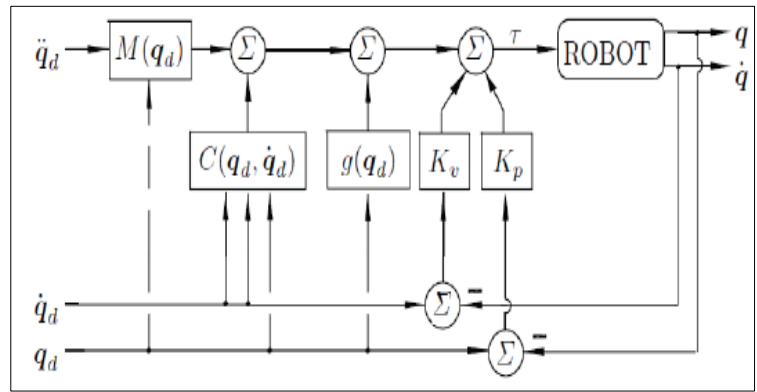


Figure 2 Block diagram of PD control system

6. Mathematical Delineation of the Trajectory Planning

The main concern in trajectory planning is the problem of determining $q(t)$ which represents a scalar joint variable. It was supposedly noted that at time t_0 the joint variable satisfies:

$$q(t_0) = q_0 \dots\dots\dots (6)$$

$$\dot{q}(t_0) = v_0 \dots\dots\dots (7)$$

The quantity q_0 is the initial angle at initial time t_0 and v_0 is the angular velocity at initial time t_0 . Considering attaining the values at t_f (final time):

$$q(t_f) = q_f \dots\dots\dots (8)$$

$$\dot{q}(t_f) = v_f \dots\dots\dots (9)$$

Here, q_f is the final angular rotation at final time t_f and v_f is the angular velocity at final time t_f .

Since StanR2022 is a 5 DOF robot arm, the angle of rotation and angular velocity were determined by using cubic polynomial trajectories as in the equations below.

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \dots\dots\dots (10)$$

$$\dot{q}(t) = a_1 + 2a_2 t + 3 a_3 t^2 \dots\dots\dots (11)$$

Where q is the angle of rotation and \dot{q} is the angular velocity of the expected and articulated motion of the robot arm.

7. Result of the Simulated Robot Arm (Control) and Discussion

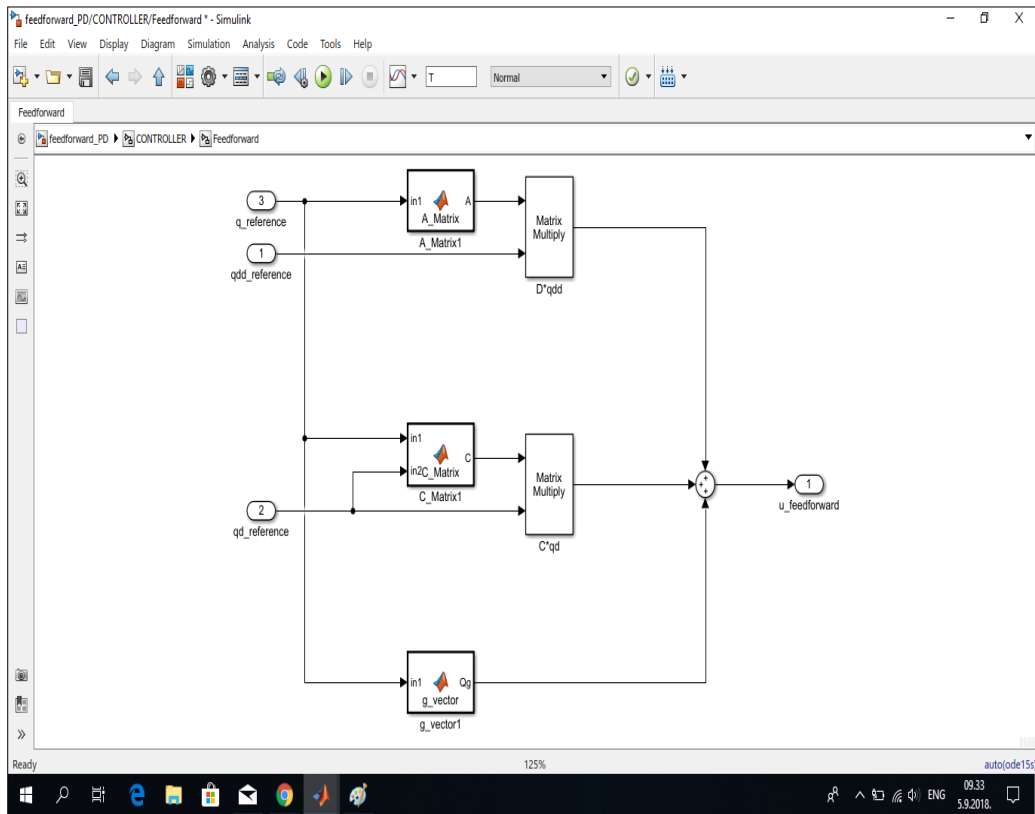


Figure 3 Feedforward PD: controller/feedforward - simulink

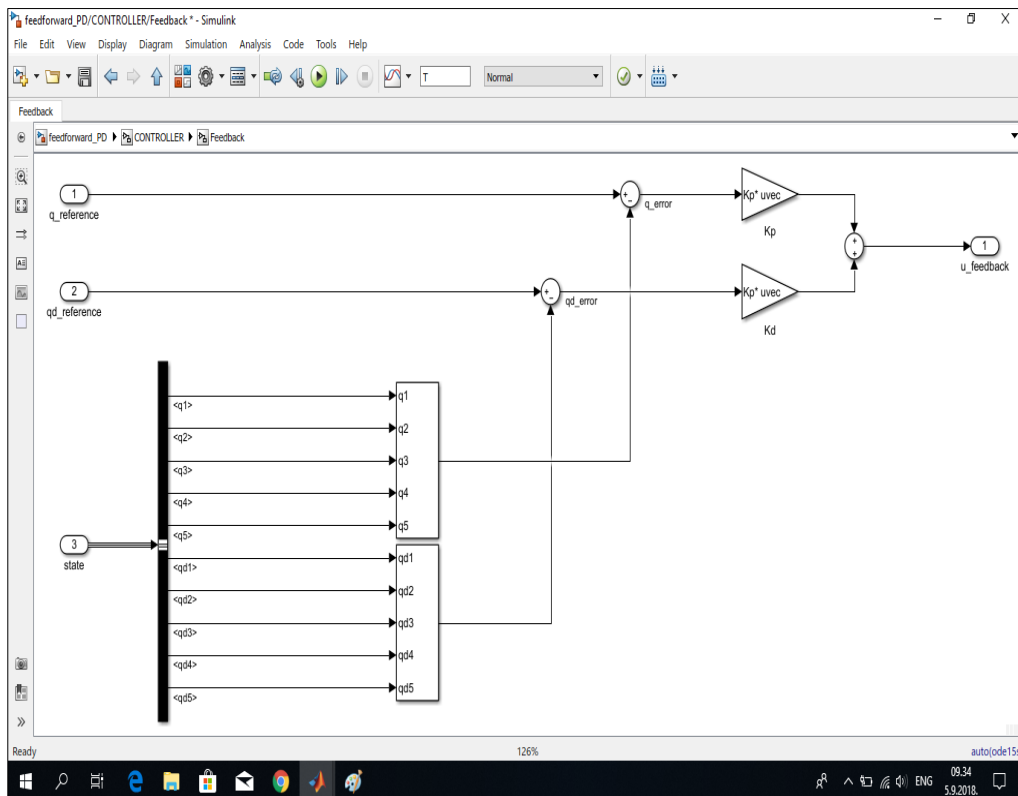


Figure 4 Feedforward_PD: controller/feedback – Simulink

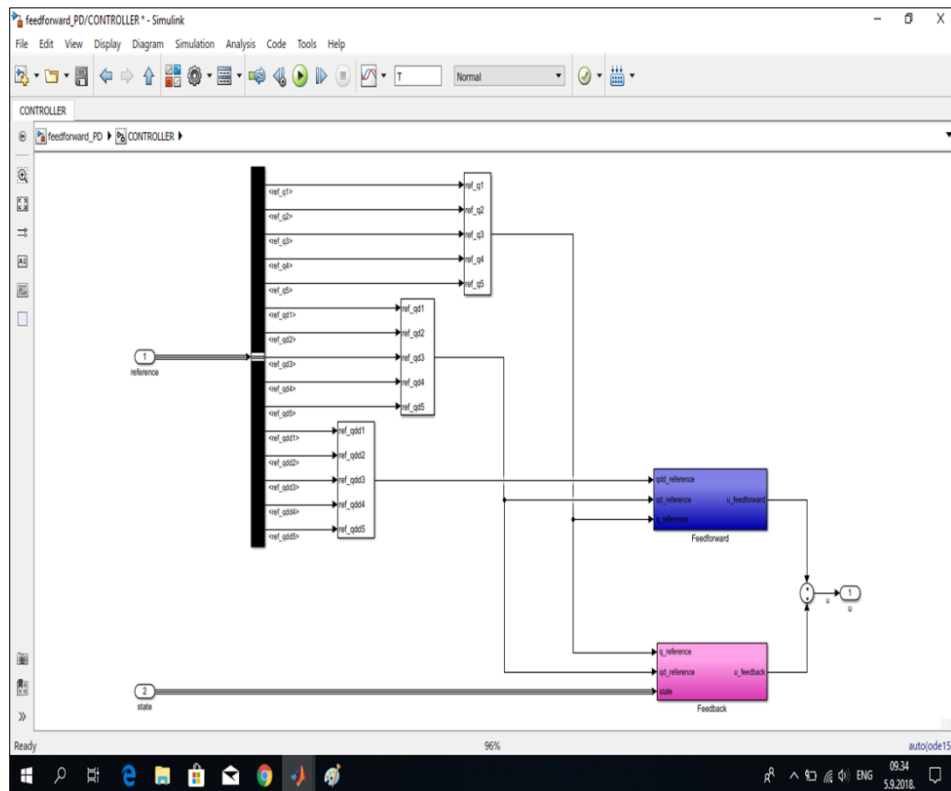


Figure 5 Feedforward_PD controller (feedforward and feedback) - simulink

From the diagram of figure 3, states 1, 2 and 3 are the reference angular velocities, angular acceleration and the angle of rotation respectively and they are line 1, 2 and 3 of the trajectory generation simulated version. The three states underwent matrix simulation in which two of the states also underwent a matrix multiply. State 1 and 3 were simulated through matrix multiply; state 2 and 3 were matrix in multiple while the state 3 was simulated into a vector. Then the requests of state 1 and 3 matrix multiple ($C \cdot Q_{dd}$); state 2 and 3 matrix multiple ($C \cdot q_d$) and the vector form of state 3 (g_vector) collectively were merged to produce the feedforward ($u_forward$).

As illustrated in figure 4, the state 1 and 2 represent the reference angle (q -reference) and angular velocity (q_d -reference). But state 3 comprises of the actual angle of rotation and angular velocity of the specimen robot. Then the angles of actual robot for different joints (i.e. q_1, q_2, q_3, q_4 and q_5 corresponding to joint 1, joint 2, joint 3, joint 4 and joint 5 respectively) and the angular velocities of the joints ($q_{d1}, q_{d2}, q_{d3}, q_{d4}$ and q_{d5}) were simulated with the reference angles and angular velocities of reference human arm at each joint. The human arm that is a reference served as the controller. It resulted to feedback of output ($u_feedback$) for the robot dynamic.

In figure 5 which is the PD controller Simulink result comprised of the reference dynamic angles and angular velocities of the reference which serve as the forward Simulink result and at state 2 the feedback occurred. The reference and the state 2 were compared as output 1 in the diagram. The output becomes the final result of the controller.

8. Conclusion

The output (results) gotten in the simulation of robot arm StanR2022 in Simulink software established a relationship between the angle of the joints of the human arm and robot arm. The relationship shows that the joints were labeled joint 1 as link 1, joint 2 as link 2, joint 3 as link 3, joint 4 as link 4 and joint 5 as link 5. The angle values of each five joints of robot were represented as q_1, q_2, q_3, q_4 and q_5 and the angle values of each five joints of real human arm were represented as $ref_q_1, ref_q_2, ref_q_3, ref_q_4$ and ref_q_5 which served as the reference angles of comparison. The angle of rotation and angular velocity of the joints of the simulated robot arm, StanR2022, performed tremendously in same controlling (selection, gripping, handling and movement) capability of real human arm. Therefore, the simulated robot arm (StanR2022) could conveniently and comfortably supplant human arm in the requiring and supposed dynamic activities as seen from the results.

Recommendations

- In the advent of a robot arm, the joint angles (rotational angles) should be satisfactorily actuated. Hence, control mode of the angles should be appropriately articulated for controlling efficiency.
- Robot arm if properly and carefully modeled and simulated is a reliable supplant to human arm in accomplishing heavy and difficult tasks in an unsafe environment.

Compliance with ethical standards

Acknowledgments

As the lead researcher, I, Okechukwu Stanley Ikwunze, hereby acknowledge the effort of everyone whoe contributed to the success of this research, especially my research assistant, Victor Ikedichi Okparaku. Also, my gratitude goes to TET Fund (Tertiary Education Trust Fund) for the sponsorship of this research.

Disclosure of conflict of interest

No conflict of interest exists between the researchers (authors).

References

- [1] Chun HA, Khin TL, Yin MM. SS Modeling Motion Control System for Motorized Robot Arm using MATLAB, Proceedings of World Academy of Science, Engineering and Technology. 2008, 32: ISSN 2070-3740.
- [2] Gentilucci M, Castiello U, Corradini ML, Scarpa M, Umilta C, Rizzolatti G. Influence of different types of grasping on the transport component of prehension movements. *Neuropsychologia*. 1991, 29(5): 361–378.
- [3] Javier AP, Fani D, Daniele R, Guang-Zhong Y. Artificial Intelligence and Robotics. UK-RAS Network Robotics & Autonomous System. 2016, 56.
- [4] Johan JC. Introduction to Robotics Mechanics and Control. Prentice Hall, 3rd Edition. 2005, 109-114.
- [5] Lehman J, Miikkulainen R. Neuroevolution. *Scholarpedia*. 2013, 8(6): 30977.
- [6] Marco F, Butzler J, Schlick CM. Human-robot cooperation in future production system: Analysis of requirements for designing an ergonomic work system. Chair and institute of industrial engineering and ergonomics. 2015, 3: 510-517.
- [7] Monkman GJ, Hesse S, Steinmann R, Schunk H. Robot Grippers. Wiley VCH Verlag GmbH and Co. KGaA, Weinheim. 2007.
- [8] Muhammad IJ, Habibollah H, Dewi N. Development of Robot Simulation Software for Five Joints Mitsubishi RV-2AJ Robot using MATLAB/Simulink and V-Realm Builder. Fifth International Conference on Computer Graphics, Imaging and Visualization. 2008.
- [9] Om Prakash Gujel, Vidhatri Gujela, Dev Prakash Gujela. Six DOF Spray Painting Robot Analysis. International Journal of Advance Research in Elevation, Electronics and Instrumentation Engineering. 2015, 4(9): 7721.