Session 3: *Engineering Education and Practice*
Professional paper
UDC: 007.52:621.865.8

Programming of an Industrial Robot and Optimization of its Path Using the PSO Algorithm

Zvonko Petrović [1*], Vladeta Jevremović[1], Slobodan Aleksandrov[1]
[1] Academy of professional studies Šumadija, Trstenik, Serbia
[*] zvonko.petrovic@vtmsts.edu.rs

Abstract: *Modern technical systems must meet the high technical and technological requirements that the market offers today. The introduction of robot systems into the production process is produced from the basic steps to increase the flexibility and productivity of the technological system. The paper uses the program RobotStudio for programming the movement of robots and the Particle swarm optimization (PSO) algorithm for optimizing the path of movement of the robot. The aim of this paper is to obtain optimal values of the rotation angles of the robot segments so that the trajectory of the robot is minimal.*

Keywords: *industrial robot, path optimization, PSO algorithm*

1. INTRODUCTION

Based on the analysis conducted in the assembly sector of 355 German companies in the machine industry, it is pointed out that the main potentials of rationalization of production costs lie in the design of assembly-oriented products and in the automation of assembly operations [1]. This paper presents the application of the SCARA robotic system for the assembly of a single assembly. The programming of the robotic system was done in the RobotStudio software, and with the help of the biologically inspired PSO algorithm, the optimal values of angles were obtained, which should be entered into the program in order for the robotic system to avoid obstacles during installation with the shortest distance traveled.

Before introducing the robotic system into the configuration of the technological system, it is necessary to perform a systematic analysis of all aspects of economy and technological justification. There are several criteria on the basis of which the justification of the use of robots in performing certain technological operations in the technological system can be assessed, and they can be classified into three groups [2]:

- Technical and technological criteria,
- Economic criteria for justifying process automation and
- Criteria for humanization of the work process.

2. SOFTWARE ROBOTSTUDIO

Offline programming is the best way to maximize return on investment for robot systems. ABB's simulation and offline programming software, RobotStudio, allows robot programming to be done on a PC in the office without shutting down production.

RobotStudio provides the tools to increase the profitability of your robot system by letting you perform tasks such as training, programming, and optimization without disturbing production. This provides numerous benefits including:

- Risk reduction,
- Quicker start-up,
- Shorter change-ove,
- Increased productivity.

RobotStudio is built on the ABB VirtualController, an exact copy of the real software that runs your robots in production. This allows very realistic simulations to be performed, using real robot programs and configuration files identical to those used on the shop floor [3].

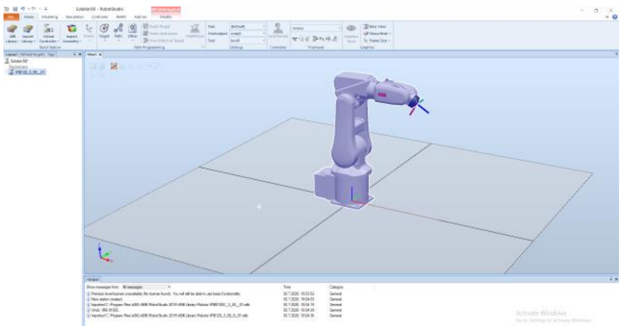

Figure 1. *Appearance of the home screen of ABB's RobotStudio robot programming software.*

3. ALGORITHM PSO

Particle swarm optimization PSO (Figure 1) represents metaheuristic method of optimization based on agents (particles) population, which was accidentally discovered by James Kennedy and Russell Eberhart in 1995, while studying the simulation of social behavior of bird flocking [4]. Just as it is the case with all algorithms based on population, initial particle population is generated first. Position of the particle represents vector of parameters which are optimized.

$$\mathbf{x}=(x_1,x_2,...,x_n) \quad (1)$$

or potential solution. Random position in space which is explored, as well as initial velocities, is given to each particle. After that, the value of goal function of each particle is determined, and that value is added to it as the best value for the particle in question, and the initial position becomes the best position of the particle $\mathbf{p}_{\mathbf{best}}$. When all the best values of particles are determined, the particle with the minimum value is searched, and its position becomes the best position for the entire swarm $\mathbf{p}_{\mathbf{gbest}}$. Afterwards, it needs to be checked whether the criteria of optimization are satisfied, and if they are, the obtained results are shown. If the criteria are not satisfied, new velocities and positions need to be calculated.

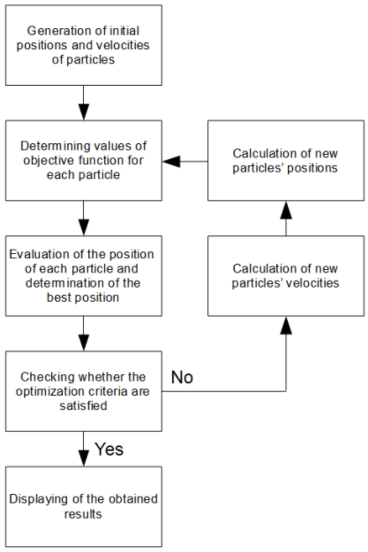

Figure 2. *Algorithm of the method of particle swarm optimization.*

Figure 2 graphically shows how to determine new velocities and positions in two-dimensional space of search.

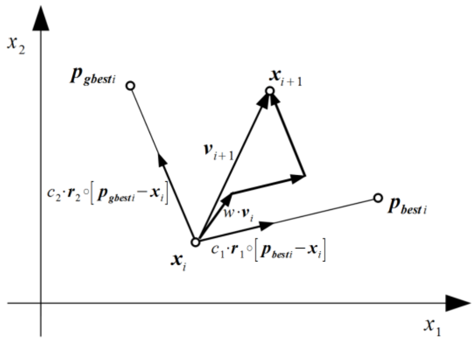

Figure 2. *Updating of velocity and position of the i particle.*

New velocity of each particle consists of three components:

1. the component which depends on instantaneous particle velocity,
2. the component which is proportional to the distance of instantaneous position of the particle and its best value,
3. the component which is proportional to the distance of instantaneous position of the particle and its best position for the entire swarm.

$$\mathbf{v}_{i+1}=w\cdot\mathbf{v}_i+c_1\cdot\mathbf{r}_1\circ\left(\mathbf{p}_{\mathbf{best}i}-\mathbf{x}_i\right)+c_2\cdot\mathbf{r}_2\circ\left(\mathbf{p}_{\mathbf{gbest}i}-\mathbf{x}_i\right) \quad (2)$$

where w represents inertia weight, c_1,c_2 are acceleration coefficients or correction factors, $\mathbf{r}_1,\mathbf{r}_2$ represent two random vectors of the length n within the limits [0,1]. The symbol $\circ$ represents Hadamard product:

$$(A\circ B)_{i,j}=(A)_{i,j}\cdot(B)_{i,j} \quad (3)$$

Inertia weight w impacts the first component, and for the values in the range of 0,9 – 1,2 [5] it gives the best results, that is, the algorithm has greater chances of finding the global minimum for a reasonable number of iterations. For coefficient values which are smaller than 0,8, if algorithm finds global minimum it will find it fast. Particles in this case move quickly and it can happen that they "fly over" some area, so it can happen that they do not find global minimum. On the other side, if inertia weight has bigger value, then particles search the solution space more thoroughly and the chances of finding global minimum are greater.

Acceleration coefficients c_1 and c_2, when multiplied by random vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, stochastically manage the impact of the two other velocity components. Usually, their assumed value is approximately 2, in order for the middle value of the product of acceleration coefficient and random vector to be approximately 1. New position of the particle is determined by simple adding of the current position $\mathbf{x}_i$ and new particle velocity $\mathbf{v}_{i+1}$.

$$\mathbf{x}_{i+1}=\mathbf{x}_i+\mathbf{v}_{i+1} \quad (4)$$

The values of the goal function for new positions of the particle are determined again, and for each particle new and old values of the goal function are compared. If the new value is smaller, then it becomes new best value and the current position becomes the best position of that particle. The position of the particle with the smaller value becomes new best position for the entire swarm. Again, it needs to be checked whether the optimization criteria are satisfied; if they are, the results are shown, and if not, the entire procedure will be repeated until the criteria are satisfied.

This is the simplest version of the algorithm of particle swarm optimization. Other versions do not have constant values for the parameters w, c_1 and c_2, but they alter by specific rules during the implementation of the algorithm. In addition, other PSO algorithms also include different swarm topologies, that is, the way in which particles in the swarm communicate.

4. ANALYSIS OF MOUNTING ASSEMBLY TO BE INTRODUCED FOR ROBOTIC SYSTEM

The assembly to be assembled consists of a pump body (15), a timing shaft subassembly (25), and a washer subassembly (24). Skop is a part of the manual pump for lifting the cabin of the "Kamaz" truck. The pump body is fixed while first the pin (25) is inserted into the body and then on the other side of the washer subassembly (24) as shown in the assembly plan:

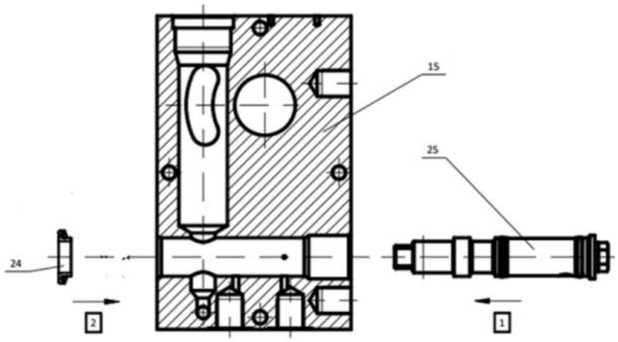

Figure 3. *The assembly plan*

In the SolidWorks software package, we have to model these components, which we later import into RobotStudio together with the robot. In this way we form a real environment and the programming and optimization of robot movements is performed in a real environment. Figures 4 and 5 show the two assembly steps that the robot needs to perform.

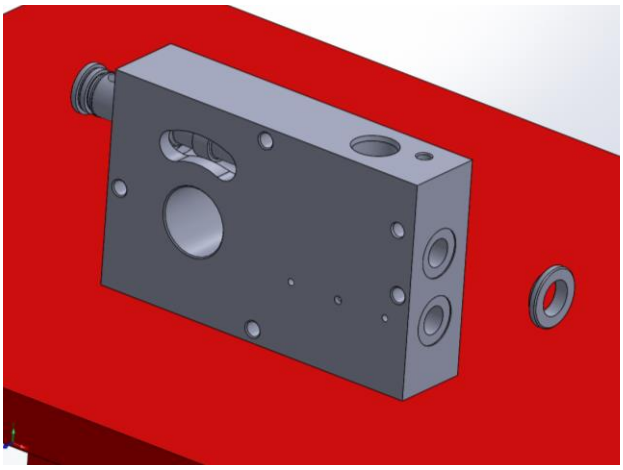

Figure 4. *Pin assembly*

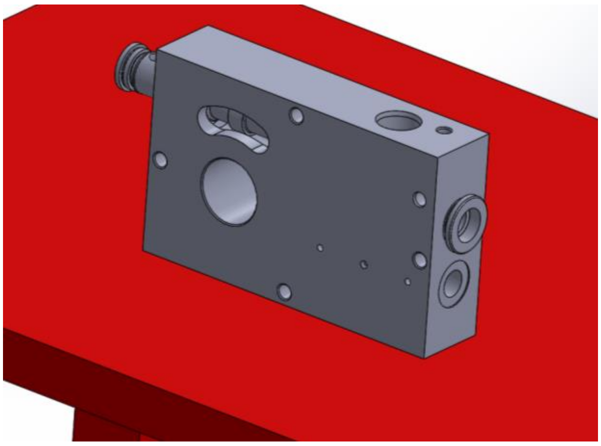

Figure 5. *Assembly of the washer subassembly*

For these assembly operations, a SCARA robot should be introduced and the shortest assembly path determined. RobotStudio is used to program the robot's movement and the biologically inspired PSO algorithm is used to optimize the path.

5. OPTIMIZATION OF THE ROUTE OF THE ROBOT SYSTEM USING THE PSO ALGORITHM

By analyzing the assembly plan and the need to move the robotic system, it can be concluded that we have the movement of the robotic system around the obstacle that the housing makes. Analysis of this problem can be found in scientific papers [6].

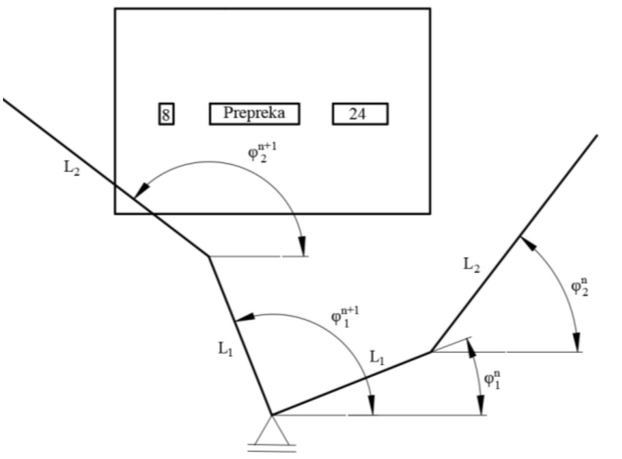

Figure 6. *Schematic representation of optimization problems*

In order to determine the optimization problem, the path of the robot during assembly is first analyzed. We see that the disputed path is "3" then the robot must go around the obstacle (pump body) and continue with the assembly.

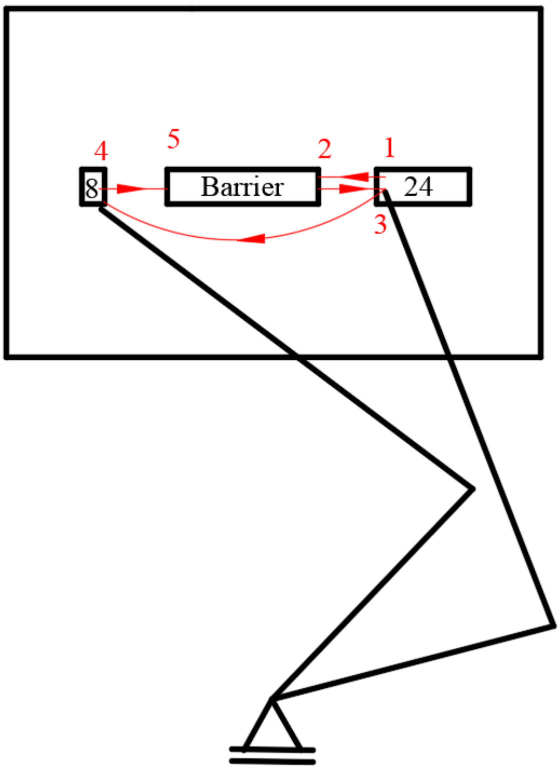

Figure 7. *Robot system motion analysis*

The robotic system consists of two segments that can rotate around the "Z" axis. The final equations of motion of the robot endpoint called the Tool Center Point (TCP) can be written in the form of expression (1):

$$X_P = L_2 \cdot \cos(q_1 + q_2) + L_1 \cdot \cos(q_1)$$
$$Y_P = L_2 \cdot \sin(q_1 + q_2) + L_1 \cdot \sin(q_1) \quad (1)$$

The robotic system consists of two segments that can rotate around the "Z" axis. The final equations of motion of the robot endpoint called the Tool Center Point (TCP) can be written in the form of expression (1).

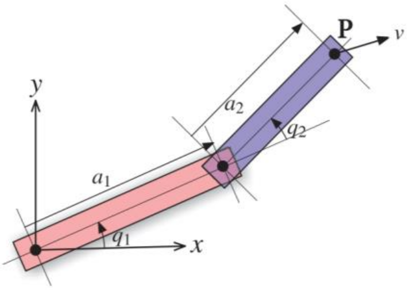

Figure 8. SCARA robot configuration [6]

The desktop space where the assembly is performed is actually the space searched by the PSO algorithm (green surface) and the area occupied by the pump body is the restriction where the robot must not pass (red surface).

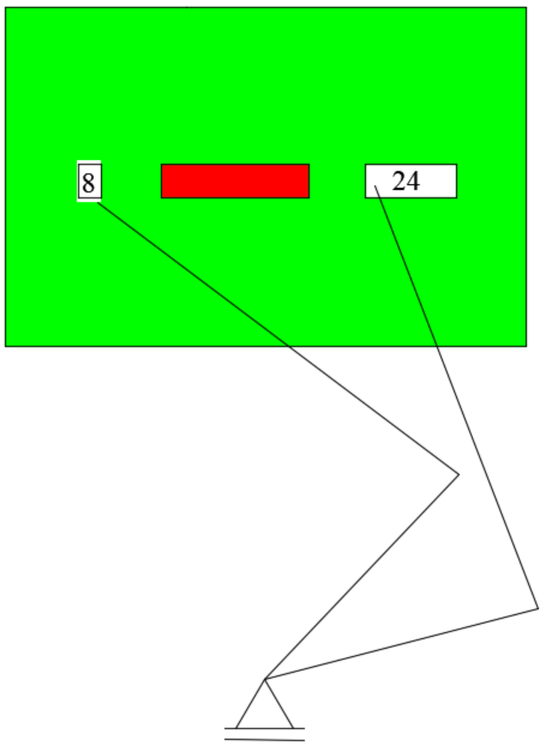

Figure 9. Workspace with barrier

The limitations that must be met are:

$$\begin{aligned} &0 < q_1 < 360 \\ &0 < q_2 < 360 \\ &X_P \in \{0,180\} \cup \{320,460\} \\ &Y_P \in \{0,130\} \cup \{160,300\} \end{aligned} \quad (2)$$

Figure 10 shows the limitations of the area occupied by the pump body.

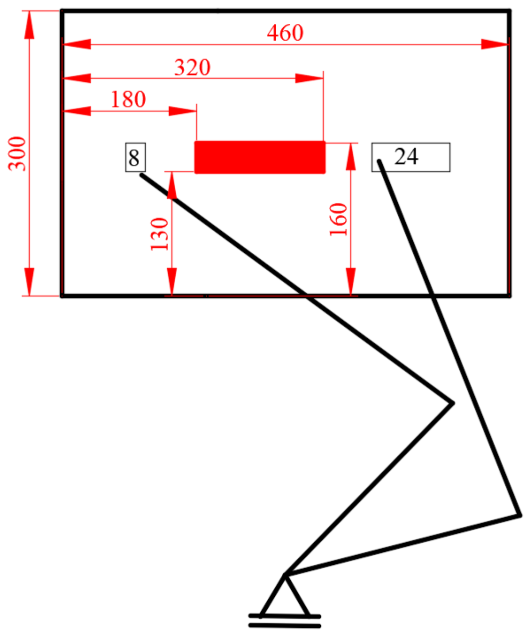

Figure10. *Position of the barrier in the workspace*

Based on the objective function (1) and constraint (2), with the help of the PSO algorithm, the values of the coordinates k1 and k2 are obtained, which the robotic system should occupy while crossing the shortest path. We enter these values when we program the movement of the robotic system in RobotStudio and by simulating the movement we can check if the path we have programmed is adequate.

6. ROBOT SYSTEM PROGRAMMING IN THE ROBOT STUDIO SOFTWARE PACKAGE

In the RobotStudio studio package, we can import the robot we want and the product that is mounted in the right size. In this way, we perform programming in a real environment. The product is modeled in SolidWorks, and the robot is in the RobotStudio database.

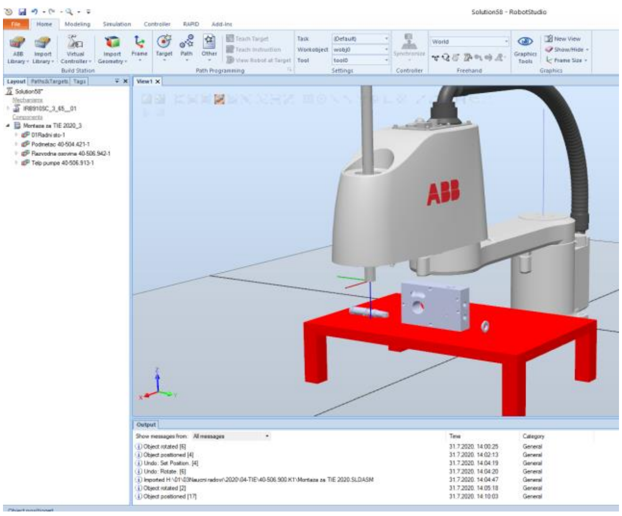

Figure 11. *The first step, forming a real environment*

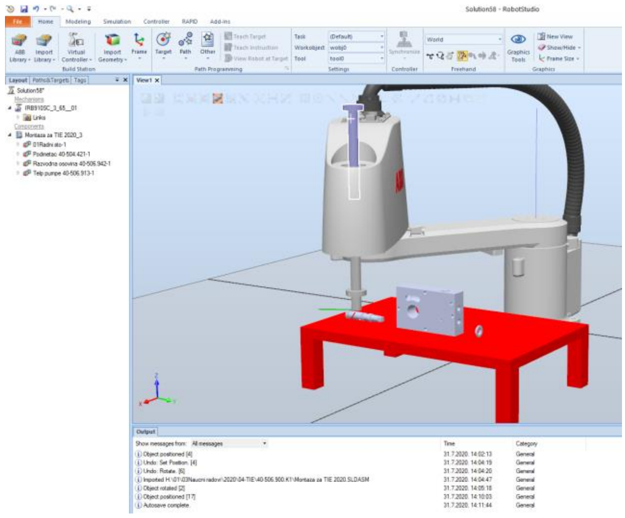

Figure 12. *The second step, mounting an axle*

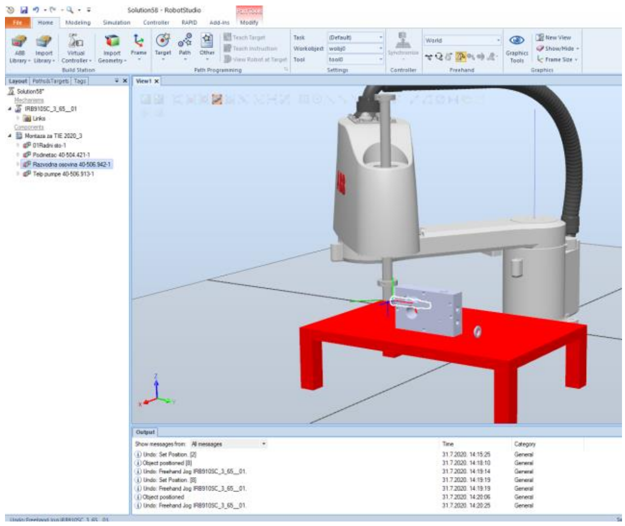

Figure 13. *The third step, bypassing the obstacle*

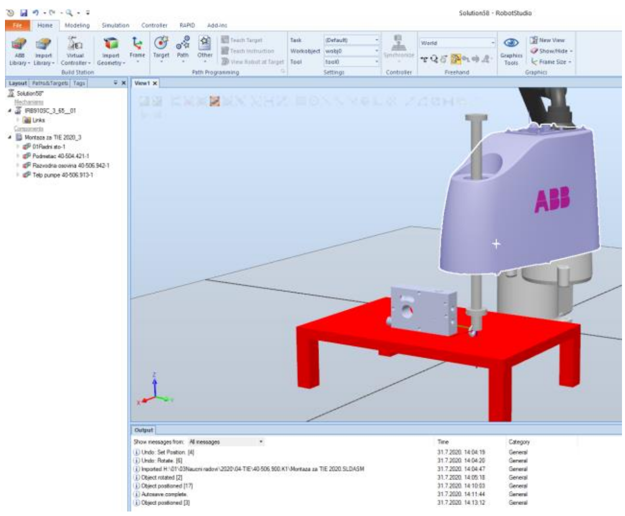

Figure 14. *The fourth step, mounting the washer subassembly*

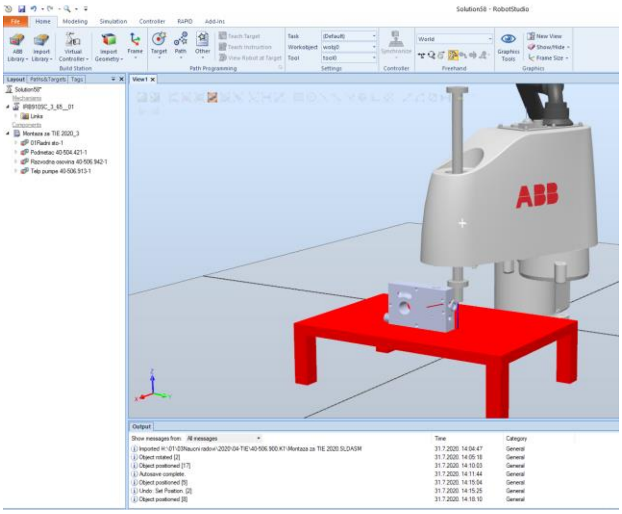

Figure 15. *Fifth step, mounting the washer subassembly*

7. CONCLUSION

Modern technological systems must be able to respond to all the challenges posed by the market today. That is, they must be intelligent technological systems. The application of biologically inspired algorithms in combination with the simulation of production processes enable the technological system to increase its productivity.

In addition, the application of application programs for robot programming is more efficient than for the designer to determine the paths that the robot must describe.

The paper shows on the example of the assembly of a hand pump that it is possible to find the minimum path that the robot should go during the assembly in order to make the assembly process as short as possible and therefore cheaper.

REFERENCES

[1] *Djapic M., "Evidential systems in product and process development", Lola institute, Beobrad, (2005)*

[2] *Lukić, Lj. "Fleksibilni tehnološki sistemi", Mašinski fakultet Kraljevo, Kraljevo, 2008*.

[3] https://new.abb.com/rs

[4] J. Kenedy and R. Eberhart*. Particle swarm optimization. In IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.*

[5] Shi, Y.; Eberhart R. (1998)*. A modified particle swarm optimizer. Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69-73*.

[6] *Petar Ćurković. Evolutive algorithm for managinf a multi-agent robot system. Universitz of Zagreb, faculty of mechanical engineering and shipbuilding. 2010.*