9th International scientific conference
Technics and Informatics in Education – TIE 2022
16-18 September 2022

Session: *Engineering Education and Practice*
Professional paper
DOI: 10.46793/TIE22.201P

Application of Robotic Vision and PSO algorithm for determining the optimal path of movement of the robotic system

Zvonko Petrović [1*], Milica Tufegdžić[1], Vladeta Jevremović[1] and Predrag Pravdić[1]
[1] Academy of professional studies Šumadija/Department, Trstenik, Srbija
[*] zpetrovic@asss.edu.rs

Abstract: *The application of robotic vision and biologically inspired algorithms improves the process of programming the movement of a robotic system. In the work, the optimal path of movement of the robotic system during the welding process was determined. The shape of the path to be taken was recorded using a 2D camera, i.e. robotic vision, and the optimal path of movement determined by the PSO (Particle swarm optimization) algorithm.*

Keywords: *robotic vision; PSO algorithm; optimal path.*

1. INTRODUCTION

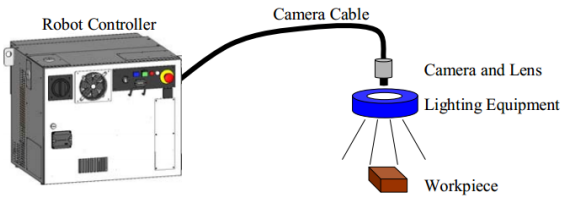
In order to achieve accuracy in the movement of the robot system, reduce the time of completing tasks and maintain quality, it is necessary to apply Industry 4.0 concepts. The Fanuc company has developed the iRVision program, which is used for cooperation between the robot and the camera in order to better perform the task assigned to the robot. With this approach, so that the robotic system can communicate with the camera and coordinate its work, opportunities are created for solving tasks in assembly, welding, control, etc. In the production of boilers for steam heating, there is a need for welding parts. The seams are mostly straight, but they are in space along different axes and at different heights. In the work, first the position of the seams through which the robotic system should pass was recorded with a 2D camera, and then the optimal path along which the robotic system should move was determined by the PSO algorithm in order to minimize the time required to perform the task.

2. *i*RVision FANUC VISION SYSTEM

The camera and lens of 3D Laser Vision Sensor are sane as the two-dimensional camera, so the 3D Laser Vision Sensor can also be used for the two-dimensional applications.

*i*RVision consists of the following components:

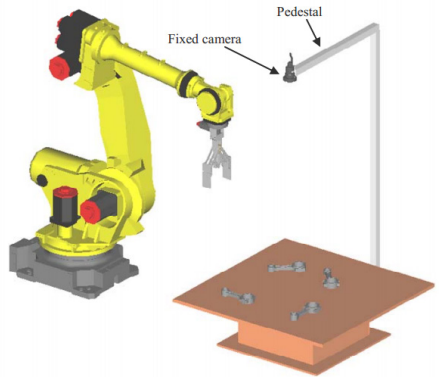
- Camera and lens,
- Camera cable,
- Lighting Equipment,
- Camera multiplexer (used if needed)



2.1. Fixed camera and robot-mounted camera

Fixed camera

- Detect workpieces using the camera installed on the stand,
- A fixed camera will always snap the same place from the same distance,
- While the robot transfers the workpieces, *i*RVision can detect the other workpieces, so the cycle time can be shortened,
- Use a sufficient strength camera stand so that the camera doesn't vibrate.



Robot-mounted camera

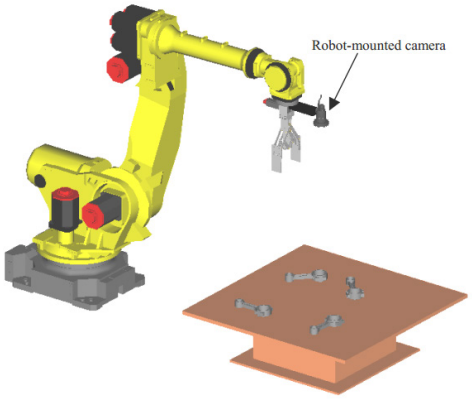
• The robot-mounted camera is mounted on the wrist unit of the robot,

• By moving the robot, you can measure different places with a robot-mounted camera,

• When a robot-mounted camera is used, *i*RVision calculates the position of the workpiece based on the movement of the robot,

• The camera must be mounted on the final axis of the robot. For example, when a six axis robot is used, the camera must be mounted on the sixth axis of the robot.

• The camera cable moves according to the robot movement, so be careful so that the cables doesn't tangle.



Depending on the size and location of the workpiece, determine the size of the field of view of the camera. The size of the field of view of the camera is determined by three factors: The size of the image sensor, the focal distance of the lens, and the distance from the camera to the workpiece. The size of the image sensor (Lc) is calculated by the following formula:

$$\text{Lc} = \text{Cell size} \times \text{Image size (pixels)} \quad (1)$$

The rough value of the field of view of the camera (L) is calculated by the following formula.

$$\text{L} = (\text{D} - \text{f}) \div \text{f} \times \text{Lc} \quad (2)$$

When the distance D from a camera to a workpiece is 700mm and the monochrome camera (SC130EF2) is used, the view size is shown below table.

Table 1. *Table captions should be placed above the table*

Distanc	Area
8 mm	587 mm x 469 mm
12 mm	389 mm x 311 mm
16 mm	290 mm x 232 mm
25 mm	183 mm x 147 mm

The calculation result is an approximate value. Some difference may occur between the calculated value and the actual measurement value. When an accurate value is required, please confirm by the actual measurement.

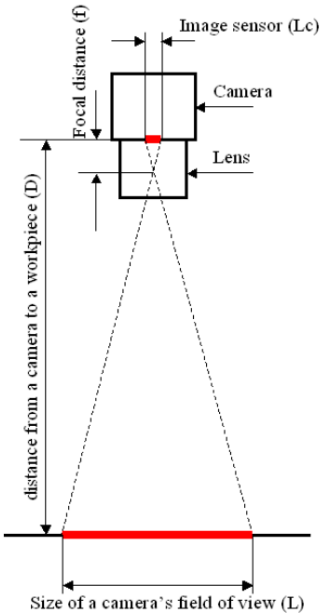


Figure 4. Size of field of view of a camera [1]

2.2. Fixed camera and robot-mounted camera

The fixed frame offset and the tool offset can be used to offset the robot positions. *i*RVision supports both kinds of robot position offsets. Fixed frame offset detect the workpiece on the table, and offset the robot positions so that the robot works (for example, the robot picks up the workpiece.) in correct.

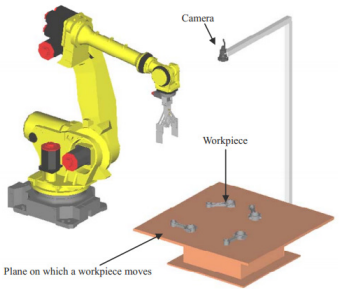


Figure 5. Fixed frame offset [1]

Tool offset detect the workpiece which gripped by the robot, and offset the robot positions so that the robot works (for example, the robot places up the workpiece.) in correct.

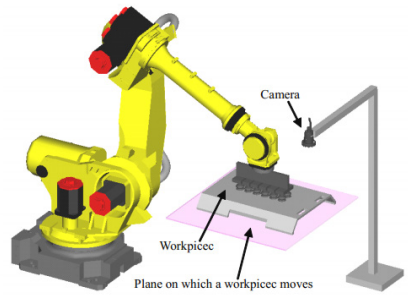


Figure 6. Tool offset [1]

2.3. Calculation of the offset data

The offset data is calculated from the position of the workpiece of when teaching the robot program and the position of the current workpiece. The position of the workpiece of when the robot program was taught is called as the reference position, and the current position of workpiece is called the actual position. *i*RVision measures the reference position when the robot program is taught, and stores it internally. The operation of teaching the reference position to *i*RVision is called reference position setting.

In the case of the following figure, the position of "+" mark is a found position of a workpiece. If a robot approaches only to the position of "+" mark, the offset data can be calculated by subtracting the value of the actual position and the reference position. When the calculation of the offset data is subtraction, it is easy to understand, however there are also limitations.

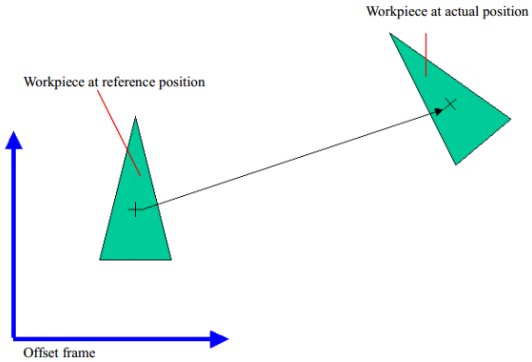


Figure 7. Tool offset calculation by subtraction[1]

In the following figure, the position M is the reference position and the position m is the actual position. A workpiece is placed on the reference position and the robot traces from the position A to the position B and C. When the workpiece is placed at the actual position, to trace the -- a, b and c --, each positions information are required. However, the movement of (a – A), (b – B) and (c – C) differ from the movement of the found position (m – M). So, it is necessary to calculate the offset data of a, b and c individually.

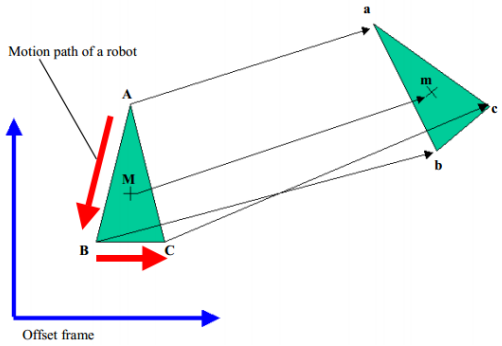
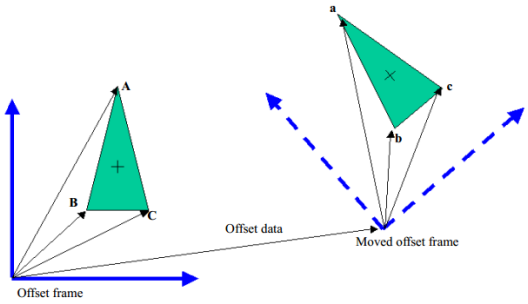


Figure 8. Position information and movement amount [1]

*i*RVision uses an offset frame, it is unnecessary to calculate each position individually. In the following figure, *i*RVision moves the offset frame to a new position. The position of the workpiece relative to the offset frame is the same as the position of the workpiece at the reference position by moving the offset frame, it becomes unnecessary to calculate the offset data for each point individually, and teaching becomes easy. *i*RVision outputs the movement of offset frame as the offset data. Since the offset data is the movement of the user frame, it is not the physically movement of the workpiece. Moreover, the offset data does not become an intuitive value in many cases. Normally, when the amount of rotation of the workpiece is the larger or the distance from the origin of the user frame to the workpiece is the further, the value of the offset data differs from the physically movement of the workpiece.



3. ALGORITHM PSO

Particle swarm optimization PSO (Figure 10) 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 behaviour of bird flocking [2]. 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:

$$x = (x_1, x_2, \ldots, x_n) \quad (3)$$

or potential solution. Random position in space which is explored, as well as inital 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 $\mathbf{p}_{\text{best}}$. 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 $\mathbf{p}_{\text{gbest}}$. 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.

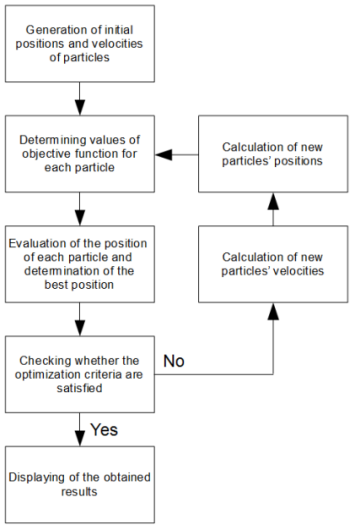


Figure 10. *Algorithm of the method of particle swarm optimization.*

Figure 11 graphically shows how to determine new velocities and positions in two-dimensional space of search.

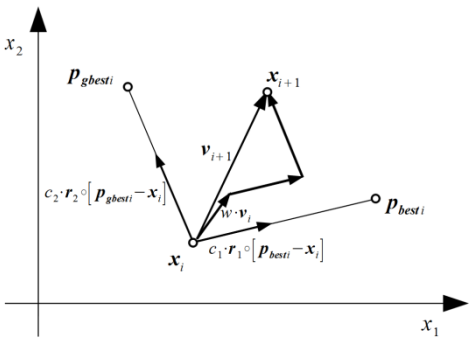


Figure 11. *Updating of velocity and position of the i particle.*

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.

$$\mathbf{v}_{i+1} = w \cdot \mathbf{v}_i + c_1 \cdot \mathbf{r}_1 \circ \left(\mathbf{p}_{\text{best}i} - \mathbf{x}_i\right) + c_2 \cdot \mathbf{r}_2 \circ \left(\mathbf{p}_{\text{gbest}i} - \mathbf{x}_i\right) \quad (4)$$

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 [4]:

$$\left(A \circ B\right)_{i,j} = \left(A\right)_{i,j} \cdot \left(B\right)_{i,j} \quad (5)$$

Inertia weight w impacts the first component, and for the values in the range of 0,9 – 1,2 [3] 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 $\mathbf{r}_2$, stochaistically 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 $\mathbf{v}_{i+1}$.

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_{i+1} \quad (6)$$

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 THE WELDING TASK THAT THE ROBOT SYSTEM SHOULD PERFORM

Welded joints are often used in the production of heating boilers. In places where it is necessary that there is no permeability of smoke and fire, a welded joint is the ideal solution. In order to solve this task, robots with six degrees of freedom of movement are applied: PUMA type industrial robot as shown in the picture.

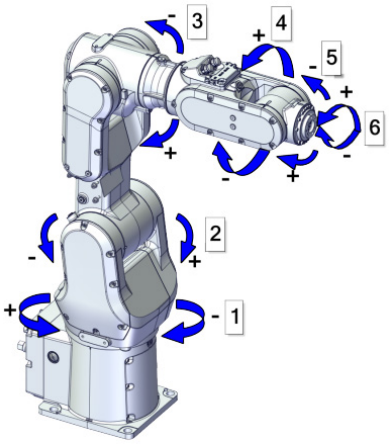


Figure 12. ABB industrial robot IRB1100[4]

Table 2. *The axis of robot system*

Pos	Description	Pos	Description
1	Axis 1	2	Axis 2
3	Axis 3	4	Axis 4
5	Axis 5	6	Axis 6

Picture 13 shows a boiler that needs to be welded together.

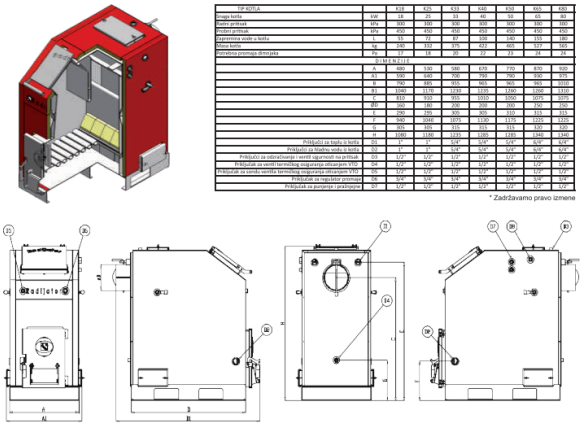


Figure 13. Technical characteristics of the boiler to be welded

5. APPLICATION OF *i*RVision FANUC VISION SYSTEM-A FOR DEFINING THE PICTURE OF THE TASK THAT THE ROBOT SYSTEM SHOULD PERFORM

To program the robot movement using the iRVision FANUC VISION SYSTEM, a 3D model of the products to be welded is first required.

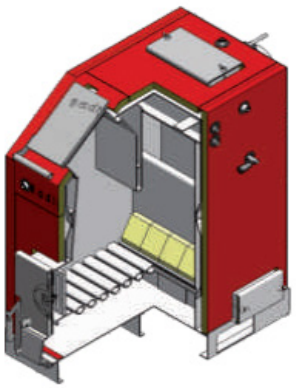
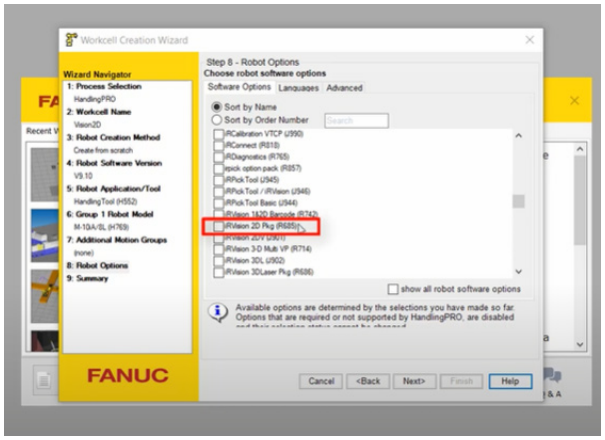


Figure 14. 3D model of the product to be welded

By starting the program, we select a 2D robot vision based on which we can create an image of the path that the robot system should describe.



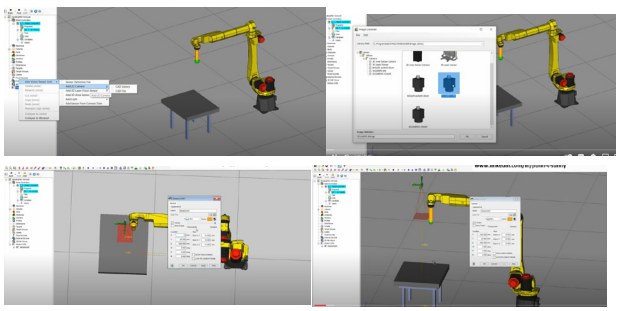


Figure 16. Necessary steps to set up the camera and workspace where the robot should perform its task

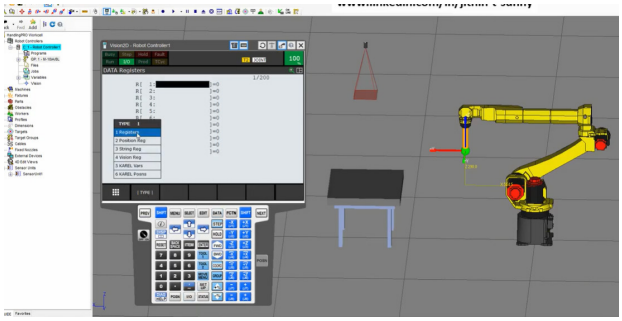


Figure 17. Adjusting the position of the 2D camera

Based on the application of robotic vision, the paths of movement of the robot 1,2,3,4,5 were derived, but the movement of the robot system from point P1 to path 5 was not. The optimal path (shortest) from point P1 to circle 5 can be found using the PSO optimization algorithm. First of all, the objective function and constraints on the basis of which the algorithm can find the optimal movement path should be defined.

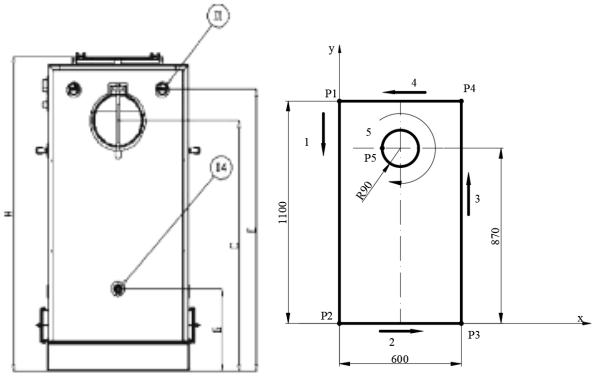


Figure 18. Task analysis in iRVision program of FANUC VISION SYSTEM

6. APPLICATION OF THE PSO ALGORITHM FOR FINDING THE OPTIMAL ROBOT SYSTEM MOVEMENT PATH

Based on the analysis of the geometry and position of the points and the circle, it can be concluded that the objective function must be the minimum path of the robot from point P1 to the circle and it can be written in mathematical form [5]:

$$F(x)=\sqrt{(x_B-x_A)^2-(y_B-y_A)^2} \qquad (7)$$

Point A is point P1 in the picture and point B is the other end of the shortest distance from point P1 to the circle.

There are limitations:

$$x \in (210, 390) \qquad (8)$$

$$y \in (780, 960) \qquad (9)$$

The condition that should be satisfied by x and y is that the equation holds:

$$(x-300)^2+(y-870)^2=90^2 \qquad (10)$$

```
% Memory and screen cleaning
clear;clc;

iterations = 200;
inertia = 1.0;
correction_factor = 2.0;
swarm_size = 3;
nd = 3;
lower_bound = [ 0, 0, 0];
upper_bound = [ 1, 1, 1];
velocity_max = 0.2.*(upper_bound -
lower_bound);

% Memory reallocation
swarm(swarm_size) = particle();

% Initial position of the swarm
for i = 1 : swarm_size
  swarm(i).position = lower_bound + rand(1,
nd).*(upper_bound - lower_bound);
  swarm(i).velocity = rand(1, nd) .*
velocity_max;
  swarm(i).best_position = swarm(i).position;
  swarm(i).best_value =
funkcija_cilja(swarm(i).position);
end

gbest = 1;
number_of_iteration = 1;
while number_of_iteration < iterations
   % Ocenjivanje pozicije
   for i = 1 : swarm_size
      % Azuriranje pozicije
      swarm(i).position = swarm(i).position +
swarm(i).velocity;
      % Ogranicavanje pozicije
      for j = 1 : nd
        if swarm(i).position(j) > upper_bound(j)
           swarm(i).position(j) = upper_bound(j);
        end
        if swarm(i).position(j) < lower_bound(j)
           swarm(i).position(j) = lower_bound(j);
        end
      end

      value = funkcija_cilja(swarm(i).position);
      % Ocenjivanje trenutne pozicije
      if value < swarm(i).best_value
        swarm(i).best_position =
swarm(i).position;
        swarm(i).best_value = value;
      end
```

```
    end
    % Searching for the best position right now
    for i = 1 : swarm_size
      if swarm(i).best_value <
swarm(gbest).best_value
        gbest = i;
      end
    end
    % Azuriranje brzine
    for i = 1 : swarm_size
      swarm(i).velocity = inertia * rand(1, nd) .*
swarm(i).velocity + ...
        correction_factor * rand(1, nd) .*
(swarm(i).best_position - ...
        swarm(i).position) + correction_factor *
rand(1, nd) .* ...
        (swarm(gbest).best_position -
swarm(i).position);
      % Ogranicavane brzine
      for j = 1 : nd
        if abs(swarm(i).velocity(j)) >
velocity_max(j)
          swarm(i).velocity(j) =
sign(swarm(i).velocity(j))* velocity_max(j);
        end
      end
    end
    number_of_iteration = number_of_iteration +
1;
end

swarm(gbest).position
funkcija_cilja(swarm(gbest).position)
```

After running the PSO algorithm, a solution in the form is obtained:

$$(228{,}6-300)^2+(924{,}8-870)^2=90^2$$

$$x_B=228{,}6\,\text{mm}$$

$$y_B=924{,}78\,\text{mm}$$

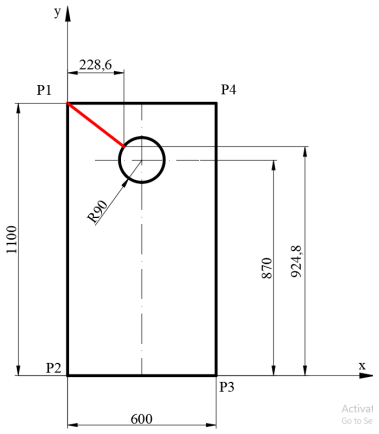


Figure 19. Defining the optimal movement path using the PSO algorithm

7. CONCLUSION

The application of software for programming the movement of robots when performing technological tasks increases the degree of usefulness of the robotic system. iRVision Fanuk's robot system is adapted to work with industrial robots and communicates with all 3D programs. This enables the simulation of the operation of the robot system in real time and the analysis of possible errors in the operation.

For movement that is not defined by the iRVision software, we must determine the direction of movement and the path ourselves. That is why it is best to analyze paths that are not defined by the program with the PSO algorithm in order to obtain optimal paths of movement of the robotic system.

In the work, based on the PSO algorithm, the coordinate of the point on the circle where the robot system should reach when it completes path 4 is obtained. In this way, the shortest path moved by the robot system and the shortest time for completing the task are obtained.

REFERENCES

[1] *FANUC Robot series R-30iB Plus controller iRVision 2D Camera Application*,(2017).

[2] Kenedy, J., Eberhart, R.,(1995). *Particle swarm optimization.* IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.

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

[4] *Oprating manual RobotStudio, ABB Robotics*.

[5] Petrović, Z., Lukić, Lj.,(2011). *Optimization of the parameters of milling machining mode by using the method of particle swarm optimization (PSO)***.** 20th international scientific conference-TRANSPORT 2011