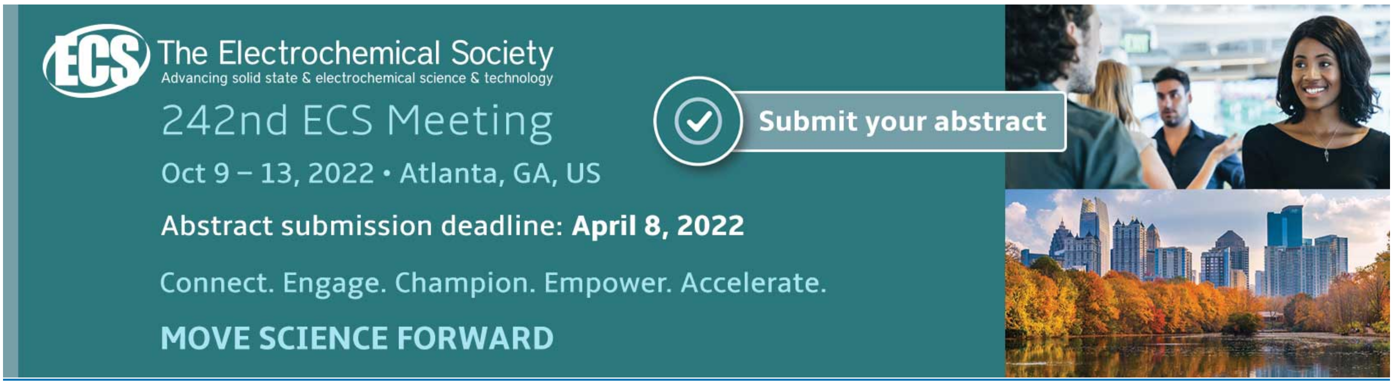
PAPER • **OPEN ACCESS**

Simulation of manipulation task using *i*RVision aided robot control in Fanuc RoboGuide software

To cite this article: V Bulej *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1199** 012091

View the article online for updates and enhancements.



Simulation of manipulation task using *i*RVision aided robot control in Fanuc RoboGuide software

V Bulej[1], M Bartoš[1], V Tlach[1], M Bohušík[1] and D Wiecek[2]

[1] Department of Automation and Production Systems, Faculty of Mechanical Engineering, University of Žilina, Žilina, SK 010 26, SK
[2] Dariusz Wiecek: ATH – University of Bielsko Biala, Institute of Industrial Engineering, 43-309 Bielsko Biala, Willowa 2, PL, wiecekd@ath.bielsko.pl

E-mail: vladimir.bulej@fstroj.uniza.sk, wiecekd@ath.bielsko.pl

Abstract. The article deals with simulation of visual guided robot (VGR) in offline programming software Fanuc RoboGuide. At the beginning there is a brief description of the Fanuc RoboGuide system. The practical part contains an example of the task where the configuration and demonstration of offline programming of visual guided robot system via industrial camera is presented. The main aim of the work is to practically verify the functionality of the system usable for intelligent handling and assembly workplaces.

1. Introduction

Robotics is one of the key branches of automation not only in the mechanical engineering sector. Industrial robots and manipulators have found their application in a wide range of tasks that can perform and where can replace human operators. Modern-generation robots are highly-efficient, highly-capable and complex handling/technological devices, usually equipped by various cognitive and “intelligent” sensors which can help to navigate robots in difficult environment conditions or processes and avoid potential damage of manipulated items or robot itself [1-3]. Here we can mention especially integrated force-torque sensors, scanners, accelerometers, high performance cameras with fast pattern recognition algorithms and other cognitive sensors based on selflearning functions for advanced decision-making as an example. Assembling is one of the fields where robotics is developing at a very high pace.

Simulation and simulation tools play an increasingly important role in the design and implementation of robotic cells and robotic production and assembly lines in recent years. This is due to the ever-increasing complexity of the processes that the customer requires to automate, the effort to avoid possible collisions and thus eliminate additional material costs already in designing phase, but also the need to reduce development time to a minimum. In particular, the latter reason is becoming very topical, as there is not enough time for adequate and “comfortable” development due to strict requirements from the customers, which may result in later technical problems during implementation phase.

Therefore, the engineers try to find adequate methods, which can help to avoid such unwanted situation. One of the possible ways is the massive application of suitable simulation tools. This is also in accordance with the idea of Industry 4.0 [4-6], which seems to be the only possible alternative for modern engineering production in Europe in the future. On the other hand, even in computer simulation, there is visible constant development due to changing the capabilities of robotic and handling equipment, as well as the application of even more and more advanced sensors, conveyors and other auxiliary equipment of robotic workplaces (any robotic workplace is not just a robot itself). Finally, we can assume, that if we want to fully and correctly simulate these workplaces, the simulation should also include the operation of these additional devices. Due to the complexity of the tasks, intelligent types of sensors have been increasingly implemented in recent years, e.g. cameras and intelligent cognitive systems working on the basis of machine vision, possibly with the application of artificial intelligence methods [7], methods for so-called digital twin modelling [8], virtual reality / virtual commissioning [9-11], and the like. The capability of such camera or intelligent systems is nowadays almost unlimited and due to their properties we are now able to solve task, which have been considered impossible or very difficult to automate decades before. Therefore, our research in recent years has focused not only on unconventional kinematic structures of robots, but also camera systems in robotics – so-called visual guided robot (VGR) systems [9]. The output of such solutions can bring high flexibility and robustness at the same time. Such systems can perform specified tasks with high accuracy and reliability [10-14], even under extremely changing conditions. In most previous installations containing VGR robot system, we prepared the basic processes of the workplace in the offline programming mode, but some auxiliary processes (such as the detection of shapes using a camera system) were usually programmed later, already at the real workplace (i.e using the so-called online programming methods). Such an approach had some advantages, but did not allow the full potential of the offline programming method to be exploited. Therefore, we decided to verify and test the possibilities of integrating the simulation of camera systems and whole VGR system directly in the offline programming environment of the Fanuc RoboGuide software.

2. Simulation of robotic cells without downtime

Fanuc RoboGuide is an offline simulation package that simulates both the robot's motion and application commands, significantly reducing the time it takes to create new motion setups (FANUC). To ensure minimal impact on production, cells can be designed with imported CAD tested and modified entirely offline. Built on Virtual Robot Controllers to give you accurate motion and cycle times. Working with the software can be considered as intuitive, however for more advanced application there is necessary to have some practical skills. This was also our main motivation for such project.

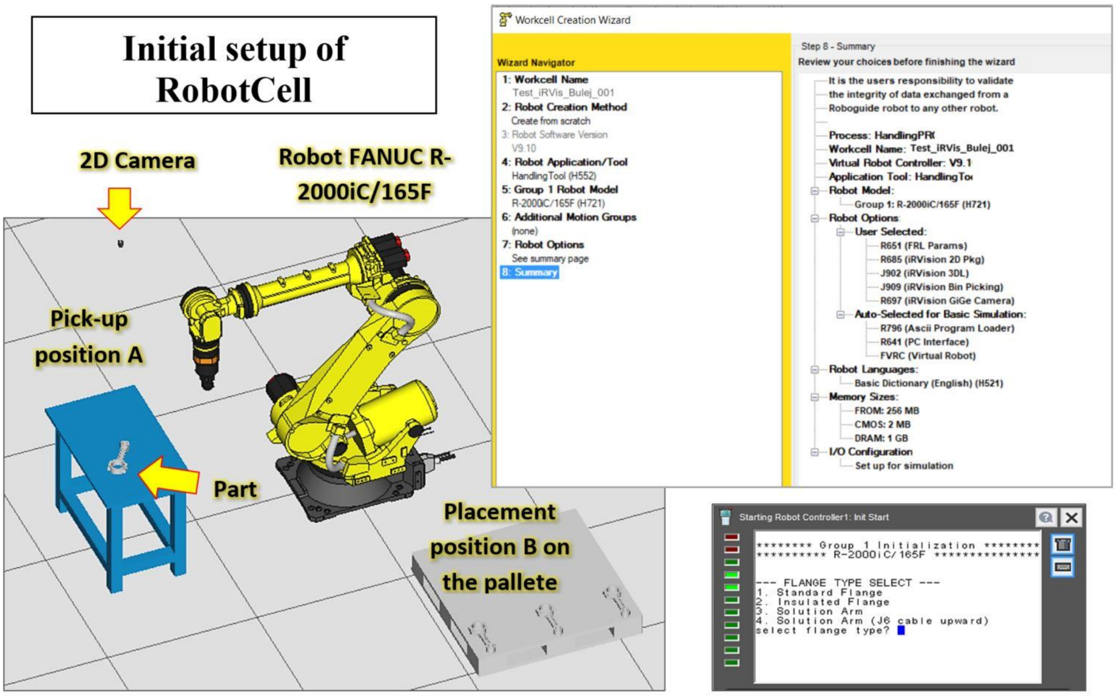
The RoboGuide simulation tool from Fanuc makes it possible to simulate all the functions offered by real robotic controllers, such as handling, welding, measuring and even recognition tasks [15-20]. Similar to the mentioned competition, this software supports, in addition to text programming, also conventional programming using the virtual TeachPendant. The main advantage of the concept of Fanuc control systems (e.g. R-30iA), and thus also the RoboGuide simulation tool, is the possibility to use the integrated *i*RVision camera system. The configuration, learning and also the programming of *i*RVision takes place directly on the TeachPendant robot or in a web browser on a regular PC. The *i*RVision system supports two camera-based architectures for 2D applications, or a combination of camera and laser technology for 3D applications (including so-called bin-picking).

3. Configuration and programming of robot control via *i*RVision in software Fanuc RoboGuide

We start the Fanuc RoboGuide installation in the standard way using the setup.exe file. After confirming the installation of the missing software components and confirming the Fanuc license conditions, it is necessary to mark the required HandlingPRO process plugin. Subsequently, we get to the selection of functions that we intend to use. As we can see in the picture (figure 1), during the installation we chose to install all functions of the program. The most important and crucial for our project was the Bin Pick function. After confirming the creation of links on the desktop, we get to the selection of a virtual robotic controller, whose functions and equipment will be installed in RoboGuide. From the available we choose the latest available controller Fanuc FRVAC V9.1 9.1071.07.04. After entering the type of virtual robotic controller, the actual installation of the FANUC RoboGuide program will take place.

3.1. Initial configuration and creation of testing robot workcell model

After successfully installation of the Fanuc RoboGuide software in specific setup, we can proceed to the creation and configuration of the robotic workplace itself via so-called Workcell Creation Wizard containing 8 steps in total. At the beginning, we proceed ordinary – firstly naming the project; choosing the method of creating the robot cell (since we are creating a new workplace, we choose the HandlingPRO config option); from the offer of robotic controllers, we will select the latest available version (in our case Virtual Robot Controller V9.1); then we select the HandlingPRO process with the default HandlingTool H552 for our application. Furthermore, the selected robot model depends on the task conditions, mainly requirements for maximum reach and payload capacity and the manipulated part required degrees of freedom (DOF). In our project we decided for industrial robot with serial kinematic structure R-2000iC/165F (H721).


Initial setup of RobotCell
2D Camera
Robot FANUC R-2000iC/165F
Pick-up position A
Part
Placement position B on the pallete
Workcell Creation Wizard
Wizard Navigator
Starting Robot Controller1: Init Start

1: Workcell Name
Test_iRVis_Bulej_001
2: Robot Creation Method
Create from scratch
3: Robot Software Version
V9.10
4: Robot Application/Tool
HandlingTool (H552)
5: Group 1 Robot Model
R-2000iC/165F (H721)
6: Additional Motion Groups
(none)
7: Robot Options
See summary page
8: Summary

Step 8 - Summary
Review your choices before finishing the wizard

```
It is the users responsibility to validate
the integrity of data exchanged from a
Roboguide robot to any other robot.

Process: HandlingPR(
Workcell Name: Test_iRVis_Bulej_001
Virtual Robot Controller: V9.1
Application Tool: HandlingToı
Robot Model:
    Group 1: R-2000iC/165F (H721)
Robot Options:
    User Selected:
        R651 (FRL Params)
        R685 (iRVision 2D Pkg)
        J902 (iRVision 3DL)
        J909 (iRVision Bin Picking)
        R697 (iRVision GiGe Camera)
    Auto-Selected for Basic Simulation:
        R796 (Ascii Program Loader)
        R641 (PC Interface)
        FVRC (Virtual Robot)
Robot Languages:
    Basic Dictionary (English) (H521)
Memory Sizes:
    FROM: 256 MB
    CMOS: 2 MB
    DRAM: 1 GB
I/O Configuration
    Set up for simulation
```

```
******** Group 1 Initialization ********
********** R-2000iC/165F ***************

--- FLANGE TYPE SELECT ---
1. Standard Flange
2. Insulated Flange
3. Solution Arm
4. Solution Arm (J6 cable upward)
select flange type?
```

Figure 1. Initial setup of RobotCell in Fanuc RoboGuide software.

The most important configuration step when using *i*RVision system is selecting the right application packages in "Robot Options" (step 7). The figure (figure 1, right) shows all the necessary application packages for the correct function of the *i*RVision system, these are:

- R651 FRL Params,
- R685 iRVision 2D Pkg,
- J902 iRVision 3DL,
- J909 iRVision Bin Picking,
- R697 iRVision GiGe Camera.

After completing the RobotCell configuration, we will start the RoboGuide and let the standard initialization takes place, as in the case of a real robot. For the first initialization it is necessary to select the robot flange type as obvious in the bottom of figure 1 (we chose the standard flange configuration). After that the virtual robotic controller will start and the TeachPendant will be displayed.

For our experimental project there was designed a simplified simulation model of whole robotic cell including (see figure 1, left):

- pick-up position A - where the robot will pick-up the manipulated object, in our case piston rod,
- placement position B on the pallet – where the robot releases the object after manipulation,
- 2D camera statically placed over the pick-up position A,
- part – area where the parts will be automatically generated at random position / orientation.

To remove and generate a random location of the manipulated part, we selected a standard table from the drag-and-drop menu: Cell - add Fixtures - CAD library - fixtures - table - table with legs. Other necessary configuration for part random generation will be setup later. For storage, we chose a pallet from the category of fixtures - pallets - Plt_1100X1100.

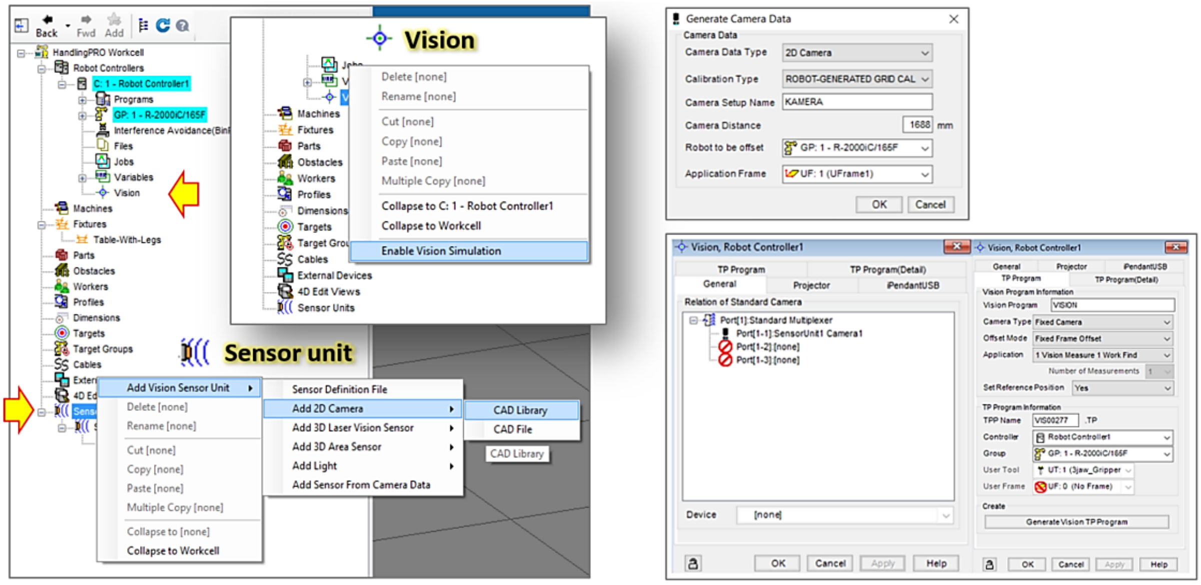
3.2. Basic configuration of robot controller with FANUC iRVision 2D

The first step in configuration of any vision system in RoboGuide software is to enable the Vision feature in the robotic controller properties. Enabling can be done in the project tree under the tab Robot Controller1 - Vision [right click] - Enable Vision Simulation (figure 2, left).

When designing a workplace, it is necessary to choose from the two types of camera location: statically placed (fixed to the main frame) or dynamically placed (mounted on the robot arm or end-effector). Since our application deals with the manipulated object picking-up from a defined zone of the worktable, which roughly corresponds to camera the field of view (FOV), it is not necessary to extend camera range by carrying it on the robot arm – so we choose a fixed location above the worktable.

The camera model can be selected from left menu: Sensor units - Add Vision Sensor Unit - Add 2D Camera - CAD Library (figure 2, left). From the model library, we select the appropriate type of camera, which we then place above the pick-up table at a sufficient height to allow the movement of the robotic arm. It is needed only to name the camera and the lens distance from the pick-up table surface in the camera properties window (figure 2, right up) at the current phase due to these data are necessary to generate Camera Data for Vision TP program. Then we will return to the camera settings.

In the Cell Browser (left menu), select following path: RobotController - Vision - Vision Properties, which will take you to the Vision system settings menu in the robotic controller (figure 2 on the right down). On the General tab, set the camera connection port Port (1): Standard Multiplexer - Port (1-1) SensorUnit1 / Camera1. On the TP Program tab, set the camera type to Fixed Camera, offset mode to Fixed Frame Offset. Select the application 1Vision Measure 1 Work Find. This application works by creating an image of the trigger signal on the camera and identifying just one match with the trained model. Basic information about the program itself, such as name, relevant robotic controller, motion group, tool and user frame are shown at the bottom of the card. As a last step, we generated the program using the Generate Vision TP Program button.



Generate Camera Data

Camera Data

Camera Data Type: 2D Camera
Calibration Type: ROBOT-GENERATED GRID CAL
Camera Setup Name: KAMERA
Camera Distance: 1688 mm
Robot to be offset: GP: 1 - R-2000iC/165F
Application Frame: UF: 1 (UFrame1)

Vision, Robot Controller1

Relation of Standard Camera

Port[1]:Standard Multiplexer
Port[1-1]:SensorUnit1 Camera1
Port[1-2]:[none]
Port[1-3]:[none]

Device: [none]

Vision, Robot Controller1

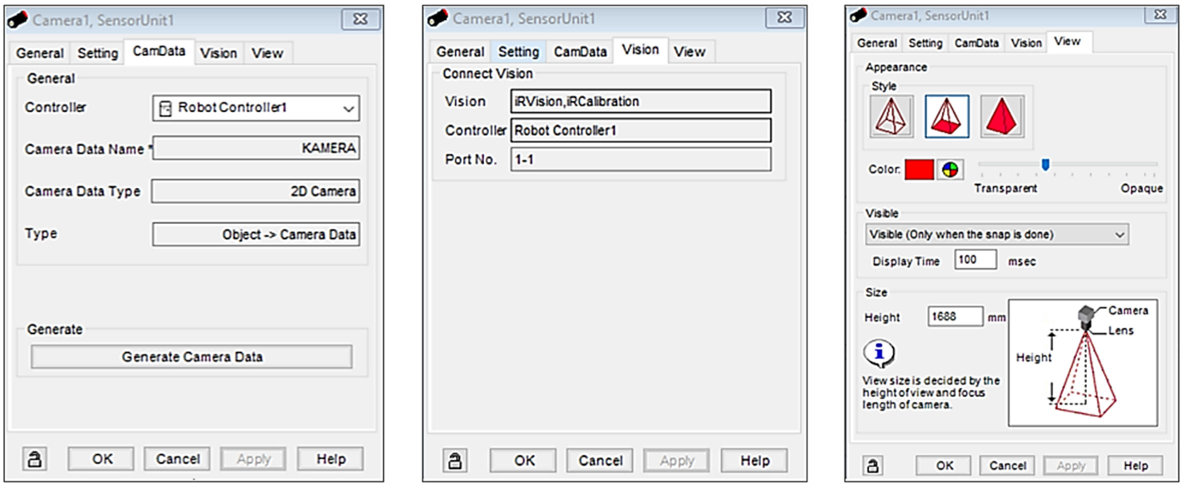
Vision Program Information
Vision Program: VISION
Camera Type: Fixed Camera
Offset Mode: Fixed Frame Offset
Application: 1 Vision Measure 1 Work Find
Number of Measurements: 1
Set Reference Position: Yes

TP Program Information
TPP Name: VIS00277 .TP
Controller: Robot Controller1
Group: GP: 1 - R-2000iC/165F
User Tool: UT: 1 (3jaw_Gripper
User Frame: UF: 0 (No Frame)

Create

Figure 2. Basic configuration of robot controller with FANUC iRVision 2D: Cell Browser – Enabling Vision Simulation and adding new Vision Sensor Unit from CAD library (left), Generate Camera Data tab (right up), Vision Robot Controller setup (right bottom).

After generating the Vision TP program, we return to the camera properties, where on the CamData tab (figure 3 - a) click on the Generate Camera Data button and then get the dialog box shown in the figure. Here it is necessary to set the Calibration Type to the Robot-Generated Grid Cal option and select the Camera Setup Name, under which we will find these settings in the Vision application (we named it KAMERA). In the tab Vision should be defined the type of Vision (in our case iRVision, iRCalibration), robot Controller and Port number for communication. In the last tab, View, can be defined how the camera FOV will be displayed, as well as visibility settings, display time and working height of the camera. With these steps, we have successfully completed the addition and setting of camera properties, the creation of the Vision TP program and the generation of Camera Data. All these steps are necessary for the correct communication of the camera with the Vision application.



General
Controller: Robot Controller1
Camera Data Name *: KAMERA
Camera Data Type: 2D Camera
Type: Object -> Camera Data

a)

Connect Vision
Vision: iRVision,iRCalibration
Controller: Robot Controller1
Port No.: 1-1

b)

Color: 🟥
Visible: Visible (Only when the snap is done)
Display Time: 100 msec
Size
Height: 1688 mm
View size is decided by the height of view and focus length of camera.

c)

Figure 3. Basic camera setup process for “Camera1” sensor unit via three tabs: a – CamData tab, b – Vision tab, c –View tab.

3.3. *Designing of vision application via iRVision Vision Setup – camera calibration*

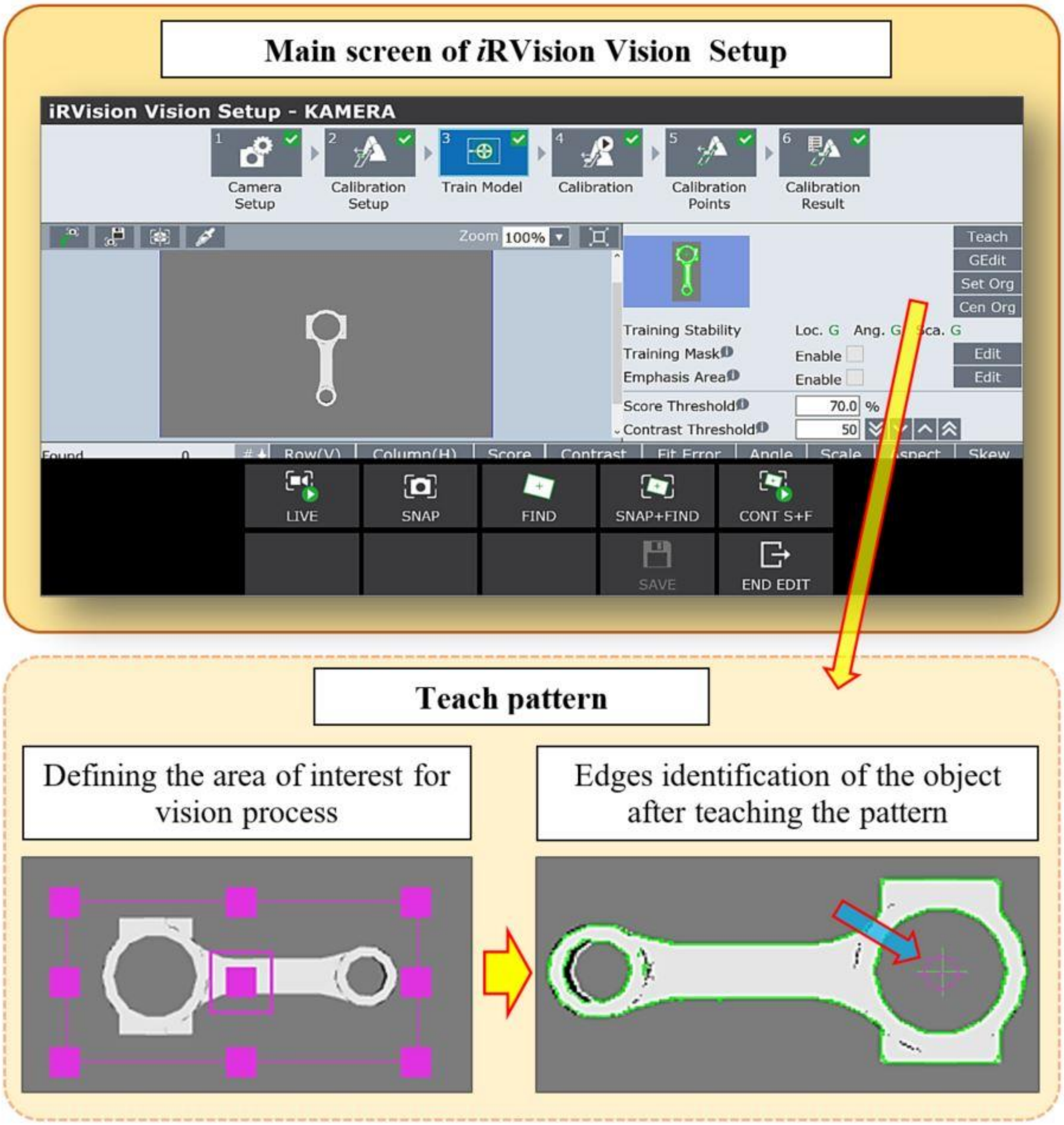
After performing the assignment, location and configuration of the camera, we get to the recognition and location settings for application itself. One of the Fanuc system advantage is that the application can be configured directly in the TeachPendant or via an internet browser (so-called WEBserver) on a standard computer without the need for specialized software. Fanuc requires the use of Internet Explorer, which can be started in the RoboGuide environment by selecting it from the Robot - Internet Explorer tab. For the application to function properly, it is necessary to add the address of the virtual robotic controller to the trusted sites in Internet Explorer. After the next opening of Internet Explorer in the RoboGuide environment, we will get the ROBOT Homepage.

ROBOT Homepage or otherwise also Robot browser offers us access to all vision applications, robot status, but also to programs and their diagnostics. For our application, the most important is the Vision Setup, which is located on the first selection tile. Before starting to teach the camera, it is necessary to place the manipulated part within the camera FOV (its view area).

After entering the Vision Setup procedure, we can see the Camera Data called KAMERA, which was generated in previous step (see figure 3 - a). However, some data still needs to be filled manually. By double-clicking on Camera Data, we get information about the camera settings and details of automatic calibration, which are distributed in six tabs. On the Camera Setup tab, you will find information about the camera and the calibration method. Then it is necessary to check camera type and set that we are not using a robot mounted camera. Since the camera is calibrated, there can be selected Robot-Generated Grid Cal. The corresponding User Frame aligned with our pick-up worktable should be selected from the Calibration Setup tab. With this step, the camera calibration can be considered completed.

3.4. Designing of vision application via iRVision Vision Setup – vision process tool

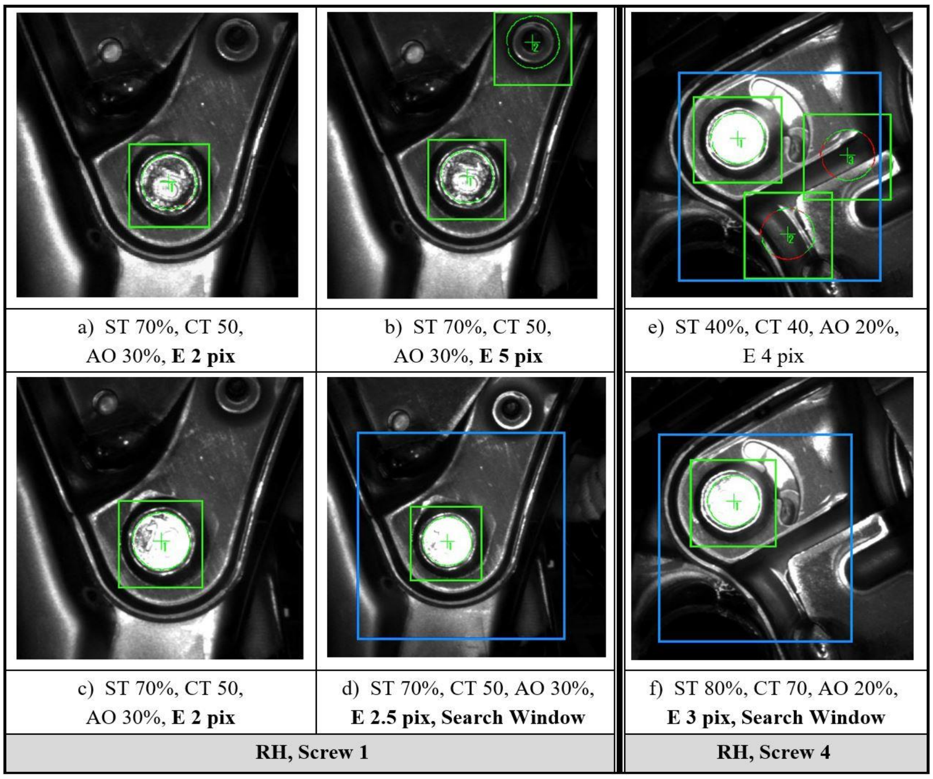
The Train Model card (figure 4) is the first point where the camera system is taught for our exact task. First you need to get the training image of our manipulated object with the SNAP button. Use the Teach button to get to the graphic window, in which we use the crop tool to delimit the component (figure 4 – bottom left). Use the Set Org button to get to determine the origin of the part's coordinate system via mouse click or moving by arrow keys on the keyboard figure 4 – bottom right). The coordinates that the robot will receive from the camera system relate to this point and basically it corresponds to the touch point for gripping the component.



Zoom 100%
Training Stability Loc. G Ang. G Sca. G
Training Mask Enable □
Emphasis Area Enable □
Score Threshold 70.0 %
Contrast Threshold 50
Found 0 | # | Row(V) | Column(H) | Score | Contrast | Fit Error | Angle | Scale | Aspect | Skew

Figure 4. Main screen of iRVision Vision Setup and Tech pattern process.

The next step is the so-called Vision Process Tool setup. We add the new vision process tool by clicking on the CREATE icon, fill in the process name (in our case Piston rod or "ojnica") and select the process type 2-D Single-View Vision Process. After opening it by double-clicking, it is necessary to set the camera Offset Mode and Offset Frame in the 2-D Single-View Vision Process tab. In this tab, we also set the height of the part that we obtained by measuring (height over the worktable top surface). On the Snap Tool 1 tab, we set the Snap Window, which determines the area in which the object will be searched, the exposure time and the type of additional lighting (in our case is not applied). We have added a new GPM Locator Tool via the Add button. The GPM process doing the search for the same shape as captured in the sample image, or as was trained. In addition to this process, the Fanuc *i*RVision system offers a number of other processes, e.g. based on the search for colors, lines, transitions between the shadow and the illuminated part of cylindrical parts, reading texts, QR codes, etc. After assigning the GPM process, we set the Snap Tool 1 as the image source. With the TEACH button we got to the screen in which we used the mouse to set the border of the searched part to ensure part recognition. We got the exact position of the part by setting its origin point. We got this setting by pressing the Set Org button. After pressing it, we got an analogous screen to teaching phase, however with small difference. Now, we do not move the field but the axis-cross, which marks the zero point of the part. After these operations, it is necessary to return to the process settings where we get the image by the SNAP button.


a) ST 70%, CT 50, AO 30%, **E 2 pix**

b) ST 70%, CT 50, AO 30%, **E 5 pix**

e) ST 40%, CT 40, AO 20%, E 4 pix

c) ST 70%, CT 50, AO 30%, **E 2 pix**

d) ST 70%, CT 50, AO 30%, **E 2.5 pix, Search Window**

f) ST 80%, CT 70, AO 20%, **E 3 pix, Search Window**

RH, Screw 1

RH, Screw 4

Figure 5. An example of real influence of Vision Process Tool parameters on part recognition (other real project - robotized screwing application): ST - Score Threshold, CT - Contrast Threshold, AO - Area Overlap, E – Elasticity, NONE - Search Window (according to Saga et al, 2020 / [1]).

Then we use the FIND function and set the RefPos Status parameter by pressing the appropriate Set button. Successfully done training and setup process is shown by the text displayed as green. There we have a number of other qualitative parameters and options that affect the recognition process sensitivity, efficiency, stability, quality and repeatability. The real influence of these parameters is shown in figure 5, where is presented an example from another our project – finding the screw heads.

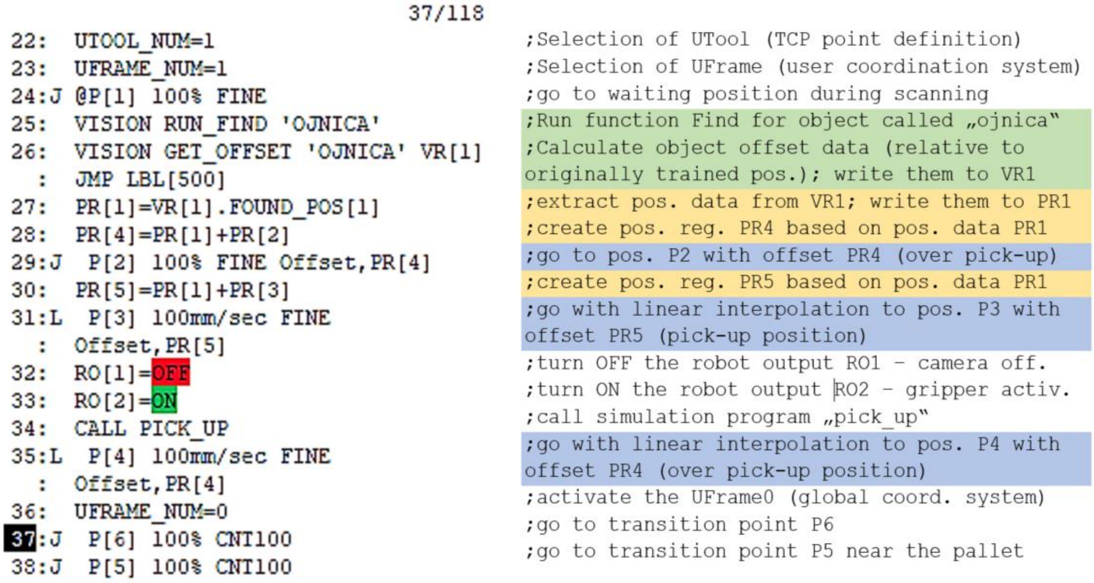
We can set the following parameters:

- sensitivity of the component search with the Score Threshold (ST) parameter – level of the pixel matches between the training and current image (if the match percentage is lower than the set threshold, the image will be rejected and evaluated as non-matched),
- the Contrast Threshold (CT) parameter refers to the value of the contrast required to detect the object - this parameter is more common used in real applications where contrast effects, such as surface finishes, dust and variable lighting conditions; we do not consider any effects in the simulation and left the parameter in the basic settings,
- Area Overlap (AO)- is a parameter by which we set the threshold value of the accepted overlap of searching object by any other obstacles or objects (if there is accepted any overlap or not),
- The Elasticity (E) parameter, is the value in pixels by which the geometry of the object may differ from the learned model - in practice, this parameter makes it possible to solve problems caused by changing the distance of the object from the focus of the camera, which changes the perspective, i.e. enlargement / reduction of the object, or deformation of its apparent geometry.
- The last set parameter is the Search Window - where we determine / crop the area in which the component is searched.

After setting the parameters, we returned to the Camera Data application, performed the calibration according to the instructions on the Calibration tab, obtained the calibration points and their coordinates on the Calibration Points tab. The last tab, Calibration Results, offers summary information, among other things, for the areas of cameras against the User Frame that we use in the program.

3.5. *Control program design*

Prerequisite for successful implementation of the *i*RVision system are the general knowledge of Fanuc robot advanced programming using the Teach Pendant. Therefore, we will focus only on program blocks aimed with the vision system processing due to the range of the article. The program can be divided into several blocks that are responsible for each specific function, e.g. the first complete block is the control of the robot system and its initialization (registers initialization, testing the gripper status, User Frame and User Tool setting, etc.). The next comprehensive, and for us the most important, is the block responsible for controlling the Vision system. The last one is responsible for placing the parts on the pallet and returning the robot to the home position.



```
  UTOOL_NUM=1                         ;Selection of UTool (TCP point definition)
  UFRAME_NUM=1                        ;Selection of UFrame (user coordination system)
J @P[1] 100% FINE                     ;go to waiting position during scanning
  VISION RUN_FIND 'OJNICA'            ;Run function Find for object called „ojnica“
  VISION GET_OFFSET 'OJNICA' VR[1]    ;Calculate object offset data (relative to
  JMP LBL[500]                        originally trained pos.); write them to VR1
  PR[1]=VR[1].FOUND_POS[1]            ;extract pos. data from VR1; write them to PR1
  PR[4]=PR[1]+PR[2]                   ;create pos. reg. PR4 based on pos. data PR1
J  P[2] 100% FINE Offset,PR[4]        ;go to pos. P2 with offset PR4 (over pick-up)
  PR[5]=PR[1]+PR[3]                   ;create pos. reg. PR5 based on pos. data PR1
L  P[3] 100mm/sec FINE                ;go with linear interpolation to pos. P3 with
  Offset,PR[5]                        offset PR5 (pick-up position)
  RO[1]=OFF                           ;turn OFF the robot output RO1 - camera off.
  RO[2]=ON                            ;turn ON the robot output RO2 - gripper activ.
  CALL PICK_UP                        ;call simulation program „pick_up“
L  P[4] 100mm/sec FINE                ;go with linear interpolation to pos. P4 with
  Offset,PR[4]                        offset PR4 (over pick-up position)
  UFRAME_NUM=0                        ;activate the UFrame0 (global coord. system)
J  P[6] 100% CNT100                   ;go to transition point P6
J  P[5] 100% CNT100                   ;go to transition point P5 near the pallet
```

Figure 6. Control program design – detail of block responsible vision system control and obtaining the location data of the searched object.

In the picture (figure 6) we can see a listing of the program block that is responsible for the Vision function including comments. Basically we can describe its function as follows: line 22 set User Tool = 1, so upload the appropriate TCP (Tool Center Point) data for applied gripper. The following line analogously carries the settings of the coordination system of the pick-up table. The camera works in this coordination system and also the robot receives the coordinates of the position of the part. Point P[1] is the so-called home position of the robot, in which the arm of the robot is outside the search camera FOV. Line 25, the Vision Process Tool called Piston rod or "ojnica" is started. Vision tool processes run in the background while the obtained data are imported on line 26 into so-called Vision Register (VR). In case of unsuccessful identification and localization of the searched pattern, the program is stopped and the error message is announced. Line no. 27, we assign the coordinates obtained from the Vision tool to the position register. This is followed by a shift of the position in the positive direction of the Z axis by adding the position register, which has a non-zero positive value in the Z axis. It is essential to reach this position precisely due to the subsequent linear movement into the hole of the part so that no collisions occur. Since the settings of the vision system show that the origin / gripping point of the part is located on its upper surface, it was necessary to transform the pick-up position in the direction of the Z axis by the value of the part thickness. We ensured this by operating with registers on line 30. This is followed by the actual approach to the pick-up point, where the robotic outputs are set to activate the gripper. The CALL PICK_UP command is a simulation command which ensures that the part is anchored to the gripper in the visualization. This is followed by a linear movement to the point P[4], which is transformed by a position register PR[4] identical to the approach position. The subsequent two passage points serve to avoid a collision with the pick-up table and to align the sixth axis of the robot so that the position is not reached when deposited on the pallet.

4. Configuration of simulation program and practical verification in simulation space

The importance of the simulation configuration lies in defining the behavior of the part during the program running. Here we define the simulation behavior of random part generator on pick-up table. By placing a call to the PICK_UP command, we assign the part to the robot end-effector while by calling the PLACE command, we release the part in the storage location (pallet). However, it should be borne in mind that all these steps are not necessary for the real application, but only for the simulation itself.

4.1. Control program design

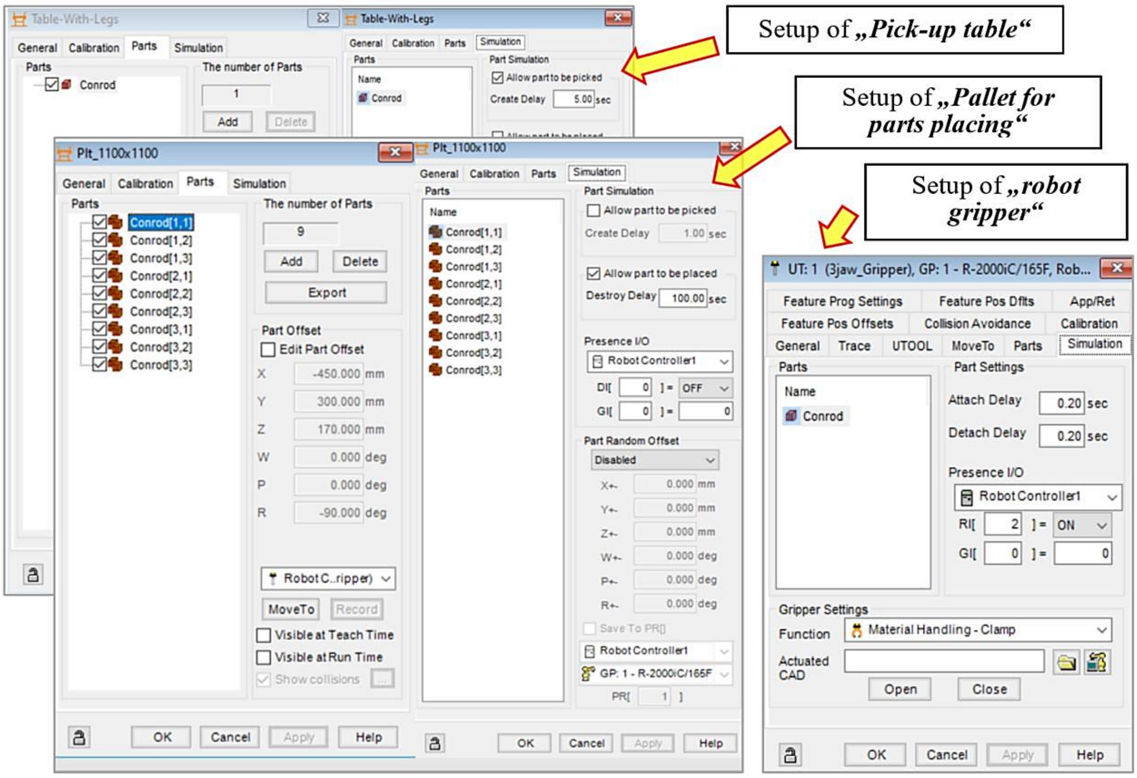
The simulation settings for the pick-up table are shown in figure 7 – left rear. On the Parts tab, we mark the relevant simulation object, in our case piston rod called "ojnica" / "Conrod". We chose the number of objects per 1 piece and determined the coordinates of the object location. In the drop-down menu, we selected Robot Gripper and define object visibility at Teach Time and at Run Time. On the Simulation tab, we activate the option Allow parts to be picked and define the delay time between removal and the creation of a new object for 5 seconds. Subsequently, in the Part Random Offset section, we determined the intervals and the individual axes of the Cartesian coordination system, in the range of which there will be randomly generated new occurrences of our object (e.g. with different orientation about Z).

4.2. Robotic gripper simulation configuration

The settings for robotic gripper are shown in figure 7 – left front. In the Part Settings section, we set the delay time for removing and postponing the object to and from the gripper. In the Gripper Settings section, we can set the function to: Material Handling - Clamp, which means that the tool is designed for manipulating objects with the help of a mechanical gripper.

4.3. Stacking pallet simulation configuration

The last one are the settings for the storage pallet (figure 7 – right), where on the Parts tab we can find objects that will be stored on the pallet by the robot; in the Part Offset section we set its coordinates for each object while on the Simulation tab we activated option Allow part to be placed and set the Time for automatic part removal.



Setup of *„Pick-up table“*

Setup of *„Pallet for parts placing“*

Setup of *„robot gripper“*

Table-With-Legs — Parts: ☑ Conrod; The number of Parts: 1

Table-With-Legs — Simulation — Part Simulation: ☑ Allow part to be picked; Create Delay 5.00 sec

Plt_1100x1100 — Parts: ☑ Conrod[1,1], ☑ Conrod[1,2], ☑ Conrod[1,3], ☑ Conrod[2,1], ☑ Conrod[2,2], ☑ Conrod[2,3], ☑ Conrod[3,1], ☑ Conrod[3,2], ☑ Conrod[3,3]; The number of Parts: 9

Part Offset: □ Edit Part Offset; X -450.000 mm; Y 300.000 mm; Z 170.000 mm; W 0.000 deg; P 0.000 deg; R -90.000 deg

□ Visible at Teach Time; □ Visible at Run Time; ☑ Show collisions

Plt_1100x1100 — Simulation — Part Simulation: □ Allow part to be picked; Create Delay 1.00 sec; ☑ Allow part to be placed; Destroy Delay 100.00 sec

Presence I/O: Robot Controller1; DI[0] = OFF; GI[0] = 0

Part Random Offset: Disabled; X+- 0.000 mm; Y+- 0.000 mm; Z+- 0.000 mm; W+- 0.000 deg; P+- 0.000 deg; R+- 0.000 deg

□ Save To PR[]; Robot Controller1; GP: 1 - R-2000iC/165F; PR[1]

UT: 1 (3jaw_Gripper), GP: 1 - R-2000iC/165F, Rob... — Simulation — Parts: Conrod

Part Settings: Attach Delay 0.20 sec; Detach Delay 0.20 sec

Presence I/O: Robot Controller1; RI[2] = ON; GI[0] = 0

Gripper Settings: Function Material Handling - Clamp; Actuated CAD

Figure 7. Configuration of simulation program: simulation settings for the pick-up table (left rear), settings for robotic gripper (left front) and settings for the storage pallet (right).

4.4. Create and use simulation control commands in the program

The simulation settings described above must be supplemented with simulation control commands placed within the program where the object is picked-up or released. These commands control the graphical interface of the software and allow dynamic graphical display of the manipulated object and change its position.

On the Teach tab, we select the Add Simulation Program command. In our simulation, we are removing only one part from the pick-up table within one cycle. Therefore, we created one pick-up command named PICK_UP and configured as follows. Firstly, we have selected the Pick-up instruction on the Inst tab and then added the attributes - the first line shows the object we are manipulating, the second one is the place of origin, which in our case is the pick-up table, the third line binds the robot tool with which we are manipulating the object, and to which the object will be assigned after the execution of this command.

Analogously, it was necessary to create a simulation command for placing an object on a storage pallet. However, the paging palette has nine different positions, so we had to create nine paging orders, one for each position. After adding the command on the Inst tab, we selected the Drop instruction and add attributes such as the manipulation object, the object origin (this time the robot's gripper), and the target location. We must pay attention to the selection of the correct position on the pallet so that the object is released from the gripper to the right position in which the robot is physically located. We have to perform this sequence of settings for each position on the palette, so the result is nine commands to save the object - one for each position on the palette. When programming, it is necessary to ensure that a command belonging to the given position is invoked anytime the object is placing on the pallet. In our program, we solved this by a closed loop with a register, the value of which was incremented every program run. Based on its current value, the corresponding position register for storage position was used and the corresponding storage order was invoked.

5. Final simulation and practical verification

Finally, we entered the phase of practical verification of the functionality of the created simulation. The Control-Run - Run Panel is used to control the entire workplace simulation. In its upper part there are quick control buttons, the graphical representation of which allows intuitive control of the simulation. In other sections of the panel, it is possible to set attributes such as the simulation speed, the frequency of information refresh, but also the display of curves that describe the TCP movement during the simulation run. Figure 8 presented the time-laps from final simulation and functional verification. There is obvious, how the robot detects the part via camera overtop (b), go to over the part (c), go down to fixed the part via gripper (d), manipulate the part (e) and finally go to the pallet position (f). At the end the robot completed whole loop and come back to the home position where it is waiting for new part.

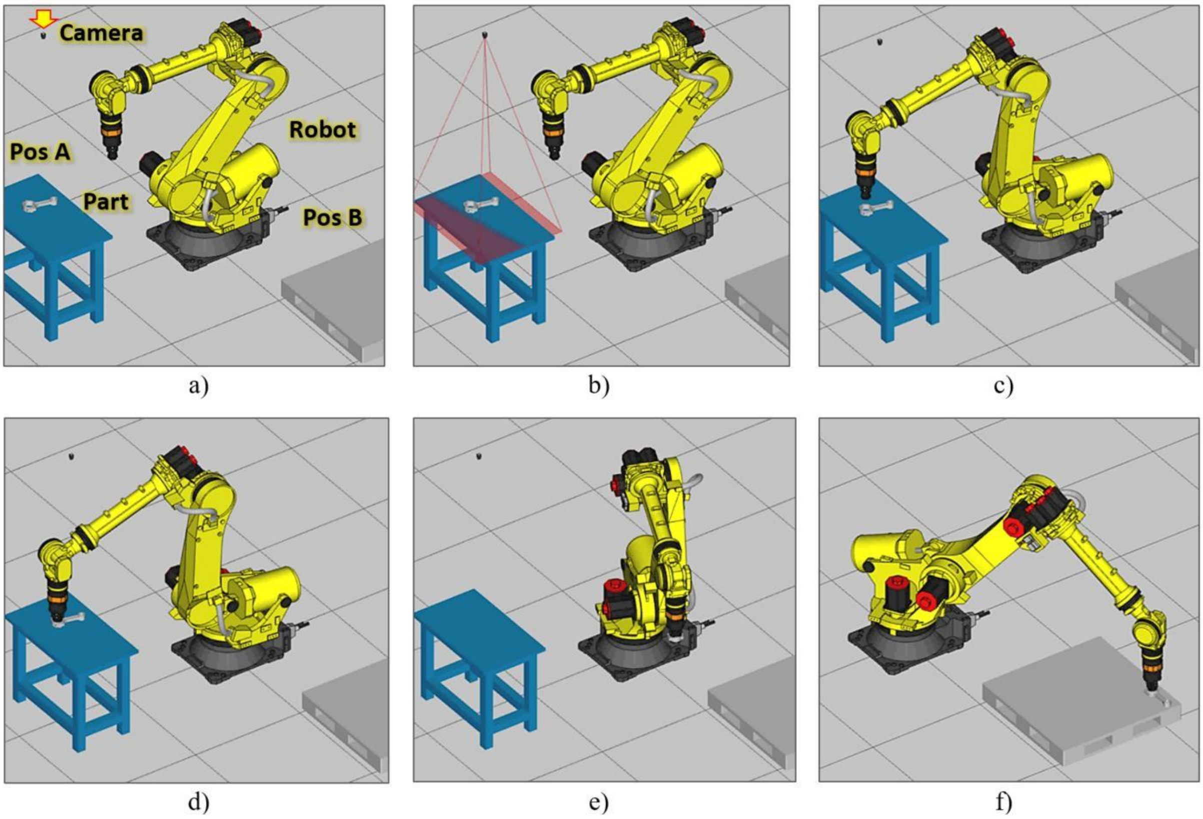


Figure 8. Time-lapse from final simulation and functionality verification: a – workplace arrangement, components and starting position, b – scanning part position via camera, c – robot moving over the pick-up position A, d – gripping part, e – moving from position A to position B, f – part is placed on the pallet, position B.

6. Conclusions

The article dealt with the design, creation, configuration, programming and subsequent simulation of a robotic workcell of intelligent manipulation with the support of the Fanuc *i*RVision system in offline simulation software Fanuc RogoGuide. The core chapters describe the experimental research project focused on demonstration the capability of manipulation task based on visual guided robot system in Fanuc RoboGuide software, where the object of manipulation (piston rod) is detected by 2D camera, the coordinates X and Y as well as orientation about axis Z is calculated to navigate the gripper to pick-up this object. Furthermore, it can be used as a comprehensive guide for simulating a robotic application supported by the *i*RVision camera system as well. There were presented the following processes in this article:

- configuration of the robot workplace robot cell - from the creation of the project through the implementation of the robot and support devices to the implementation of manipulated objects
- robot configuration and vision application - setting up communication between the robot and the vision application, camera setup, training the searching pattern, teaching the application, and clarification of individual parameters influencing the process of object recognition and localization.
- programming and simulation of a robotic workplace - robot software for data processing from the *i*RVision application and their subsequent use for guiding the robot in space in order to achieve a random and variable position of the object. Simulation of the created workplace in order to verify the results of previous settings.

In the future work we plan to extend our task to objects randomly located not only on the two-dimensional surface but also in three-dimensional space with the help of higher-order technologies. Last but not least, we consider it a benefit for practice to apply the simulation in the laboratory on a real device and thus verify the usability and precision of off-line settings in a real environment.

Acknowledgement

The authors disclosed receipt of the following financial support for the research, authorship, and publication of this article: This work is partially supported by the project APVV-16-0283: “Research and development of multi-criteria diagnosis of production machinery and equipment based on the implementation of artificial intelligence methods” and by grant agency VEGA, project No. 1/0864/21

References

1. Saga M, Bulej V, Cubonova N *et al* 2020 Case study: Performance analysis and development of robotized screwing application with integrated vision sensing system for automotive industry *Int. J. Adv. Robot. Syst.* **17**(3) 1-23 DOI: 10.1177/1729881420923997
2. Bulej V, Stancek J and Kuric I 2018 Vision guided parallel robot and its application for automated assembly task *Advances in Science and Technology-Research Journal* **12**(2) 150-157 DOI: 10.12913/22998624/91887
3. Svetlik J, Baron P, Dobransky J and Kocisko M 2014 Implementation of Computer System for Support of Technological Preparation of Production for Technologies of Surface Processing *Applied Mechanics and Materials* **613** 418 DOI: 10.4028/www.scientific.net/AMM.613.418
4. Svetlik J and Demec P 2013 Methods of Identifying the Workspace of Modular Serial Kinematic Structures *Applied Mechanics and Materials* **309** 75-79 DOI: 10.4028/www.scientific.net/AMM.309.75
5. Ruzarovsky R, Holubek R, Sobrino DRD *et al* 2018 The Simulation of Conveyor Control System Using the Virtual Commissioning and Virtual Reality. *Advances in Science and Technology-Research Journal* **12**(4) 164-171 DOI: 10.12913/22998624/100349
6. Holubek R, Ruzarovsky R and Sobrino DRD 2019 An Innovative Approach of Industrial Robot Programming Using Virtual Reality for the Design of Production Systems Layout. *6th Int. Sci.-Technical Conf. on Advances in Manufacturing II – Advances in Manufacturing II, Vol 1 – Solutions for Industry 4.0* **1** 223-235 DOI: 10.1007/978-3-030-18715-6_19
7. Dodok T, Cubonova N and Kuric I. Workshop programming as a part of technological preparation of production. *Advances in Science and Technology-Research Journal* **11**(1) 111-116 DOI: 10.12913/22998624/66504
8. Velisek K, Holubek R, Sobrino DRD *et al* 2017 Design of a robotized workstation making use of the integration of CAD models and Robotic Simulation software as way of pairing and comparing real and virtual environments *4th Int Conf on Comp and Solut in Man Eng 2016 - COSME'16, MATEC Web of Conf.* **94** UNSP 05008, DOI: 10.1051/matecconf/20179405008
9. Labudzki R, Legutko S, Raos P 2014 The essence and applications of machine vision *Tehnicki vjesnik-Technical Gazette* **21**(4) 903-909
10. Taghirad HD 2013 Parallel robots - mechanics and control. 1st edn. CRC Press, London

11. Tlach V, Cisár, M.; Kuric, I. et al.:Determination of the Industrial Robot Positioning Performance. Modern Technologies in Manufacturing (MTEM 2017 - AMATUC) Book Series: MATEC Web of Conferences Volume: 137 Article Number: UNSP 01004, 2017, Cluj Napoca
12. Kuric I, Tlach V, Saga M *et al* Industrial Robot Positioning Performance Measured on Inclined and Parallel Planes by Double Ballbar. *Appl Scie – Basel* **11**(4) 1777
13. Kelemen, M., Virgala, I., et al.: A novel approach for a inverse kinematics solution of a redundant manipulator. Appl Sci-Basel 8(11), 1-14 (2018). DOI: 10.3390/app8112229
14. Saga, M., Vasko, M., Handrik, M., Kopas, P.: Contribution to random vibration numerical simulation and optimization of nonlinear mechanical systems. Scientific journal of Silesian University of Technology - Series Transport 103, 143-154 (2019)
15. Kuric I, Klačková I, Nikitin YR, et al 2021 Analysis of Diagnostic Methods and Energy of Production Systems Drives. Processes 9 DOI: 10.3390/pr9050843
16. Kuric I, Gorobchenko O, Litikova O, et al (2020) Research of vehicle control informative functioning capacity. IOP Conf Ser Mater Sci Eng 776:. https://doi.org/10.1088/1757-899X/776/1/012036
17. Hrubos M, Svetlik J, et al. 2016 Searching for collisions between mobile robot and environment *Int. J. of Adv. Robotic Systems* **13**(5) DOI10.1177/1729881416667500
18. Kelemen M Virgala I et al. 2018 A novel approach for a inverse kinematics solution of a redundant manipulator *Appl Scie - Basel* **8**(11) 1-14 DOI: 10.3390/app8112229
19. Stejskal T Dovica M Svetlik J et al. 2019 Experimental assessment of the static stiffness of machine parts and structures by changing the magnitude of the hysteresis as a function of loading *Open Eng.* **9**(1) 655-659 DOI10.1515/eng-2019-0078
20. Virgala I, Kelemen M, Prada E *et al* 2021 A snake robot for locomotion in a pipe using trapezium-like travelling wave *Mechanism and Machine Theory* **158** 104221 DOI: 10.1016/j.mechmachtheory.2020.104221