SENSOR APPLICATION IN ONLINE ROBOT PROGRAMMING

Dipl. Ing. Haris Lepić
Prof. Dr.Sc. Malik Čabaravdić
University of Zenica, Faculty of Mechanical Engineering,
Fakultetska 1, Zenica
Bosnia&Herzegovina

ABSTRACT

Sensor application in online robot programming is discussed in this paper. The possibility of reprogramming, besides physical features and control system, separates robots from traditional automatics and give them opportunity to accomplish versatility of tasks. By integrating sensors and adequate programming systems, the possibilities of using robots in different kind of tasks are significantly improved. In practical examples, done with MITSUBISHI RV-2AJ robot, the importance of sensors is shown, since without of using them, it would be impossible to achieve goals in the specific tasks.

1. INTRODUCTION

A large part of potential robot efficiency is lost because of high investment in special equipment and environment organization in order to eliminate all sources of disorder/confusion. Sensors allow robots to accomplish tasks in presence of significant environment disorder without special purposes tools installing. Sensors are used in different purposes in robot programming, and each of them has unique influence on system design. Basic sensor applications in robot programming are:

- Initializing and terminating of motion;
- Selection between alternative action;
- Information gathering about position and features of object;
- Complying to external constraints.

2. PRACTICAL EXAMPLES

Three practical examples done with MITSUBISHI RV-2AJ with the use of external sensors are presented hereinafter.

2.1. I example – Inspection of object features

DEF IO is command for input/output variables defining. Three input variables are defined in first example regarding on three existing sensors in robot work cell. Sensor names are PART_AV, COLOUR and B2 and their bit numbers 8, 9, 10 respectively. MOV command is command for robot's end effector motion in joint coordinate system. The initial position of program is P1. Robot set end effector in this position and wait for a signal from PART_AV sensor, so the executing of program can be continued. Command WAIT is delaying next command executing until variable or signal reach predefined value. When the work piece gets to its initial position, sensor PART_AV is activated, the signal is changed from 0 to 1, and the conditions for continuation of program execution are met. Then, from safety reasons, command DLY delays next command execution for 5 seconds. Robot's end effector is then positioned 40 mm above P2 position in Z-axis direction using MOV P2,-40 command. This positioning is necessary, because of safe manipulation and collision avoiding with work piece. The end effector is opened with command HOPEN 1, before work piece approaching. The speed of approaching is also limited with command OVRD.

The end effector is then moved in position for work piece taking up with command MVS – linear interpolation. The end effector stays in position P2 1 second (DLY 1), before closing (HCLOSE 1), and also after closing. Work piece is then placed in position P3 in the same way as in position P2. The end effector is positioned in position P4 and using sensor COLOUR installed on robot's end effector, colour of work piece is determined. This sensor has ability to distinguish black colour from other colours, and since in this exercise work pieces are black and red, they will be sorted. IF…THEN…ELSE as a branching command provides subprogram choose depending on signal state from sensor COLOUR. If the signal is 1 (sensor active), the subprogram *RED will be called, and subprogram *BLACK otherwise. The subprogram sequences are same in both subprograms, but since the height of black and red pieces is different, there is necessary to record different positions for further program execution, so there are two subprograms.

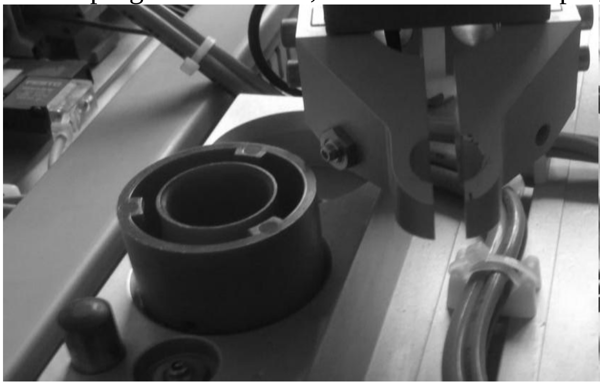

Figure 1. Colour detection

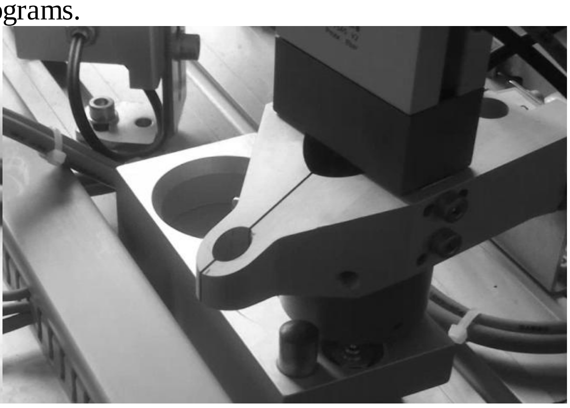

Figure 2. Hole detection

The end effector is then moved in position for work piece taking up with command MVS – linear interpolation. The end effector stays in position P2 1 second (DLY 1), before closing (HCLOSE 1), and also after closing. Work piece is then placed in position P3 in the same way as in position P2. The end effector is positioned in position P4 and using sensor COLOUR installed on robot's end effector, colour of work piece is determined. This sensor has ability to distinguish black colour from other

colours, and since in this exercise work pieces are black and red, they will be sorted. IF…THEN…ELSE as a branching command provides subprogram choose depending on signal state from sensor COLOUR. If the signal is 1 (sensor active), the subprogram *RED will be called, and subprogram *BLACK otherwise. The subprogram sequences are same in both subprograms, but since the height of black and red pieces is different, there is necessary to record different positions for further program execution, so there are two subprograms.

Work pieces on the bottom have three symmetrically distributed holes, so they can be placed in position P210 where there is vertical pin, and work piece is perched through the any of bottom holes on it. In order to do this, third sensor B2 is used for work piece bottom hole detecting. Starting position for hole detecting is P6. After work piece positioning in P6, it rotates about its own axis by 1°, until a hole is found, using WHILE M_IN(10) = 1 command. Command WHILE executes set of instructions until condition is true. In this case condition is signal from sensor B2 – bit number 10 equal 1. When hole is above sensor, the signal becomes 0, which means that condition from WHILE command is not fulfilled anymore, so the execution of set of instruction within WHILE…WEND loop is stopped, and execution of instructions after WEND starts. Position M2 represents constant geometric difference between desired position for work piece placing and position in which the hole on work piece is found. By adding this position with current robot's end effector position (command P_CURR) where hole is found, position P210 is obtained. In the case that in starting positioning of work piece, the hole is above sensor, sensor will not be activated. So robot immediately starts with positioning in P210, based on geometric constant and current position. Counting dimension specificity of inflected parts, such case could cause collision between work piece and pin. From this reason, the correction of start position is made. If the hole of work piece is above sensor in starting position, and sensor is not activated, the work piece will rotate around its own axis by 20°. The hole is found from this new starting position degree by degree, in order to find hole always with same orientation of end effector. In the end, work piece stacking is made in two columns, depending of colour – black or red.

2.2 II example – Work pieces sorting

The second example shows sensor use in work pieces sorting. The sensor with ability to distinct black from other colours is used in this example. The work pieces are black and red and they have got cylindrical shape. Let assume that four work pieces in a row are coming in work area of the robot. The robot task is to arrange them in two groups: black and red ones. In any work pieces 'delivery', it is possible any ratio of black and red pieces, i.e. the number of black and the number of red pieces are random variables.

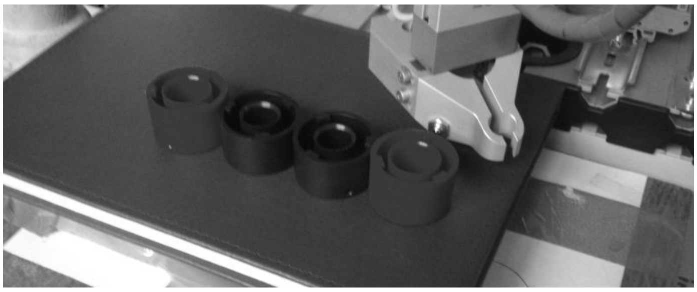

Figure 3. Work pieces sorting

After defining input/output variables, i.e. sensors in program, the robot is moved to starting position P1. The sensor installed on robot's end effector will scan all pieces and then it will start with sorting. Since it is known that four pieces are coming in every delivery, FOR loop is used, and inside this loop the position P2 is defined. This position P2 is added to current position of robot and robot moves 40 mm in X-axis direction, because the distance between two pieces is 40 mm and they are aligned parallel to X-axis. The sensor state in every position next to work piece is checked with IF command. If the sensor signal is 1, the subprogram *RED is called and the subprogram *BLACK vice versa. The arrays M(100) and M(200) are defined in both subprograms respectively and number 1 is written in them when the sensor is activated.

After scanning of all pieces, robot's grip is set to position to take work piece. FOR loop is also used in this case, therewith that variable M in FOR loop is changed from 4 to 1, as opposed first case, when was changed from 1 to 4, because robot will not back in beginning of row, but the first will be take the last inspected piece. According to value in M100(M) array (0 or 1), the work piece will be stored in one of two stacks, calling subprogram *STACK1 apropos *STACK2.

2.3. III example – Bottle opening

In third example, robot is used for bottle opening. In first try of realization of this task, it is found that bottles' height varies, what was a problem for task accomplishing using fixed positions of end effector during opening of bottles. This was a reason for another try to accomplish this task, but this time using a sensor. The idea was to define height of bottle with sensor on end effector, and using this point to define all movements of end effector with opener.

Sensors are defined in beginning of code. PART_AV sensor is used as a kind of initiator, since start of program execution waits for signal from this sensor using command WAIT. Position P1 is start – safe position in program execution, and from this position robot's end effector is moved in position P2, where accessory for bottle taking up is located. Considering robot arm reach, it is possible to take bottles only from one row in bottle crate. FOR loop is used for robot arm moving along row of bottles and at the same time the number of bottles which will be opened is determined. The distance between two bottles is constant and position P80.X is defined in respect of which robot arm will be moved for every bottle that should be picked up. Two bottles with obvious difference in height (approx. 40 mm) are used for bottle opening demonstration. After bottle picking up from crate, it is necessary to put it on place for bottle opening, but since there is difference in height, the correction of smaller bottle positioning is necessary. That is the reason why is bottle set above sensor B2 with number 10. The bigger bottle will activate sensor and otherwise, smaller one, will not. IF command and input information from sensor 2 (M_IN(10)) determine correction of bottle positioning on place for bottle opening, depending on is it bigger or smaller bottle. Beside correction depending on sensor B2 state, variable M15 is also defined and its function is to 'remember' if is on position for bottle opening smaller or bigger bottle placed. This information is needed for bottle taking up after it is opened. After bottle positioning in place for opening, the accessory for bottle manipulating is left on its place and end effector is positioned in P4,-80 position.

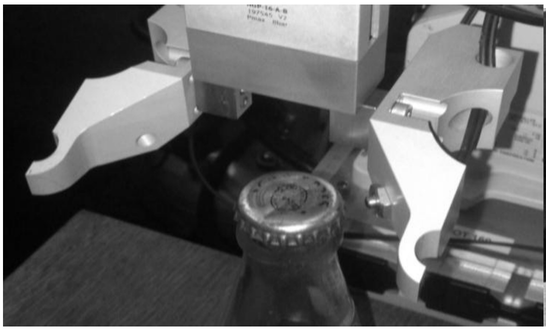

Figure 4. Bottle height detection

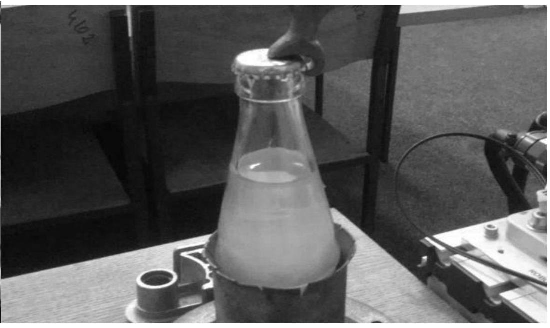

Figure 5. Bottle opening

From this position is lowered by half millimetre in Z-axis direction using WHILE command. The end effector will lower until sensor COLOUR placed on end effector activation. In the moment of sensor activation end effector is stopped and its current position is stored as P6 position. Sensor would be activated when it gets to the top of the bottle and so defined position will be used for defining of positions for bottle opening.

The end effector is now moved in position P3, where the opener is placed. After opener taking, it is placed in P20 position for bottle approaching. This position is calculated based on P6 position determined by sensor and predefined position P100, which represents connection between position depending of bottle height and position for bottle approaching. After opener positioning in this position, bottle is opened by relative movement of end effector passing through positions P61 to P67. The opener takes back in its place and accessory for bottle manipulation is used again. Depending of M15 variable value (0 or 1), it is determined is it big or small bottle in question. Then the end effector takes bottle and places it on P40 or P42 position.

3. CONCLUSION

Sensors incorporated with high sophisticated programming languages allow robots most complex tasks accomplishment (i.e. robots used in medical purposes) and adaptability in disordered and unknown environments (i.e. robots launched in space). In order to make possible that robots accomplish their tasks easily even in disordered and unknown environment, certain degree of information of disorder is required from sensor systems. In practical examples in this paper several improved functions and applications of robot are shown, even with very simple sensor system. It is shown sensor use for initializing of program execution, where it is waited for sensor signal for program starting. Sensor signal is also used for stopping of end effectors movement. The certain features of objects are detected using sensors, for example colour detection or finding specific features of objects (opening on bottom of work piece). Different objects (bottles different in height) are detected using sensor. Selection of alternative actions and movements is made based on sensor information. It can be concluded that without appropriate external sensors, robots are almost useless, and on the other hand with proper sensor choice and adequate program support, possibility of robots use is tremendously increased. Artificial intelligence which tends to be the highest level of robotic systems is inseparably related to complex sensor systems. All this lead to conclusion that in future with increasing use of robots, increasingly will be used complex, advanced sensor systems.

REFERENCES

[1] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics Modelling,Planning and Control, Springer-Verlag, London, 2009

[2] Lozano-Perez, T.: Robot Programming, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1982

[3] Binti Yusopa Y.: Study in industrial robot programming, Kolej Universiti Teknologi Tun HusseShhn, 2004

[4] A. Zabka, G. Kinder, COSIMIR Robotics FESTO training, Festo Didactic GmbH & Co.KG, 2007

[5] Mitsubishi Industrial Robot, RV-1A/RV-2AJ Series, Standard Specifications Manual