Robot Programming Lab #11 Position registers and Offsets. JD Jones and John Nelson

Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Point Value = \_\_\_\_\_\_\_\_\_/150 points

In this lab you will be learning about Position Registers and Offsets. This is a big one. It is a good idea to have John or JD introduce these to you as it is some of the more difficult material in the class. You will create 3 new programs in this lab. The names of the programs will be the first 5 of your name and 11A, 11B and 11C.

Let’s understand the Position Register.

We will back up a little bit. When you teach the robot a POINT what happens is it takes into account the user frame, tool frame and current values of the encoders of each axis. The position has a value for each axis. The robots you are using are 6 axis so we have 6 data elements. They are X, Y, Z, W, P and R. You need to see where these values are located. Go into your program and place the cursor on P[1]. The F key should say position (F5). When you press it the screen will change. **Fill in the values for your P[1].**

**X\_\_\_\_\_\_\_\_\_\_\_ W\_\_\_\_\_\_\_\_\_\_**

**Y\_\_\_\_\_\_\_\_\_\_\_ P\_\_\_\_\_\_\_\_\_\_\_**

**Z\_\_\_\_\_\_\_\_\_\_\_ R\_\_\_\_\_\_\_\_\_\_\_**

When you tell the robot to move to P[1] it moves the motors until the encoders are those values. How it moves to those locations is controlled by the J, L or C element of the move command.

Let’s get started on the Position Register (PR). Think of a PR as a combination of a Position and a Register. It is a Register with 6 elements or storage locations instead of 1 like a register. The PR[1] has the same 6 elements as a position, X, Y, Z, W, P and R. A good application for a PR is when you want to modify the location. A great example is a robot palletizer. You create a PR for the first location and just modify it from that point for each box to be placed on the pallet. PR’s are great for being able to manipulate a position.

**Setting A PR:** There are three ways to set up a PR:

1. **Direct Entry**: Go into the PR from the PR Screen (Data -> Registers -> Position Registers) go into the PR and enter the data manually. This is a handy way to do it when you want to create a “shift” type of register that adds to the Axis of another position register.

2. **Shift Record**: Go to the Position Registers screen. Arrow up or down until the desired PR is highlighted then Shift+Record records the current position as that register. This has the big advantage that the program is NOT tied to a user frame One nice thing about position registers is that you can shift user frames and the PR’s move with them.

**3. Setting it to a position**: In your program, you can set a position register as a currently programed position:

PR[1] = P[1]

Would set PR[1] to the same position as P[1]. Every time the program routes through this line it would reset PR[1]. Remember that this ties you to a user frame.

**Offsets:** Offset is a way to move a programmed position. There are 3 different ways to perform an offset.

NOTE: When you offset a PR it offsets with regard to the active USER FRAME, not the world frame. So if your user frame is screwed up any offsetting is not going to work right.

.

1) Method 1 by adding 2 PR’s.

2) Method 2 by modifying an element of the PR

3) Method 3 by adding an offset command at the end of a move command.

**Method 1 Adding 2 PR’s.**

We will use 2 PR’s to add one to the other. Here is a simple program where the robot just keeps moving 30 millimeters to the X+ direction. The instructor will program and demonstrate what is below.

Make P[1] about 1 inch above the table near the middle of the work area.

**Program name: XXXXX11A**

1: PR[1] = P[1]

2: LBL[1]

3: L PR[1] 250mm/sec Fine

4: Wait = .5 sec

5: PR[1] = PR[1] + PR[2]

6: Jmp Lbl [1]

7: J P[1] 100% fine

8: END

The data table looks like this. Of course, the actual values for P[1] and PR[1] will be different. PR[2] will be the 30 mm offset.

P[1]

X =100 W=100

Y =150 P =150

Z =200 R =200

After line 1: is complete then PR[1] will look like this.

PR[1]

X =100 W=100

Y =150 P =150

Z =200 R =200

PR[2:offset]

X =30 W=0

Y =0 P =0

Z =0 R =0

When the robot runs the program.

1: PR[1] = P[1] Makes Position Register [1] the same as Position 1

2: LBL[1] If I need to explain this go back to lab 1!

3: L PR[1] 250mm/sec Fine Tells the robot to move to Position Register [1]

4: Wait = .5 sec Waits ½ second so you can see it move to each pos in auto.

5: PR[1] = PR[1] + PR[2] Adds 30 mm offset to the X direction only.

6: Jmp Lbl [1] Sends it back up to Line 2 to do it all over again.

7: J P[1] 100% fine This is required line so the robot knows where P[1] is!

8: END

What would PR[2:offset] look like if you wanted it to move 35 mm in the Y+ direction and 0 in the X direction? **Fill in the values!**

**PR[2:offset]**

**X =\_\_\_\_\_\_\_ W=\_\_\_\_\_\_\_**

**Y =\_\_\_\_\_\_\_ P =\_\_\_\_\_\_\_**

**Z =\_\_\_\_\_\_\_ R =\_\_\_\_\_\_\_**

Put it into the program and see if it works. Feel free to play with it and see what happens.

**What would you do if you wanted the robot to move in a diagonal direction?**

**Method 2 Modifying an element.**

You will CREATE A NEW PROGRAM name it LAB11B with your name as the first 5 characters.

Let’s try the Position Register Offset by another method.

There is another command on how to do PR offsets. It is similar to the CNC world. You can have an instruction such as PR[i,j]=PR[i,j] + 30. This is how it breaks down.

PR = Position Register

**i** = the PR number

j = the element of the position register. This is talking about the X, Y, Z, W, P, and R.

1 = X

2 = Y

3 = Z

4 = W

5 = P

6 = R

The instruction PR[1,2]=PR[1,2] +30 means it will offset the Y direction of Position register 1 by 30 mm.

**TASK A:**

Using any of the methods above Create a program that:

-Draws a square (about 30mm width) on the table using a PR for each corner.

Use one of the offset methods (PR(i,j) or PR+PR) and a counter to make it copy the square 3 times across in the positive y direction.

Will the program do the same thing the next time you run it?

The program name will be **XXXXX11B. X being your name.**

Show the instructor your program.

**Method 3 adding an offset to a move command.**

The offset command simply takes whatever the position values are and adds the PR offset to the move command. This will work with both positions and position registers. Let’s break it down.

The program says this.

1: J P[1] 100% fine offset PR[5]

The robot will add PR [5] to P[1] values and then move to the result position. Let’s put some numbers in this.

P [1]

X 100 W 50

Y 150 P 75

Z 200 R 100

PR[5]

X 23 W 0

Y 100 P 0

Z 0 R 0

Therefore when the robot runs the command

1: J P[1] 100% fine offset PR[5]

The actual position it will move to is the result of adding P[1] and PR[5].

X 123 W 50

Y 250 P 75

Z 200 R 100

This is an awesome way to do the approach and pick up or placement of an item. Simply modify the Z axis and the rest of the position stays the same. This is an easy way to do your next lab of palletizing.

**Task B:**

**Now create a program using this method and show it to your instructor.**

The program MUST include:

1) The basic 5 position program with only 3 programmed positions. Use the offset method 3 to create the Approach Pos A and Approach Pos B positions.

2) Physically move an item.

3) Create the program from scratch.

4) Set the frames, open gripper, and set the speed at the start of the program.

5) Program name **XXXXX11C**.



Be ready to demonstrate the following to the instructor.

**A brief note on Interaction between PR’s and User Frames.**

A big advantage of PRs is that you can write one program with them and then run it in different user frames to perform the same operation in multiple locations. An example of this would be deburring several parts that are next to each other. You build a user frame around the first part and write the program using PRs (set them with shift record!) on the first part. Then for successive parts all you need to do is create a user frame around the other parts, switch to the new user frame and run the program for the first part.

Points for

A) Method 1

a. P [1] Data filled out. 5 pts

b. PR[2] Data filled out. 5 pts

c. Answering the diagonal question. 10 pts

B) Method 2

a. Creating a program. 10 pts

b. Modifying the elements. 10 pts

c. Being creative. 10 pts

d. Creating the loop 3 times. 10 pts

e. Printing out of the program. 10 pts

f. Written explanation of the program. 10 pts

C) Method 3

a. Beginning of the program info, setting frames etc. 10 pts.

b. Creating a program 10 pts

c. Program descriptions. 10 pts

d. 3 positions or less programmed. 10 pts. PR[x] offset does not count as a position.

e. Printing out of the program. 10 pts

f. Written explanation of the program. 20 pts

INSTRUCTOR’S INITIAL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_