RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Robotics**

Subject Code: **ME-604**

Semester: **6th**

DEPARTMENT OF MECHANICAL ENGINEERING

ROBOTICS (ME 604 A)

UNIT IV (ROBOT PROGRAMMING)

LECTURE NOTES

Variable Assembly Language (VAL) is a computer-based control system and language designed specifically for use with Unimation Inc. industrial robots.

The VAL robot language is permanently stored as a part of the VAL system. This includes the programming language used to direct the system for individual applications. The VAL language has an easy to understand syntax. It uses a clear, concise, and generally self-explanatory instruction set. All commands and communications with the robot consist of easy to understand word and number sequences. Control programs are written on the same computer that controls the robot. As a real-time system, VAL's continuous trajectory computation permits complex motions to be executed quickly, with efficient use of system memory and reduction in overall system complexity. The VAL system continuously generates robot control commands, and can simultaneously interact with a human operator, permitting on-line program generation and modification.

A convenient feature or VAL is the ability to use libraries or manipulation routines. Thus, complex operations may be easily and quickly programmed by combining predefined subtasks.

The VAL language consists of monitor commands and program instructions. The monitor commands are used to prepare the system for execution of user-written programs. Program instructions provide the repertoire necessary to create VAL programs for controlling robot actions.

Robot Programming Languages

A robot will require a programming language for describing the operations that are to be done. Recently, there are plenty of robot programming languages available. Among them, five robot languages are commonly and basically used. They are:

1. RAIL
2. AML
3. VAL
4. AL
5. RPL

1. RAIL:

RAIL will be a best language for controlling two major tasks such as the manipulation and vision system. It is a high – level robot language based on Pascal, and it will implement a Motorola–68000 central processor, teach pendant, and terminal. This language was designed by Automatix for arc welding and inspection purposes.

RAIL comes with three different kinds of systems, namely:

Hitachi Process Robot – Arc Welding

Cartesian Arm – Assembly functions

Vision system without arm

2. AML:

AML (A Manufacturing Language) is a high level language based on sub routine, which is mainly implemented to manage RS / 1 Assembly Robot, End Effectors Active Force Feedback, and Cartesian Arm with hydraulic motors. The RS / 1 assembly robot incorporates a mini – computer (IBM Series/1), 192 KB memory, matrix printer, disk drive, display terminals, and keyboard. AML was developed by IBM Corporation for robot programming. An important reason for creating this language is to offer simple subsets and as well as powerful base language.

3. VAL:

VAL (Variable Assembly Language) Robot Programming Language is adopted mainly for Unimation Robots. As this language is designed with simple syntax, it is capable of illustrating the robot functions very easily. It includes two major tasks such as:

Program instructions are used to provide VAL programs in order to manage the robot functions.

Monitor commands are used to execute the user written programs.

4. AL:

AL robot language was developed in Artificial Intelligence Lab at Stanford University. It is the second generation language based on simultaneous Pascal. The programs are written and executed on PDP – 10. If the program is developed with high level code, then it should be written in Stanford Artificial Intelligence Language (SAIL). The AL system includes a big mainframe computer, and it generally runs on PDP 11/45. The PDP 11/45 implements one terminal, 128 KB RAM memory, and floating point processor. This language has got the capability to control two Stanford Scheinman and two PUMA 600 arms simultaneously.

5. RPL:

RPL robot language makes the improvement, checking, and correction of control algorithms very easy. It can be done even by an unskilled programmer like line foreman, production engineers, etc. The RPL programs are translated to interpretable code with the help of a compiler in SRI Robot Programming System. The programs are typically written in BLISS – 11 and run in RT – 11. The DECPDP – 10 is cross compiled into the LSI – 11 or PDP – 11. This robot language was designed in SRI International

Program algorithm

A programming algorithm as a recipe that describes the exact steps needed for the computer to solve a problem or reach a goal. We've all seen food recipes - they list the ingredients needed and a set of steps for how to make the described meal. Well, an algorithm is just like that. In computer lingo, the word for a recipe is a procedure, and the ingredients are called inputs. Your computer looks at your procedure, follows it to the letter, and you get to see the results, which are called outputs. A programming algorithm describes how to do something, and your computer will do it exactly that way every time. Well, it will once you convert your algorithm into a language it understands!

However, it's important to note that a programming algorithm is not computer code. It's written in simple English (or whatever the programmer speaks). It doesn't beat around the bush--it has a start, a middle, and an end. In fact, you will probably label the first step 'start' and the last step 'end.' It includes only what you need to carry out the task. It does not include anything unclear, often called ambiguous in computer lingo, that someone reading it might wonder about.

It always leads to a solution and tries to be the most efficient solution we can think up. It's often a good idea to number the steps, but you don't have to. Instead of numbered steps, some folks use indentation and write in pseudocode, which is a semi-programming language used to describe the steps in an algorithm. But, we won't use that here since simplicity is the main thing. Other folks just use a diagram called a flowchart, which we will discuss soon.

Distinguish Between Manual Lead Through And Teach Pendant Robotic Languages.

The difference between powered lead through and manual lead through is the manner in which the manipulator is moved through the motion cycle during programming.

Powered lead through involves the use of a teach pendant with toggle switches and/ or contact buttons for controlling the movement of the manipulator joints. Using the toggle switches or buttons, the programmer power drives the robot arm to the desired positions, in sequence, and records the positions into memory. Manual lead through requires the operator to physically grasp the end of arm or the tool that is attached to the arm and move it through the motion sequence, recording the path into memory.

Offline Programming Method Used in Robotics

Offline programming, or simulation, is most often used in robotics research to ensure that advanced control algorithms operate correctly before testing them on a real robot. Programs are developed offline, meaning that the robot only has to be halted while the new program is downloaded and tested. It is currently used by many industrial robot technicians as it causes the least disruption to the production line.

Despite its popularity, only a handful of companies are actually using this technology as it is still in its infancy, but its use is increasing each year. Developers are looking for ways to improve the intuition of the software in order to reduce the time taken to develop the simulation and test it on the robot.

No matter what type of programming an operator chooses, they don't need to spend huge sums of money to introduce automation into their manufacturing line. Once the robot is up and running, the manufacturer can rely on reconditioned and obsolete parts to keep their robot operating at optimum efficiency.

You might not have completed *Robot Odyssey* back in the day, but if you understand the basic techniques of teach, lead through and offline programming, you're all set to level up to the automation era.

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in