4.2 PROGRAMMING LANGUAGES FOR ROBOTICS

This article is all about giving an introduction about some of the programming languages which are used to design Robots.

There are many programming languages which we use while building Robots, we have a few programming languages which we always prefer to use in designing. Actually, the programming languages which we use mainly depend on the hardware one is using in building robots.

Some of them are- URBI, C and BASIC. URBI is an open source language. In this article we will try to know more about these languages. Let's start with URBI.

URBI :

URBI stands for Universal Real-time Behavior Interface. It is a client/server based interpreted language in which Robot works as a client and controller as a server. It makes us to learn about the commands which we give to Robots and receive messages from them. The interpreter and wrapped server are called as "URBI Engine". The URBI Engine uses commands from Client and receives messages to it. This language allows user to work on basic Perception- action principle. The users just have to write some simple loops on the basis of this principle directly in URBI.

PYTHON :

There is another language which is used in designing Robots. Python is an object- oriented language which is used to access and control Robots. Python is an interpreted language; this language has an application in working with mobile robots, particularly those manufactured by different companies. With python it is possible to use a single program for controlling many different robots. However, Python is slower than C++ but it has some good sides as well as it proved very easy to interact with robots using this language, it is highly portable and can be run in windows and MAC OSX plus it can easily be extendable using C and C++ language. Python is a very reliable language for string manipulation and text processing.

ROBOTC :

Other Languages which we use are C, C++ and C # etc. or their implementation, like ROBOTC, ROBOTC is an implementation of C language. If we are designing a simple Robot, we do not need assembly code, but in complex designing we need well-defined codes. ROBOTC is another programming language which is C-based. It is actually a text-based programming language. The commands which we want to give to our robot, first written on the screen in the form of simple text, now as we know that Robot is a kind of machine and a machine only understands machine language. So these commands need to be converted in machine language so that robot can easily understand and do whatever it is instructed to do. Although commands are given in text form (called as codes) but this language is very specific about the commands which is provided as instruction. If we do even a minor change in given text it will not accept it as command. If the command which is provided to it is correct it colorizes that text, and we came to know that the given command in text form is correct (as we have shown in our example given below). Programming done in ROBOTC is very easy to do. Commands given are very straightforward. Like if we want our robot to switch on any hardware part, we just have to give code regarding to that action in text form.

ROBOTICS.NXT :

ROBOTICS.NXT has a support for a simple message-based control. It direct commands, nxt-upload is one of its programs which is used to upload any file. It works on Linux. After getting introduction on programming languages, it becomes necessary to know something about MRDS as well, MRDS is an environment which is designed especially for controlling robots.

Microsoft Robotics Developer Studio:

Microsoft Robotics Developer Studio is an environment given for simulation purpose of Robots. It is based on a .net library concurrent implementation. This environment has support so that we can add other services as well. It has features which not only include creating and debugging Robot Applications but also it becomes easy to interact with sensors directly. C# programming language is used as a primary language in it. It has 4 main components:

Â· Concurrency and coordination Runtime (CCR)

Â· Decentralized software services (DSS)

Â· Visual Programming Language (VPL)

Â· Visual simulation environment (VSE)

Concurrency and coordination Runtime is a synchronous programming library based on .net framework. Although it is a component of MRDS but it can be used with any application. DSS is also a .net runtime environment, In DSS services are exposed as resources which one can access through programs. DSS uses DSSP (Decentralizes software services protocol) and HTTP.

If we want to graphics and visual effects in our programming, we use VPL. Visual Programming language is a programming language which allows us to create programs by doing manipulations in programming languages graphically. We use boxes and arrows in this kind of programming while we want to show dataflow kind of things.

Visual programming language has huge application in animations. The last component which we are going to describe is Visual Simulation Environment. VSE provides simulates physical objects. Visual Simulation environment is an integrated environment for picture-based, object oriented and component-based applications of simulation.

Programming in robotics is a very vast topic that we can’t cover in a single article. This isjust an introduction for those who want to get an idea about using languages in building of robots.