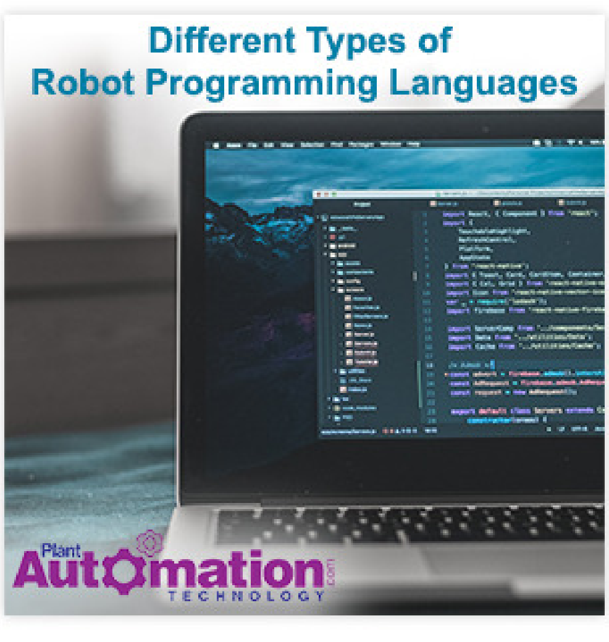
Different Types of Robot Programming Languages

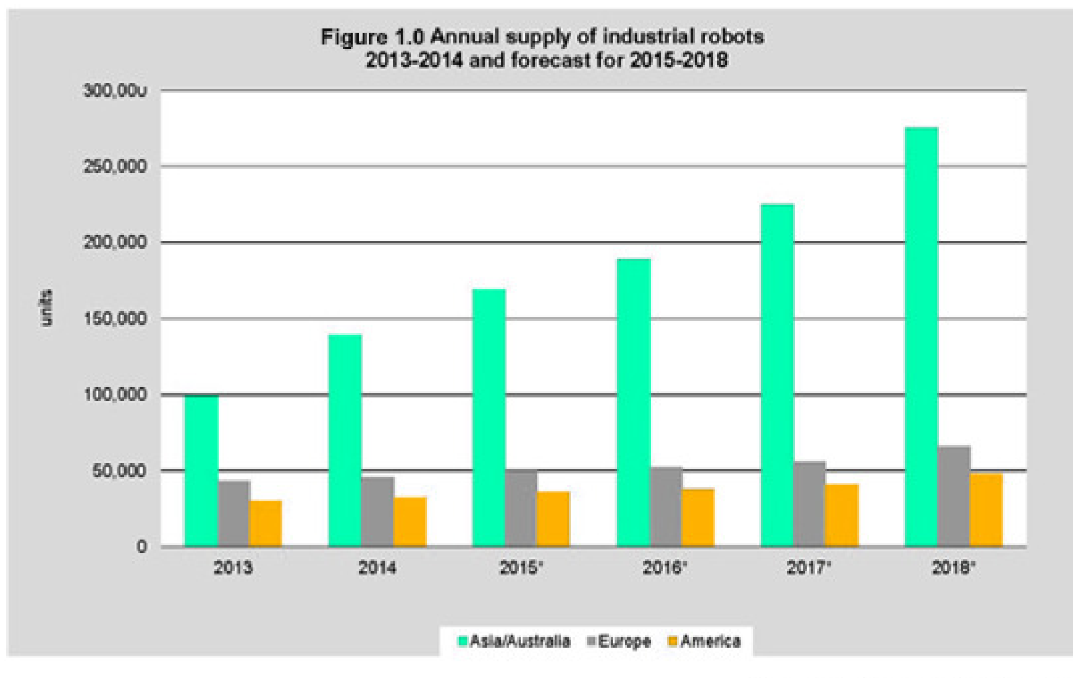


Robotics Market- Overview

With the help of robots, human efforts to perform complex works become simpler and precise. With this, many companies are showing interest and are investing on robots due to their benefits.

" According to ResearchAndMarkets, Industrial Robotics market is expected to grow by US$ 48,166.9 million by 2025 from US$ 16,847.6 million in 2017".

While robots are traditionally used in structured environments, with known and regulated inputs and outputs, industrial robots is gaining momentum in the recent years. With this, people are showcasing interest on programming and anyone with some experience in programming want to take advantage of the robot programming. It is predicted that the demand for robot programmers is expected to grow by a large amount over the rest of the decade.



Source: IFR World Robotics 2015

Figure 1: Industrial Robots Statistics

Robotic Programming:

To perform a particular action, robots are programmed either by guiding or by off-line programming. Most of the industrial robots are programmed by guiding a robot from point to point through the phases of an operation, with each point stored in the robotic control system.

Robots receive instructions through computer commands and this is referred to as manipulator level off-line programming. Usage of off-line programming involves higher-level languages, in which robotic actions are defined by tasks or objectives.

Robotic programmers must have knowledge on different types of programming languages as switching from computers to robots is not the smooth transition that many developers/programmers may think.

Different Types of Robot Programming Languages

Programming languages are having a insightful impact on robot integration in industrial machines. The sky is the limit for robotic industrial applications. Robotic software plays a key role in performing complex operations and accurate functionalities. To start programming in robotics, you must start learning various types of languages to make a robot work. Although robotics is a complex subject, learning these programming languages will help you design a project to create an easy-to-use interfaces.

There are over 1500 programming languages in the world, which are used to learn and implement. Here are the top and most popular programming languages in robotics:

1) C/C++

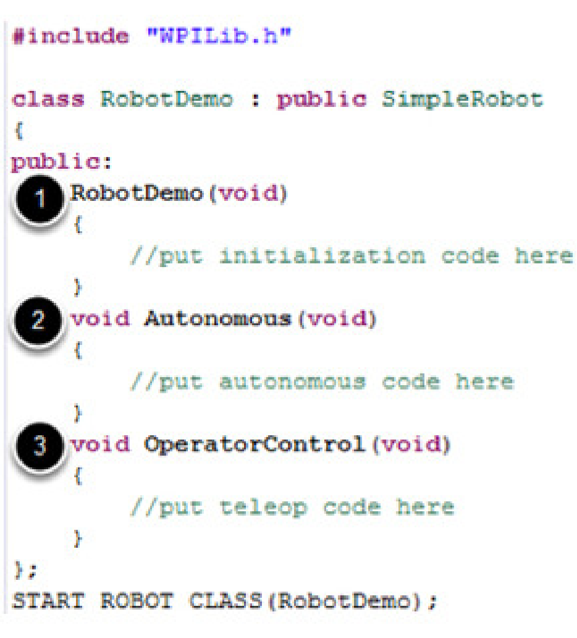
Choosing to learn C and C++ is the best start for an aspiring roboticists as it is a general purpose programming language that contains imperative, object-oriented and generic programming features.

Why C++ is the Number 1 Programming Language in Robotics?

C++ allows interaction with low level hardware, and also real time performance. C and C++ are very mature programming languages. To ensure the best performance of a robot, it will be better to use C++.

As the robotics is very dependent on the real time performance, C and C++ are the best options to the roboticists.

Example: The default Simple Robot template project



```
#include "WPILib.h"

class RobotDemo : public SimpleRobot
{
public:
    RobotDemo(void)
    {
        //put initialization code here
    }
    void Autonomous(void)
    {
        //put autonomous code here
    }
    void OperatorControl(void)
    {
        //put teleop code here
    }
};
START ROBOT CLASS(RobotDemo);
```

Source: 2014 FRC Control System

Now, let us explore the code contained inside each section. There are three main parts:

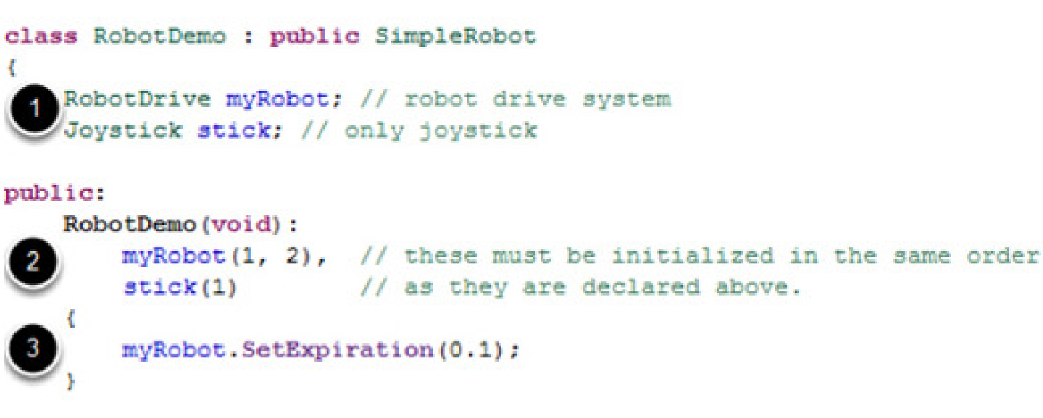
Constructor: This is where initialization code runs to create a class. In this case, it will run when the program first starts, before the robot is enabled. Constructor helps to initialize sensors, and create other WPILib objects.

While initializing, we cannot use the constructors to position actuators as motors will not operate.

Autonomous method: This method is used to run code during the autonomous period. The Autonomous method will run every time when the robot is put into Autonomous mode. A programmer must be very careful while running this method as it will not be stopped at the end of the autonomous period.

OperatorControl method: This method will be called when the robot enters the tele-operation part of the match. The code includes a loop that reads operator interface values (joysticks and switches) and operates actuators until the teleop period has ended.

Defining the variables for sample robot:



```
class RobotDemo : public SimpleRobot
{
    RobotDrive myRobot; // robot drive system
    Joystick stick; // only joystick

public:
    RobotDemo(void):
        myRobot(1, 2),  // these must be initialized in the same order
        stick(1)        // as they are declared above.
    {
        myRobot.SetExpiration(0.1);
    }
```

Source: 2014 FRC Control System

The sample robot consist of a joystick on USB port 1 for arcade drive and two motors on PWM ports 1 and 2. Here we create objects of type RobotDrive (myRobot) and Joystick (stick). This section of the code does three things:

- Defines the variables as members of our RobotDemo class
- Initializes the variables as part of the constructor using an Initialization List
- Performs robot initialization (in this case sets the safety timer expiration for the myRobot object to .1 seconds, see the next step for an explanation of motor safety timers).

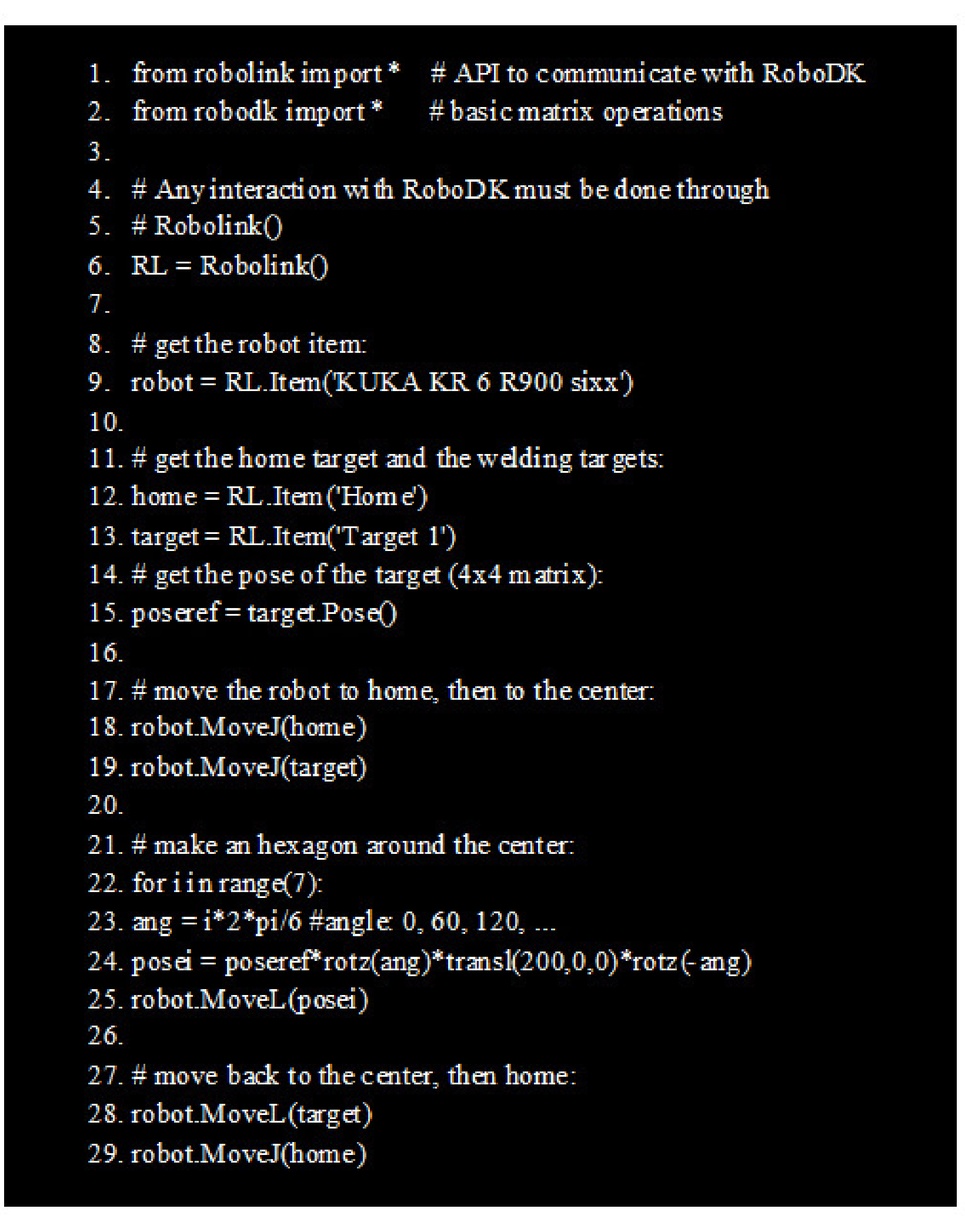
2) Python

Python is the high-level programming language and it plays a key role in building and testing robots. Python is a good platform to automate, teach, and post-process robot programs. Many people choose this programming language as we can write script that calculates, records, and simulates an entire robot program instead of manually teaching every statement to a robot. This helps to quickly test and visualize solution in simulation as well as refine the program and its logic.

Why Python is the Most Used Programming Language in Robotics?

- When compared to C, C++ and Java, you can write fewer lines of code in Python.
- Learning python is an essential guide to create an autonomous mobile robot using popular robotic software frameworks.
- This language is used in designing embedded system and it is an integral part of Robot operating system.
- The major focus of Python is ease of use and less time is required to program such as defining, and casting variable types.
- When you are required to implement some basic functionality, there are a huge number of free libraries for it, which means you do not have to 'reinvent the wheel'.

Example Program:



```
from robolink import *    # API to communicate with RoboDK
from robodk import *      # basic matrix operations

# Any interaction with RoboDK must be done through
# Robolink()
RL = Robolink()

# get the robot item:
robot = RL.Item('KUKA KR 6 R900 sixx')

# get the home target and the welding targets:
home = RL.Item('Home')
target = RL.Item('Target 1')
# get the pose of the target (4x4 matrix):
poseref = target.Pose()

# move the robot to home, then to the center:
robot.MoveJ(home)
robot.MoveJ(target)

# make an hexagon around the center:
for i in range(7):
ang = i*2*pi/6 #angle: 0, 60, 120, ...
posei = poseref*rotz(ang)*transl(200,0,0)*rotz(-ang)
robot.MoveL(posei)

# move back to the center, then home:
robot.MoveL(target)
robot.MoveJ(home)
```

This example showcases placing all the objects are carried out automatically through Python. We can easily place any object programmatically thanks to Python API. Using the Python API we can create, modify or delete any objects. In addition, we can also program the robot moves in the same Python script, this will allow us to automatically generate the robot program.

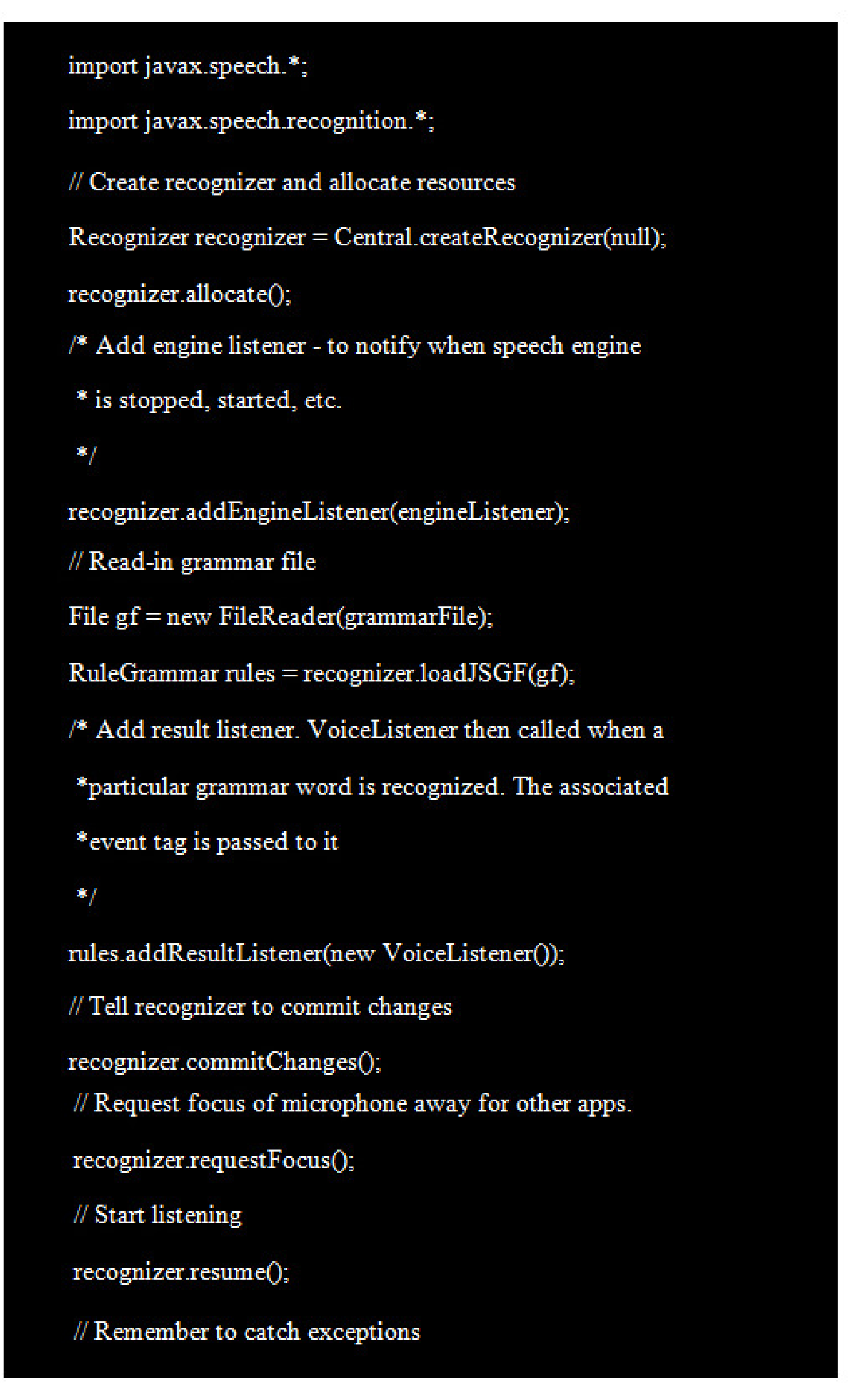
3) Java

To perform human-like tasks, robotics systems must be included with functions via programming languages. To enable robots with these functions, Java language can be used as it offers an array of APIs that are tailor-made to the needs of the robotics realm. Command-and-control recognizers, dictation systems, and speech synthesizers can be built using the Java Speech API and the Java Media Framework can be used to receive and process visual images.

Why Java is the Most Used Programming Language in Robotics?

- Java contains all the high-level features needed in robotics industry especially when it comes to artificial intelligence.
- Using Java, you can create highly efficient algorithms for machine learning, search and neural algorithms, language processing etc.
- The Java Virtual Machine interprets the instructions at run-time
- Java enables users to use the same code on many different machines, thanks to the Java Virtual Machine.
- It provides all the high level features needed to deal with various aspects of robotics.

Example Code to create a robotics voice listener using a grammar format file:



```
import javax.speech.*;

import javax.speech.recognition.*;

// Create recognizer and allocate resources

Recognizer recognizer = Central.createRecognizer(null);

recognizer.allocate();

/* Add engine listener - to notify when speech engine

 * is stopped, started, etc.

 */

recognizer.addEngineListener(engineListener);

// Read-in grammar file

File gf = new FileReader(grammarFile);

RuleGrammar rules = recognizer.loadJSGF(gf);

/* Add result listener. VoiceListener then called when a

 *particular grammar word is recognized. The associated

 *event tag is passed to it

 */

rules.addResultListener(new VoiceListener());

// Tell recognizer to commit changes

recognizer.commitChanges();
 // Request focus of microphone away for other apps.

 recognizer.requestFocus();

 // Start listening

 recognizer.resume();

 // Remember to catch exceptions
```

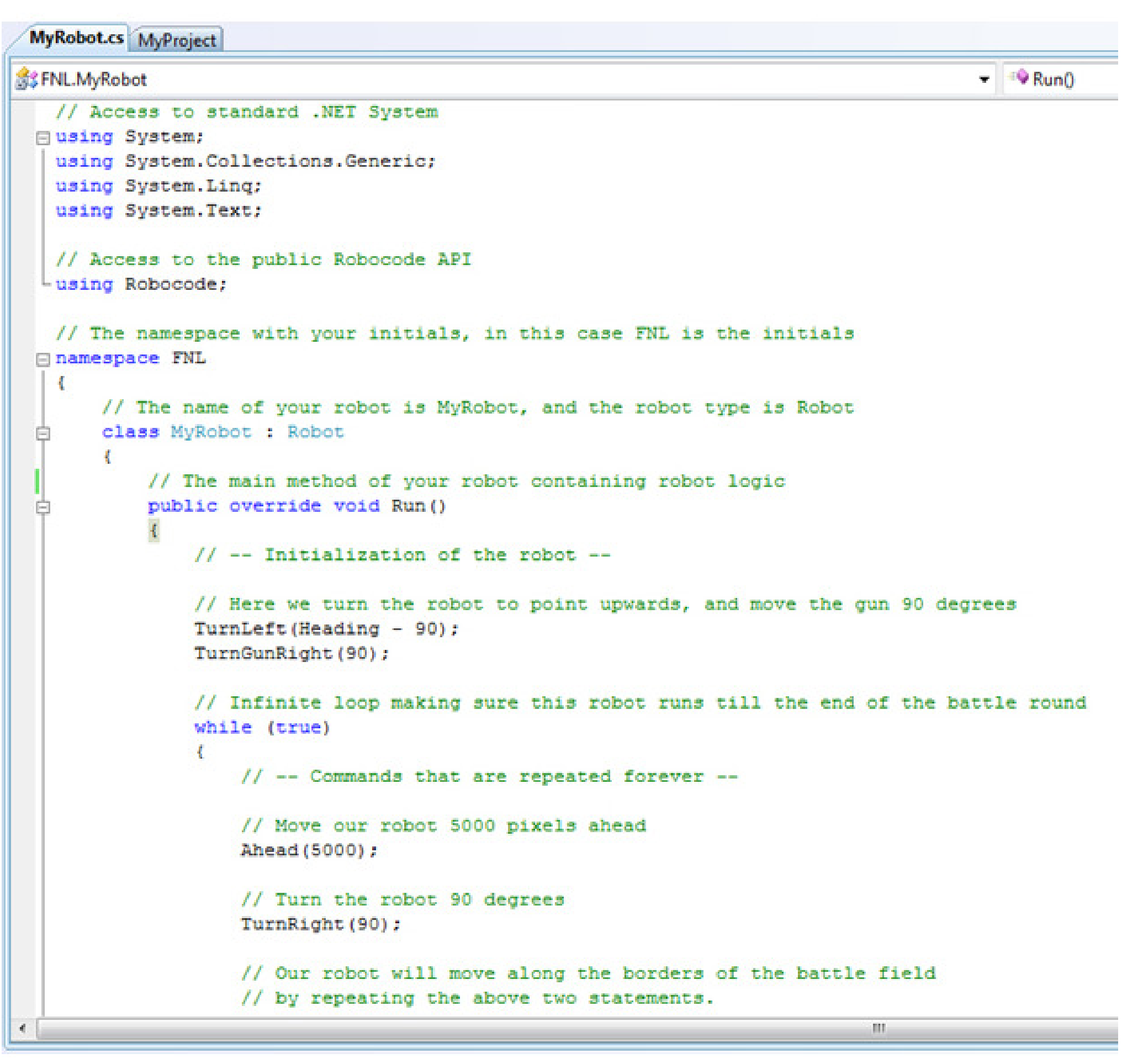
4) C#/.NET

C#/.NET is the Microsoft's proprietary language used to develop applications in Visual Studio. Examples include Netduino, FEZ Rhino and others). It provides aspiring programmers a solid foundation based on which they can branch out to different fields. C#/.NET is generally used in port and socket level programming.

Why C#/.Net is the Most Used Programming Language in Robotics?

- C#/.NET allows using multiple languages and has horizontal scalability.
- .NET creates a unified environment that allows robotic developers to create programs in C++, Java or Virtual Basic
- All tools and IDEs have been pre-tested and are easily available in the Microsoft Developer Network.
- Language integration is seamless, as you can call methods from C# to VB.NET

Example:



```
// Access to standard .NET System
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

// Access to the public Robocode API
using Robocode;

// The namespace with your initials, in this case FNL is the initials
namespace FNL
{
    // The name of your robot is MyRobot, and the robot type is Robot
    class MyRobot : Robot
    {
        // The main method of your robot containing robot logic
        public override void Run()
        {
            // -- Initialization of the robot --

            // Here we turn the robot to point upwards, and move the gun 90 degrees
            TurnLeft(Heading - 90);
            TurnGunRight(90);

            // Infinite loop making sure this robot runs till the end of the battle round
            while (true)
            {
                // -- Commands that are repeated forever --

                // Move our robot 5000 pixels ahead
                Ahead(5000);

                // Turn the robot 90 degrees
                TurnRight(90);

                // Our robot will move along the borders of the battle field
                // by repeating the above two statements.
```

Source: RoboWiki

5) MATLAB

Technically speaking, MATLAB is not a programming language but it is a tool with which you can find engineering solutions based on mathematics. Robotic developers need to learn MATLAB if they want to analyze data, produce advanced graphs or implement control systems.

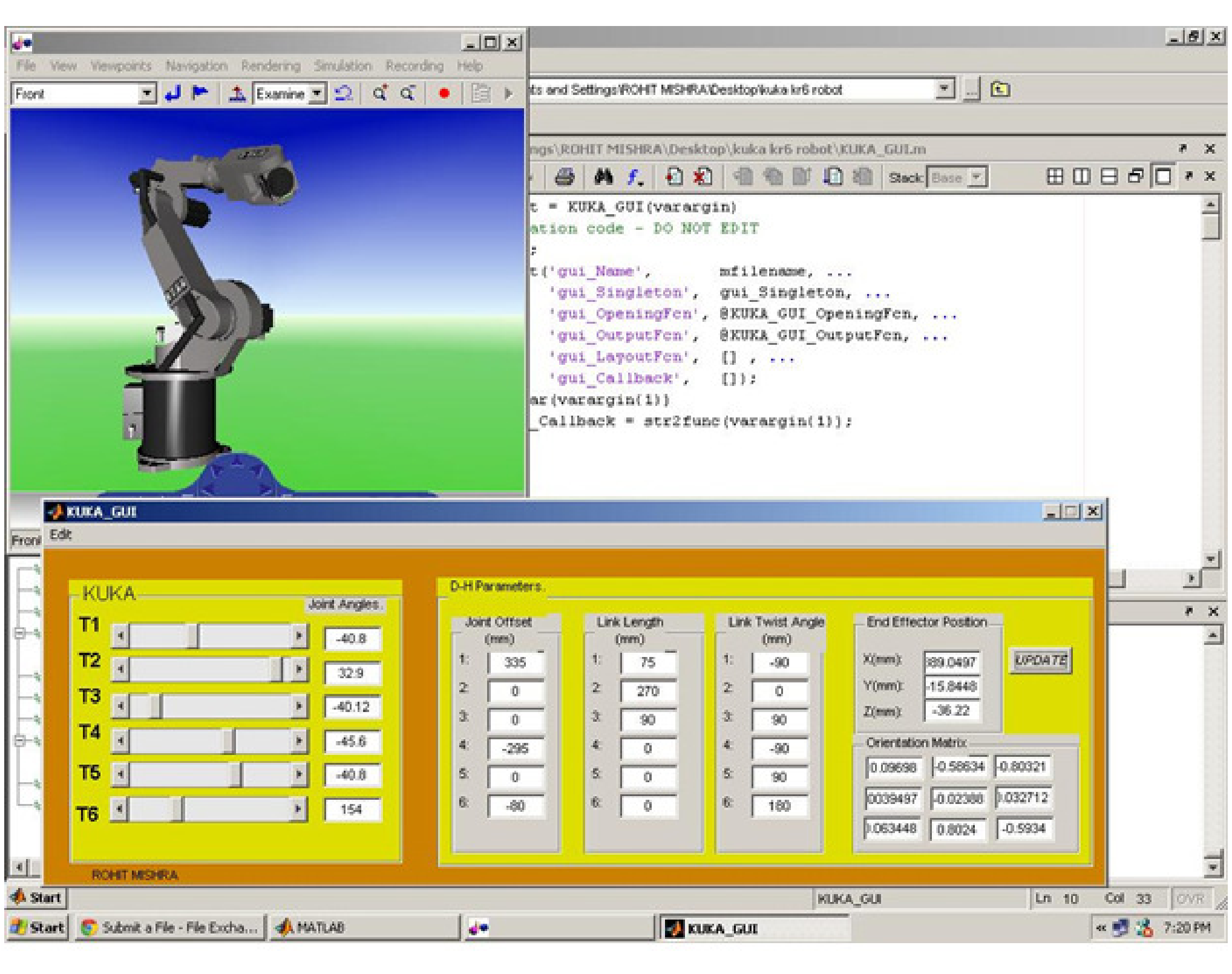
MATLAB, and its open source relatives, such as Octave, is very popular with some robotic engineers for analyzing data and developing control systems.

Programming for a robot requires designing the controller that governs robot behavior. Modeling and simulation became vital to understand how the controller interacts with the robot's environment perception, mobility, and interaction.

Why MATLAB is the Most Used Programming Language in Robotics?

- MATLAB is highly useful in designing the entire robotic system.
- It is widely used in the robotics industry as it is deeply rooted in the foundation and development of robots.
- It is a simulation tool whereby you can provide your algorithm or design and it simulates the result.
- On the other hand, simulation helps engineers to refine the system design and eliminate errors before developing hardware prototypes.

Example: Modeling and Simulation of Kuka kr6 robot.



```
t = KUKA_GUI(varargin)
ation code - DO NOT EDIT
;
t('gui_Name',       mfilename, ...
  'gui_Singleton',  gui_Singleton, ...
  'gui_OpeningFcn', @KUKA_GUI_OpeningFcn, ...
  'gui_OutputFcn',  @KUKA_GUI_OutputFcn, ...
  'gui_LayoutFcn',  [] , ...
  'gui_Callback',   []);
ar(varargin(1))
_Callback = str2func(varargin(1));
```

Joint	Joint Angles	Joint Offset (mm)	Link Length (mm)	Link Twist Angle (mm)
T1 / 1	-40.8	335	75	-90
T2 / 2	32.9	0	270	0
T3 / 3	-40.12	0	90	90
T4 / 4	-45.6	-295	0	-90
T5 / 5	-40.8	0	0	90
T6 / 6	154	-80	0	180

X(mm): 389.0497
Y(mm): -15.8448
Z(mm): -36.22

0.09698	-0.58634	-0.80321
0039497	-0.02388	0.032712
0.063448	0.8024	-0.5934

Which Is The Best One Among Them?

Learning these programming languages will help you to develop a robot or securing a job in the robotics industry. But, learning C/C++ on priority basis will help you as they are the best programming languages for robotics.

Many people feel that C and C++ are fast becoming outdated in other industries, but the usage and popularity for these languages' is increasing. Several companies choose C and C++ to develop robots as they contain more tools and library functions. These two languages are developer-friendly and developers can change line by line.

However, learning these programming languages will help you to develop your coding skills and they can help you to create an amazing Ultimate Intelligent bot.