Volume 7, Issue.7,

Year Jan. 2015

ELECTRONIKA

Yearly News letter

(Only for Internal Circulation)

HOD's Message

- Education, particularly Technical Education is playing a vital role in the development of the country by creating skilled manpower, enhancing industrial productivity and improving the quality of life. People talks about wonders of the world and include structures and monuments but Communication Engineering is the greatest wonders of the world that even no one imagined. To continuously spread the quality technical education we the Dept of Electronics and Communication Engineering are committed to take up responsibility of the holistic growth of the pupil coming in to its folds, is committed to make all possible efforts to help in realizing their dreams as well as the society to which he or she hails from.

Prof. Rupesh Dubey

HOD EC Dept.

Humanoid robot

TOPIO, a humanoid robot, played ping pong at Tokyo International Robot Exhibition (IREX) 2009.

A **humanoid robot** is a robot with its body shape built to resemble that of the human body. A humanoid design might be for functional purposes, such as interacting with human tools and environments, for experimental purposes, such as the study of bipedal locomotion, or for other purposes. In general, humanoid robots have a torso, a head, two arms, and two legs, though some forms of humanoid robots may model only part of the body, for example, from the waist up. Some humanoid robots may also have heads designed to replicate human facial features such as eyes and mouths. Androids are humanoid robots built to aesthetically resemble humans.

Enon was created to be a personal assistant. It is self-guiding and has limited speech recognition and synthesis. It can also carry things.

Humanoid robots are used as a research tool in several scientific areas.

Researchers need to understand the human body structure and behavior (biomechanics) to build and study humanoid robots. On the other side, the attempt to the simulation of the human body leads to a better understanding of it.

Human cognition is a field of study which is focused on how humans learn from sensory information in order to acquire perceptual and motor skills. This knowledge is used to develop computational models of human behavior and it has been improving over time.

It has been suggested that very advanced robotics will facilitate the enhancement of ordinary humans. See transhumanism.

Although the initial aim of humanoid research was to build better orthosis and prosthesis for human beings, knowledge has been transferred between both disciplines. A few examples are: powered leg prosthesis for neuromuscularly impaired, ankle-foot orthosis, biological realistic leg prosthesis and forearm prosthesis.

Besides the research, humanoid robots are being developed to perform human tasks like personal assistance, where they should be able to assist the sick and elderly, and dirty or dangerous jobs. Regular jobs like being a receptionist or a worker of an automotive manufacturing line are also suitable for humanoids. In essence, since they can use tools and operate equipment and vehicles designed for the human form, humanoids could theoretically perform any task a human being can, so long as they have the proper software. However, the complexity of doing so is deceptively great.

They are becoming increasingly popular for providing entertainment too. For example, Ursula, a female robot, sings, play music, dances, and speaks to her audiences at Universal Studios. Several Disney attractions employ the use of animatrons, robots that look, move, and speak much like human beings, in some of their theme park shows. These animatrons look so realistic that it can be hard to decipher from a distance whether or not they are actually human. Although they have a realistic look, they have no cognition or physical autonomy. Various humanoid robots and their possible applications in daily life are featured in an independent documentary film called *Plug & Pray*, which was released in 2010.

Humanoid robots, especially with artificial intelligence algorithms, could be useful for future dangerous and/or distant space exploration missions, without having the need to turn back around again and return to Earth once the mission is completed.

Sensors

A sensor is a device that measures some attribute of the world. Being one of the three primitives of robotics (besides planning and control), sensing plays an important role in robotic paradigms.

Sensors can be classified according to the physical process with which they work or according to the type of measurement information that they give as output. In this case, the second approach was used.

Proprioceptive sensors

Proprioceptive sensors sense the position, the orientation and the speed of the humanoid's body and joints.

In human beings the otoliths and semi-circular canals (in the inner ear) are used to maintain balance and orientation. In addition humans use their own proprioceptive sensors (e.g. touch, muscle extension, limb position) to help with their orientation._ Humanoid robots use accelerometers to measure the acceleration, from which velocity can be calculated by integration; tilt sensors to measure inclination; force sensors placed in robot's hands and feet to measure contact force with environment; position sensors, that indicate the actual position of the robot (from which the velocity can be calculated by derivation) or even speed sensors.

Exteroceptive sensors

An artificial hand holding a lightbulb

Arrays of tactels can be used to provide data on what has been touched. The Shadow Hand uses an array of 34 tactels arranged beneath its polyurethane skin on each finger tip.[3] Tactile sensors also provide information about forces and torques transferred between the robot and other objects.

Vision refers to processing data from any modality which uses the electromagnetic spectrum to produce an image. In humanoid robots it is used to recognize objects and determine their properties. Vision sensors work most similarly to the eyes of human beings. Most humanoid robots use CCD cameras as vision sensors.

Sound sensors allow humanoid robots to hear speech and environmental sounds, and perform as the ears of the human being. Microphones are usually used for this task.

Actuators

Actuators are the motors responsible for motion in the robot.

Humanoid robots are constructed in such a way that they mimic the human body, so they use actuators that perform like muscles and joints, though with a different structure. To achieve the same effect as human motion, humanoid robots use mainly rotary actuators. They can be either electric, pneumatic, hydraulic, piezoelectric or ultrasonic.

Hydraulic and electric actuators have a very rigid behavior and can only be made to act in a compliant manner through the use of relatively complex feedback control strategies. While electric coreless motor actuators are better suited for high speed and low load applications, hydraulic ones operate well at low speed and high load applications.

Piezoelectric actuators generate a small movement with a high force capability when voltage is applied. They can be used for ultra-precise positioning and for generating and handling high forces or pressures in static or dynamic situations.

Ultrasonic actuators are designed to produce movements in a micrometer order at ultrasonic frequencies (over 20 kHz). They are useful for controlling vibration, positioning applications and quick switching.

Pneumatic actuators operate on the basis of gas compressibility. As they are inflated, they expand along the axis, and as they deflate, they contract. If one end is fixed, the other will move in a linear trajectory. These actuators are intended for low speed and low/medium load applications. Between pneumatic actuators there are: cylinders, bellows, pneumatic engines, pneumatic stepper motors and pneumatic artificial muscles.

Planning and control

In planning and control, the essential difference between humanoids and other kinds of robots (like industrial ones) is that the movement of the robot has to be human-like, using legged locomotion, especially biped gait. The ideal planning for humanoid movements during normal walking should result in minimum energy consumption, like it does in the human body. For this reason, studies on dynamics and control of these kinds of structures become more and more important.The question of walking biped robots stabilization on the surface is of great importance. Maintenance of the robot's gravity center over the center of bearing area for providing a stable position can be chosen as a goal of control.[4]

To maintain dynamic balance during the walk, a robot needs information about contact force and its current and desired motion. The solution to this problem relies on a major concept, the Zero Moment Point (ZMP).

Another characteristic of humanoid robots is that they move, gather information (using sensors) on the "real world" and interact with it. They don't stay still like factory manipulators and other robots that work in highly structured environments. To allow humanoids to move in complex environments, planning and control must focus on self-collision detection, path planning and obstacle avoidance.Humanoids don't yet have some features of the human body. They include structures with variable flexibility, which provide safety (to the robot itself and to the people), and redundancy of movements, i.e. more degrees of freedom and therefore wide task availability. Although these characteristics are desirable to humanoid robots, they will bring more complexity and new problems to planning and control.

Timeline of developments

Year	**Development**
c. 250 BC	The *Lie Zi* described an automaton.[5]
c. 50 AD	Greek mathematician Hero of Alexandria described a machine to automatically pour wine for party guests.[6]
1206	Al-Jazari described a band made up of humanoid automata which, according to Charles B. Fowler, performed "more than fifty facial and body actions during each musical selection."[7] Al-Jazari also created hand washing automata with automatic humanoid servants,[8][*verification needed*] and an elephant clock incorporating an automatic humanoid mahout striking a cymbal on the half-hour.[*citation needed*] His programmable "castle clock" also

	featured five musician automata which automatically played music when moved by levers operated by a hidden camshaft attached to a water wheel.[9]
1495	Leonardo da Vinci designs a humanoid automaton that looks like an armored knight, known as Leonardo's robot.[10]
1738	Jacques de Vaucanson builds The Flute Player, a life-size figure of a shepherd that could play twelve songs on the flute and The Tambourine Player that played a flute and a drum or tambourine.[11]
1774	Pierre Jacquet-Droz and his son Henri-Louis created the Draughtsman, the Musicienne and the Writer, a figure of a boy that could write messages up to 40 characters long.[12]
1898	Nikola Tesla publicly demonstrates his "automaton" technology by wirelessly controlling a model boat at the Electrical Exposition held at Madison Square Garden in New York City during the height of the Spanish–American War.
1921	Czech writer Karel Čapek introduced the word "robot" in his play *R.U.R. (Rossum's Universal Robots)*. The word "robot" comes from the word "robota", meaning, in Czech and Polish, "labour, drudgery".[10]
1927	The Maschinenmensch ("machine-human"), a gynoid humanoid robot, also called "Parody", "Futura", "Robotrix", or the "Maria impersonator" (played by German actress Brigitte Helm), perhaps the most memorable humanoid robot ever to appear on film, is depicted in Fritz Lang's film Metropolis.
1941-42	Isaac Asimov formulates the Three Laws of Robotics, and in the process of doing so, coins the word "robotics".
1948	Norbert Wiener formulates the principles of cybernetics, the basis of practical robotics.
1961	The first digitally operated and programmable non-humanoid robot, the Unimate, is installed on a General Motors assembly line to lift hot pieces of metal from a die casting

	machine and stack them. It was created by George Devol and constructed by Unimation, the first robot manufacturing company.
1969	D.E. Whitney publishes his article "Resolved motion rate control of manipulators and human prosthesis".[13]
1970	Miomir Vukobratović has proposed Zero Moment Point, a theoretical model to explain biped locomotion.[14]
1972	Miomir Vukobratović and his associates at Mihajlo Pupin Institute build the first active anthropomorphic exoskeleton.
1973	In Waseda University, in Tokyo, Wabot-1 is built. It was able to walk, to communicate with a person in Japanese and to measure distances and directions to the objects using external receptors, artificial ears and eyes, and an artificial mouth.[15]
1980	Marc Raibert established the MIT Leg Lab, which is dedicated to studying legged locomotion and building dynamic legged robots.[16]
1983	Using MB Associates arms, "Greenman" was developed by Space and Naval Warfare Systems Center, San Diego. It had an exoskeletal master controller with kinematic equivalency and spatial correspondence of the torso, arms, and head. Its vision system consisted of two 525-line video cameras each having a 35-degree field of view and video camera eyepiece monitors mounted in an aviator's helmet.[17]
1984	At Waseda University, the Wabot-2 is created, a musician humanoid robot able to communicate with a person, read a normal musical score with his eyes and play tunes of average difficulty on an electronic organ.[15]
1985	Developed by Hitachi Ltd, WHL-11 is a biped robot capable of static walking on a flat surface at 13 seconds per step and it can also turn.[15]

1985	WASUBOT is another musician robot from Waseda University. It performed a concerto with the NHK Symphony Orchestra at the opening ceremony of the International Science and Technology Exposition.
1986	Honda developed seven biped robots which were designated E0 (Experimental Model 0) through E6. E0 was in 1986, E1 – E3 were done between 1987 and 1991, and E4 - E6 were done between 1991 and 1993.[18]
1989	Manny was a full-scale anthropomorphic robot with 42 degrees of freedom developed at Battelle's Pacific Northwest Laboratories in Richland, Washington, for the US Army's Dugway Proving Ground in Utah. It could not walk on its own but it could crawl, and had an artificial respiratory system to simulate breathing and sweating.[15]
1990	Tad McGeer showed that a biped mechanical structure with knees could walk passively down a sloping surface.[19]
1993	Honda developed P1 (Prototype Model 1) through P3, an evolution from E series, with upper limbs. Developed until 1997.[18]
1995	Hadaly was developed in Waseda University to study human-robot communication and has three subsystems: a head-eye subsystem, a voice control system for listening and speaking in Japanese, and a motion-control subsystem to use the arms to point toward campus destinations.
1995	Wabian is a human-size biped walking robot from Waseda University.
1996	Saika, a light-weight, human-size and low-cost humanoid robot, was developed at Tokyo University. Saika has a two-DOF neck, dual five-DOF upper arms, a torso and a head. Several types of hands and forearms are under development also. Developed until 1998.[15]
1997	Hadaly-2, developed at Waseda University, is a humanoid robot which realizes interactive communication with humans. It communicates not only informationally, but also

	physically.
2000	Honda creates its 11th bipedal humanoid robot, able to run, ASIMO.[18]
2001	Sony unveils small humanoid entertainment robots, dubbed Sony Dream Robot (SDR). Renamed Qrio in 2003.
2001	Fujitsu realized its first commercial humanoid robot named HOAP-1. Its successors HOAP-2 and HOAP-3 were announced in 2003 and 2005, respectively. HOAP is designed for a broad range of applications for R&D of robot technologies.[20]
2002	HRP-2, biped walking robot built by the Manufacturing Science and Technology Center (MSTC) in Tokyo.[21]
2003	JOHNNIE, an autonomous biped walking robot built at the Technical University of Munich. The main objective was to realize an anthropomorphic walking machine with a human-like, dynamically stable gait.[22]
2003	Actroid, a robot with realistic silicone "skin" developed by Osaka University in conjunction with Kokoro Company Ltd.[23]
2004	Persia, Iran's first humanoid robot, was developed using realistic simulation by researchers of Isfahan University of Technology in conjunction with ISTT.[24]
2004	KHR-1, a programmable bipedal humanoid robot introduced in June 2004 by a Japanese company Kondo Kagaku.
2005	The PKD Android, a conversational humanoid robot made in the likeness of science fiction novelist Philip K Dick, was developed as a collaboration between Hanson Robotics, the FedEx Institute of Technology, and the University of Memphis.[25]
2005	Wakamaru, a Japanese domestic robot made by Mitsubishi Heavy Industries, primarily

	intended to provide companionship to elderly and disabled people.[26]
2005	Nao is a small open source programmable humanoid robot developed by Aldebaran Robotics, in France. Widely used by world wide universities as a research platform and educational tool.[27]
2005	The Geminoid series is a series of ultra-realistic humanoid robots or Actroid developed by Hiroshi Ishiguro of ATR and Kokoro in Tokyo. The original one, Geminoid HI-1 was made at its image. Followed Geminoid-F in 2010 and Geminoid-DK in 2011.[27]
2006	RoboTurk is designed and realized by Dr Davut Akdas and Dr Sabri Bicakci at Balikesir University. This Research Project Sponsored By The Scientific And Technological Research Council Of Turkey (TUBITAK) in 2006. RoboTurk is successor of biped robots named "Salford Lady" and "Gonzalez" at university of Salford in the UK. It is the first humanoid robot supported by Turkish Government.[28]
2006	REEM-A, a biped humanoid robot designed to play chess with the Hydra Chess engine. The first robot developed by PAL Robotics, it was also used as a walking, manipulation speech and vision development platform.[29]
2006	iCub, a biped humanoid open source robot for cognition research.[30]
2006	Mahru, a network-based biped humanoid robot developed in South Korea.[31]
2007	TOPIO, a ping pong playing robot developed by TOSY Robotics JSC.[32]
2007	Twendy-One, a robot developed by the WASEDA University Sugano Laboratory for home assistance services. It is not biped, as it uses an omni-directional mobile mechanism.[33]
2008	Justin, a humanoid robot developed by the German Aerospace Center (DLR).[34]
2008	KT-X, the first international humanoid robot developed as a collaboration between the

	five-time consecutive RoboCup champions, Team Osaka, and KumoTek Robotics.[35]
2008	Nexi, the first mobile, dexterous and social robot, makes its public debut as one of *TIME* magazine's top inventions of the year.[36] The robot was built through a collaboration between the MIT Media Lab Personal Robots Group,[37] UMass Amherst and Meka robotics.[38][39]
2008	Salvius, The first open source humanoid robot built in the United States is created.
2008	REEM-B, the second biped humanoid robot developed by PAL Robotics. It has the ability to autonomously learn its environment using various sensors and carry 20% of its own weight.[40]
2009	HRP-4C, a Japanese domestic robot made by National Institute of Advanced Industrial Science and Technology, shows human characteristics in addition to bipedal walking.
2009	Turkey's first dynamically walking humanoid robot, SURALP, is developed by Sabanci University in conjunction with Tubitak.[41]
2009	Kobian, a robot developed by WASEDA University can walk, talk and mimic emotions.[42]
2009	DARwIn-OP, an open source robot developed by ROBOTIS in collaboration with Virginia Tech, Purdue University, and University of Pennsylvania. This project was supported and sponsored by NSF.[*citation needed*]
2010	NASA and General Motors revealed Robonaut 2, a very advanced humanoid robot. It was part of the payload of Shuttle Discovery on the successful launch February 24, 2011. It is intended to do spacewalks for NASA.[43]
2010	Students at the University of Tehran, Iran unveil the Surena II. It was unveiled by President Mahmoud Ahmadinejad.[44]

2010	Researchers at Japan's National Institute of Advanced Industrial Science and Technology demonstrate their humanoid robot HRP-4C singing and dancing along with human dancers.[45]
2010	In September the National Institute of Advanced Industrial Science and Technology also demonstrates the humanoid robot HRP-4. The HRP-4 resembles the HRP-4C in some regards but is called "athletic" and is not a gynoid.
2010	REEM, a humanoid service robot with a wheeled mobile base. Developed by PAL Robotics, it can perform autonomous navigation in various surroundings and has voice and face recognition capabilities.[46]
2011	Robot Auriga was developed by Ali Özgün HIRLAK and Burak Özdemir in 2011 at University of Cukurova. Auriga is the first brain controlled robot, designed in Turkey. Auriga can service food and medicine to paralysed people by patient's thoughts. EEG technology is adapted for manipulation of the robot. The project was supported by Turkish Government.[47]
2011	In November Honda unveiled its second generation Honda Asimo Robot. The all new Asimo is the first version of the robot with semi-autonomous capabilities.
2012	In April, the Advanced Robotics Department in Italian Institute of Technology released its first version of the *CO*mpliant hu*MAN*oid robot COMAN which is designed for robust dynamic walking and balancing in rough terrain.[48]
2013	In December 20–21, 2013 DARPA Robotics Challenge ranked the top 16 humanoid robots competing for the US$2 million cash prize. The leading team, SCHAFT, with 27 out of a possible score of 30 was bought by Google.[49] PAL Robotics launches REEM-C the first humanoid biped robot developed as a robotics research platform 100% ROS based.
2014	Manav -India's first 3D printed humanoid robot developed in the laboratory of A-SET Training and Research Institutes by Diwakar Vaish (head Robotics and Research, A-SET

	Training and Research Institutes) [50]

Student Achievements:

The students of EC department are enthusiastic, energetic and active. They participated in sports, cultural events organized by IES and also by other institutes.

The students who secured positions in various activities organized under annual festival "Swaranjali 2015"are

Mahendi Competition "Swaranjali 2015"

1. Anupriya Jain got 2 position (EC) I yr.

Rangoli Competition "Swaranjali 2015"

1. Deepa Bansal got 1 position (III yr)
2. Gunjan Rathore got 1 position (IIIyr)

Ananad Mela Competition "Swaranjali 2015"

1. Nidhi Piya got 2 position
2. AkshaySingh got 2 Position in games

AD-MAD Show Competition "Swaranjali 2015"

1. Deepa Bansal got 1 position
2. Ishan Vashaney got 1 position
3. Gunjan Rathore got 1 position
4. Ayush Gupta got 1 position
5. Abhishek Karoliya got 1 position
6. Prakhar Saxena got 2 Position
7. Aman Shukla got 2 Position
8. Swaraj Pillai got 2 Position
9. Karan Budana got 2 Position

Video Graph Competition "Swaranjali 2015"

1. Sameer Singh got 2 Position
2. Natash got 2 Position
3. Juhi Pare got 2 Position

Volume 8, Issue.8,

Year June 2015

ELECTRONIKA

Yearly News letter

(Only for Internal Circulation)

HOD's Message 

- Education, particularly Technical Education is playing a vital role in the development of the country by creating skilled manpower, enhancing industrial productivity and improving the quality of life. People talks about wonders of the world and include structures and monuments but Communication Engineering is the greatest wonders of the world that even no one imagined. To continuously spread the quality technical education we the Dept of Electronics and Communication Engineering are committed to take up responsibility of the holistic growth of the pupil coming in to its folds, is committed to make all possible efforts to help in realizing their dreams as well as the society to which he or she hails from.

Prof. Rupesh Dubey

HOD EC Dept.

LabVIEW

Developer(s)	National Instruments
Stable release	2015 / August 2015
Operating system	Cross-platform: Windows, Mac OS X, Linux
Type	Data Acquisition, Instrument Control, Test Automation, Analysis and Signal Processing, Industrial Control, Embedded Design
License	Proprietary
Website	ni.com/labview

The graphical language is named "G" (not to be confused with G-code). Originally released for the Apple Macintosh in 1986, LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various versions of UNIX, Linux, and Mac OS X. The latest version of LabVIEW is LabVIEW 2015, released in August 2015.

Dataflow programming

The programming language used in LabVIEW, also referred to as G, is a dataflow programming language. Execution is determined by the structure of a graphical block diagram (the LabVIEW-source code) on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data become available. Since this might be the case for multiple nodes simultaneously, G is inherently capable of parallel execution. Multi-processing and multi-threading hardware is automatically exploited by the built-in scheduler, which multiplexes multiple OS threads over the nodes ready for execution.

Graphical programming

LabVIEW ties the creation of user interfaces (called front panels) into the development cycle. LabVIEW programs/subroutines are called virtual instruments (VIs). Each VI has three components: a block diagram, a front panel and a connector panel. The last is used to represent the VI in the block diagrams of other, calling VIs. The front panel is built using controls and indicators. Controls are inputs – they allow a user to supply information to the VI. Indicators are outputs – they indicate, or display, the results based on the inputs given to the VI. The back panel, which is a block diagram, contains the graphical source code. All of the objects placed on the front panel will appear on the back panel as terminals. The back panel also contains structures and functions which perform operations on controls and supply data to indicators. The structures and functions are found on the Functions palette and can be placed on the back panel. Collectively controls, indicators, structures and functions will be referred to as nodes. Nodes are connected to one another using wires – e.g. two controls and an indicator can be wired to the addition function so that the indicator displays the sum of the two controls. Thus a virtual instrument can either be run as a program, with the front panel serving as a user interface, or, when dropped as a node onto the block diagram, the front panel defines the inputs and outputs for the node through the connector pane. This implies each VI can be easily tested before being embedded as a subroutine into a larger program.

The graphical approach also allows non-programmers to build programs by dragging and dropping virtual representations of lab equipment with which they are already familiar. The LabVIEW programming environment, with the included examples and documentation, makes it simple to create small applications. This is a benefit on one side, but there is also a certain danger of underestimating the expertise needed for high-quality G programming. For complex algorithms or large-scale code, it is important that the programmer possess an extensive knowledge of the special LabVIEW syntax and the topology of its memory management. The most advanced LabVIEW development systems offer the possibility of building stand-alone applications. Furthermore, it is possible to create distributed applications, which communicate by a client/server scheme, and are therefore easier to implement due to the inherently parallel nature of G.

Interfacing to Devices

LabVIEW includes extensive support for interfacing to devices, instruments, cameras, and other devices. Users interface to hardware by either writing direct bus commands (USB, GPIB, Serial) or using high-level, device-specific, drivers that provide kooi native LabVIEW function nodes for controlling the device.

LabVIEW includes built-in support for NI hardware platforms such as CompactDAQ and CompactRIO, with a large number of device-specific blocks for such hardware, the *Measurement and Automation eXplorer* (MAX) and *Virtual Instrument Software Architecture* (VISA) toolsets.

National Instruments makes thousands of device drivers available for download on the NI Instrument Driver Network (IDNet).[1]

Code compilation

In terms of performance, LabVIEW includes a compiler that produces native code for the CPU platform. The graphical code is translated into executable machine code by interpreting the syntax and by compilation. The LabVIEW syntax is strictly enforced during the editing process and compiled into the executable machine code when requested to run or upon saving. In the latter case, the executable and the source code are merged into a single file. The executable runs with the help of the LabVIEW run-time engine, which contains some precompiled code to perform common tasks that are defined by the G language. The run-time engine reduces compilation time and also provides a consistent interface to various operating systems, graphic systems, hardware components, etc. The run-time environment makes the code portable across platforms. Generally, LabVIEW code can be slower than equivalent compiled C code, although the differences often lie more with program optimization than inherent execution speed.[*citation needed*]

Large libraries

Many libraries with a large number of functions for data acquisition, signal generation, mathematics, statistics, signal conditioning, analysis, etc., along with numerous graphical interface elements are provided in several LabVIEW package options. The number of advanced mathematic blocks for functions such as integration, filters, and other specialized capabilities usually associated with data capture from hardware sensors is enormous. In addition, LabVIEW includes a text-based programming component called MathScript with additional functionality for signal processing, analysis and mathematics. MathScript can be integrated with graphical programming using "script nodes" and uses a syntax that is generally compatible with MATLAB.[*citation needed*]

Parallel programming

LabVIEW is an inherently concurrent language, so it is very easy to program multiple tasks that are performed in parallel by means of multithreading. This is, for instance, easily done by drawing two or more parallel while loops. This is a great benefit for test system automation, where it is common practice to run processes like test sequencing, data recording, and hardware interfacing in parallel.

Ecosystem

Due to the longevity and popularity of the LabVIEW language, and the ability for users to extend the functionality, a large ecosystem of third party add-ons has developed through contributions from the community. This ecosystem is available on the LabVIEW Tools Network, which is a marketplace for both free and paid LabVIEW add-ons.

User community

There is a low-cost LabVIEW Student Edition aimed at educational institutions for learning purposes. There is also an active community of LabVIEW users who communicate through several e-mail groups and Internet forums.

Criticism

LabVIEW is a proprietary product of National Instruments. Unlike common programming languages such as C or FORTRAN, LabVIEW is not managed or specified by a third party standards committee such as ANSI, IEEE, ISO, etc.

Dataflow programming model

Due to its thorough adoption of a data-flow programming model as opposed to the sequential ordering of arbitrary commands like most other (usually text-based) languages there is a very real barrier to many people who attempt to apply already-learned principles from other programming approaches to LabVIEW. The inherent parallel nature of the execution of LabVIEW code is a perennial source of confusion among those who are accustomed to other approaches. Due to this, most opinions tend to be highly polarised with people either being extremely fond of it or being extremely hostile to it.[*citation needed*]

Licensing

Building a stand-alone application with LabVIEW requires the Application Builder component which is included with the Professional Development System but requires a separate purchase if using the Base Package or Full Development System.[2]

Run-time environment

Compiled executables produced by version 6.0 and later of the Application Builder are not truly standalone in that they also require the LabVIEW run-time engine be installed on any target computer which runs the application.[3] The use of standard controls requires a run-time library for any language. All major operating systems supply the required libraries for common languages such as C. However, the run-time required for LabVIEW is not supplied with any operating system and has to be specifically installed by the administrator or user. This can cause problems if an application is distributed to a user who may be prepared to run the application but does not have the inclination or permission to install additional files on the host system prior to running the executable.

Race conditions and pseudo parallel execution

The G gives the impression of being a parallel language (such as VHDL) that has modules that run in parallel, however, it is essentially implemented on a non parallel platform without explicit race condition control. This greatly simplifies programming on multi-core and multi-processor systems, as long as the user takes care of the race conditions.[*citation needed*]

Performance

LabVIEW makes it difficult to get machine or hardware limited performance and tends to produce applications that are slower than hand coded native languages such as C. High performance can be achieved while using multi-core machines or dedicated toolkits for specific operations. LabVIEW makes multi-core programming much simpler and faster than text based languages.[*citation needed*]

Light weight applications

Very small applications still have to start the runtime environment which is a large and slow task. This makes writing and running small applications or applications that might run in parallel on the same platform problematic and tends to restrict LabVIEW to monolithic applications. Examples of this might be tiny programs to grab a single value from some hardware that can be used in a scripting language - the overheads of the runtime environment render this approach impractical with LabVIEW.Timing system

LabVIEW uses the January 1, 1904 epoch as its "zero" time. Other programs that use the January 1, 1904 epoch are Apple Inc.'s Mac OS through version 9, Palm OS, and Microsoft Excel (optionally).[4]

Release history

Starting with LabVIEW 8.0, major versions are released around the first week of August, to coincide with the annual National Instruments conference NI Week, and followed by a bug-fix release the following February. In 2009 National Instruments started to name the releases after the year in which they are released. The bug-fix is called a Service Pack (for instance the 2009 service pack 1 is released in February 2010).

Name/Version	**Build Number**	**Date**
LabVIEW project begins		April 1983
LabVIEW 1.0 (for Macintosh)	??	October 1986
LabVIEW 2.0	??	January 1990
LabVIEW 2.5 (first release for Sun & Windows)	??	August 1992
LabVIEW 3.0 (Multiplatform)	??	July 1993
LabVIEW 3.0.1 (first release for Windows NT)	??	1994
LabVIEW 3.1	??	1994
LabVIEW 3.1.1 (first release with "application builder" capability)	??	1995
LabVIEW 4.0	??	April 1996
LabVIEW 4.1	??	1997

LabVIEW 5.0	??	February 1998
LabVIEW RT (Real Time)	??	May 1999
LabVIEW 6.0 (6i)	6.0.0.4005	26 July 2000
LabVIEW 6.1	6.1.0.4004	12 April 2001
LabVIEW 7.0 (Express)	7.0.0.4000	April 2003
LabVIEW PDA module first released	??	May 2003
LabVIEW FPGA module first released	??	June 2003
LabVIEW 7.1	7.1.0.4000	2004
LabVIEW Embedded module first released	??	May 2005
LabVIEW 8.0	8.0.0.4005	September 2005
LabVIEW 8.20 (native Object Oriented Programming)	??	August 2006
LabVIEW 8.2.1	8.2.1.4002	21 February 2007
LabVIEW 8.5	8.5.0.4002	2007
LabVIEW 8.6	8.6.0.4001	24 July 2008
LabVIEW 8.6.1	8.6.0.4001	10 December 2008
LabVIEW 2009 (32 and 64-bit)	9.0.0.4022	4 August 2009
LabVIEW 2009 SP1	9.0.1.4011	8 January 2010
LabVIEW 2010	10.0.0.4032	4 August 2010
LabVIEW 2010 f2	10.0.0.4033	16 September 2010

LabVIEW 2010 SP1	10.0.1.4004	17 May 2011
LabVIEW for LEGO MINDSTORMS (2010 SP1 with some modules)		August 2011
LabVIEW 2011	11.0.0.4029	22 June 2011
LabVIEW 2011 SP1	11.0.1.4015	1 March 2012
LabVIEW 2012	12.0.0.4029	August 2012
LabVIEW 2012 SP1	12.0.1.4013	December 2012
LabVIEW 2013	13.0.0.4047	August 2013
LabVIEW 2013 SP1	13.0.1.4017	March 2014[5]
LabVIEW 2014		August 2014
LabVIEW 2014 SP1	14.0.1.4008	March 2015

Repositories and libraries

OpenG, as well as LAVA Code Repository (LAVAcr), serve as repositories for a wide range of Open Source LabVIEW applications and libraries. SourceForge has LabVIEW listed as one of the possible languages in which code can be written.

VI Package Manager has become the standard package manager for LabVIEW libraries. It is very similar in purpose to Ruby's RubyGems and Perl's CPAN, although it provides a graphical user interface similar to the Synaptic Package Manager. VI Package Manager provides access to a repository of the OpenG (and other) libraries for LabVIEW.

Tools exist to convert MathML into G code.[6]

Related software

National Instruments also offers a product called Measurement Studio, which offers many of the test, measurement and control capabilities of LabVIEW, as a set of classes for use with Microsoft Visual Studio. This allows developers to harness some of LabVIEW's strengths within the text-based .NET framework. National Instruments also offers LabWindows/CVI as an alternative for ANSI C programmers.

When applications require sequencing, users often use LabVIEW with TestStand test management software, also from National Instruments.

The Ch interpreter is a C/C++ interpreter that can be embedded into LabVIEW for scripting.[7]

The TRIL Centre Ireland BioMobius platform and DSP Robotics' FlowStone DSP also use a form of graphical programming similar to LabVIEW, but are limited to the biomedical and robotics industries respectively.

LabVIEW has a direct node with modeFRONTIER, a multidisciplinary and multi-objective optimization and design environment, written to allow coupling to almost any computer-aided engineering tool. Both can be part of the same process workflow description and can be virtually driven by the optimization technologies available in modeFRONTIER.

Department Workshop: Department of Electronics and Communication IES, IPS Academy organized Workshop on LabVIEW on 13 Oct. 2015