International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 05, May-2015

Gesture Controlled User Interface Using Inertial Measurement Unit

Suhas. S
Department of Studies in Electronics,
University of Mysore, P G Centre,
Hemagangothri, Hassan, India

Dileep. M. K
Lecturer, Department of Studies in Electronics,
University of Mysore, P G Centre,
Hemagangothri, Hassan, India

***Abstract*— User interface is a layer of interaction between Man and Machines, the requirement for these User Interface/ Human Machine Interfaces [HMI] were born since the invention of machines itself, since then it is continuously evolving. The ultimate goal of this evolution is to make the interface easy, enjoyable and efficient. What's better than using natural human gestures like posture of your hand or orientation of your wrist? The goal is to create a gestural interface through which you can control different peripherals or to make software initiations.**

Keywords*— Human Machine Interface [HMI], Gesture, Microelectromechanical Systems [MEMS],Degrees of Freedom [DOF], Inertial Measurement Unit [IMU]

I. INTRODUCTION

Human Machine Interface [HMI] is a combination of hardware and software elements, which provides the way for user to accomplish a task. It is deployed for control and visualization interface between human and a process, machine, application or an appliance. The goal of this interaction is to allow effective operation and control from the human end, while the machine continuously feeds back information that aids operator's decision making process. Generally the aim of user interface design is to produce a user interface which is easy, enjoyable and efficient, i.e. it must be realistic and natural. The above criteria can be fulfilled by creating a gestural interface. A gesture is a form of non verbal communication in which visible bodily actions or postures communicates particular message. In this gesture recognition system different gestures are decoded using sensors, which is then analyzed and processed by a processing unit then these gestures are mapped for different actions of the peripherals and software initiations.

The proposed Gestural Interface consists of 6 DOF Inertial Measurement Unit [IMU] consisting of 3 axis MEMS Accelerometer and 3 axis MEMS Gyroscope used to sense different gestures it is connected to the microcontroller which collects, analyzes and processes the data from the sensor, then these data is fed to the computer where we can see visual interpretation of the given data. The suitable form of data can be wirelessly transmitted to the receiver using RF modules; the receiver can manipulate the received data to identify different gestures which can then be mapped to different actions of the peripherals.

II. OBJECTIVE

Human Machine Interaction in today's world is mainly through haptic communication and peripheral interaction which suffers from giving an immersive and natural interface and is quite limited, hence the objective of the proposed system is to create a Kinetic User Interface which provides more immersive and natural way of interaction.

III. RELATED WORKS

The gesture recognition system has got its pace due to the recent development in technologies. Motion sensing system for gesture recognition usually consists of a camera which extracts different attributes and acts according to different gestures, another type of gesture recognition system comprises of a electric field sensor, any hand movements on top of it is captured and gesture is recognized, but both kinds of gesture recognition system requires the line of sight with the sensor in order to recognize the gesture. Recently there has been a breakthrough to overcome this problem by the type of gesture recognition system called Device sensing in which sensors like flex sensor, accelerometer, or gyroscope is mounted on the body and gesture is recognized.

IV. DESIGN METHODOLOGY

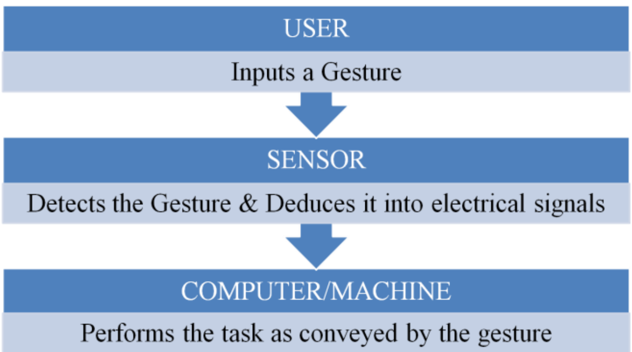

Figure 1 - Block Diagram of GCUI system

The basic design concept involves an Inertial Measurement Unit comprising of accelerometer and gyroscope fitted on the hand, based on the different hand postures both sensors gives different readings which is combined using Motion Fusion algorithm and fed into the microcontroller where the fused data is manipulated according to different applications and thus control can be established using different gestures.

V. Hardware Components

Different hardware components used in the development of the system are:

1. MPU6050 Inertial Measurement Unit [IMU]

The MPU-6050 is the world's I integrated 6-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP), with its dedicated I^2C sensor bus. By providing an integrated MotionFusion output, the DMP in the MPU-6050 offloads the intensive MotionProcessing computational requirements from the system processor, minimizing the need for frequent polling of the motion sensor output.

The MPU-6050 features:

1.1. Three-Axis MEMS Gyroscope with 16-bit ADCs and Signal Conditioning

The MPU-6050 consists of three independent vibratory MEMS rate gyroscopes, which detect rotation about the X-, Y-, and Z- Axes. When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a capacitive pickoff. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. This voltage is digitized using individual on-chip 16-bit Analog-to-Digital Converters (ADCs) to sample each axis.

1.2. Three-Axis MEMS Accelerometer with 16-bit ADCs and Signal Conditioning

The MPU-6050's 3-Axis accelerometer uses separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. The MPU-6050's architecture reduces the accelerometers' susceptibility to fabrication variations as well as to thermal drift. When the device is placed on a flat surface, it will measure 0*g* on the X- and Y-axes and +1*g* on the Z-axis. The accelerometers' scale factor is calibrated at the factory and is nominally independent of supply voltage. Each sensor has a dedicated sigma-delta ADC for providing digital outputs.

1.3. Digital Motion Processor

The embedded Digital Motion Processor (DMP) is located within the MPU-6050. The DMP acquires data from accelerometers, gyroscopes, and additional 3rd party sensors such as magnetometers, and processes the data. The resulting data can be read from the DMP's registers, or can be buffered in a FIFO. The DMP has access to one of the MPU's external pins, which can be used for generating interrupts. An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data.

2. ATmega328P Microcontroller

The ATMEL ATmega328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328P achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed. The AVR core combines a rich instruction set with 32 general purpose working registers. All the32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

Some of the important features are:

- High Performance 8-Bit Microcontroller
- Advanced RISC Architectured, AVR CPU
- 23 Programmable I/O Lines
- 32K Bytes of In-System Self-Programmable Flash program memory
- 2K Bytes Internal SRAM
- 1K Bytes EEPROM
- 32 general purpose working registers
- Three flexible Timer/Counters with CCP modules
- 6-channel 10-bit ADC
- Serially programmable USART
- Byte-oriented 2-wire Serial Interface
- SPI serial port
- Programmable Watchdog Timer with internal Oscillator
- Five software selectable power saving modes.

3.FT232 USB to Serial Converter

The FT232R is a USB to serial UART interface device which simplifies USB to serial designs and reduces external component count by fully integrating an external EEPROM, USB termination resistors and an integrated clock circuit which requires no external crystal, into the device. It has been designed to operate efficiently with a USB host controller by using as little as possible of the total USB bandwidth available.

Some of the important features are:

- Single chip USB to asynchronous serial data transfer interface.
- No USB specific firmware programming required.
- Fully integrated 1024 bit EEPROM storing device descriptors
- Data transfer rates from 300 baud to 3 Mbaud at TTL levels.
- FIFO receives and transmits buffers for high data throughput.

VI. SOFTWARES

1. Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs. It includes a code editor and code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. It also accepts plug-ins that enhances the functionality at almost every level. It also supports different programming languages and allows the code editor and debugger to support nearly any programming language.

2. Processing

Processing is an open source programming language and integrated development environment (IDE) built for the purpose of teaching the fundamentals of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks. The language builds on the Java language, but uses a simplified syntax and graphics programming model.

VII. WORKING

The MPU6050 IMU and ATmega328P microcontroller is interfaced using I2C serial communication based two wire serial interface which is connected to computer using USB UART converter.

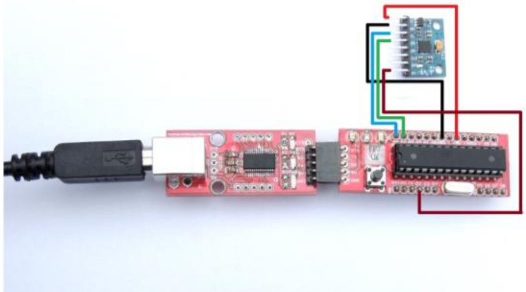

Figure 2 - Hardware Connections

The raw data is then extracted from three axes Accelerometer and Gyroscope and displayed on the serial port.

Accel:	1400	-588	15488	Gyro:	-516	-120	-111
Accel:	1360	-612	15344	Gyro:	-524	-96	-97
Accel:	1312	-680	15444	Gyro:	-538	-58	-93
Accel:	1288	-520	15476	Gyro:	-532	-113	-101
Accel:	1300	-748	15392	Gyro:	-535	-85	-130
Accel:	1308	-708	15600	Gyro:	-536	-62	-94
Accel:	1316	-628	15468	Gyro:	-529	-110	-103
Accel:	1380	-576	15276	Gyro:	-544	-40	-99
Accel:	1284	-592	15396	Gyro:	-537	-103	-99
Accel:	1344	-588	15464	Gyro:	-525	-75	-94
Accel:	1268	-608	15332	Gyro:	-521	-88	-123
Accel:	1284	-580	15412	Gyro:	-546	-84	-90
Accel:	1360	-716	15464	Gyro:	-519	-91	-103
Accel:	1392	-612	15516	Gyro:	-539	-98	-104
Accel:	1344	-568	15252	Gyro:	-535	-77	-76
Accel:	1364	-616	15416	Gyro:	-508	-115	-100
Accel:	1412	-596	15464	Gyro:	-539	-70	-83
Accel:	1260	-552	15476	Gyro:	-522	-85	-80
Accel:	1284	-632	15524	Gyro:	-553	-92	-117
Accel:	1412	-572	15512	Gyro:	-540	-108	-108
Accel:	1288	-584	15348	Gyro:	-530	-68	-93
Accel:	1296	-464	15640	Gyro:	-516	-74	-128

Table 1 – Raw Values

The raw data from Accelerometer is noisy on short time scales, and raw data from gyroscope drifts on longer timescales, so these data must be converted into something more stable and accurate. To make it so, DMP data fusion is used, which utilizes complementary filter algorithm to combine raw data from accelerometer and gyroscope and to produces an output which is in the form of 1X4 matrixed 4bit Quaternion values.

Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.12
Quaternion:	0.99	0.01	-0.02	-0.13
Quaternion:	0.99	0.01	-0.02	-0.13
Quaternion:	0.99	0.01	-0.02	-0.13
Quaternion:	0.99	0.01	-0.02	-0.13
Quaternion:	0.99	0.01	-0.02	-0.13
Quaternion:	0.99	0.01	-0.02	-0.13
Quaternion:	0.99	0.01	-0.02	-0.13

Table 2 - Quaternion values

Quaternion values in the IMU visualization format is used to initiate a software OpenGL program in the processing software for IMU visual interpretation which mimics the orientations of the sensor. Hence giving the real time feedback of the yaw, pitch and roll of the unit.

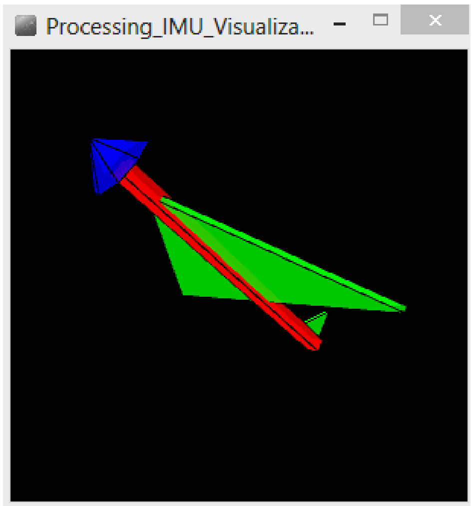

Figure 3 – Visual Interpretation

Complex quaternion values are converted into three axes Euler angles using conversion algorithm which is different for different axis:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \mathrm{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \mathrm{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix}$$

Figure 4 – Quaternion to Euler angles Formula

The converted Euler angles are stable, easy to analyze and efficient and is use to deduce different gestures.

Euler:	12.93	2.16	-0.35
Euler:	12.96	2.16	-0.35
Euler:	12.99	2.16	-0.35
Euler:	13.02	2.16	-0.35
Euler:	13.06	2.16	-0.34
Euler:	13.09	2.16	-0.34
Euler:	13.12	2.16	-0.35
Euler:	13.15	2.17	-0.35
Euler:	13.19	2.17	-0.35
Euler:	13.22	2.17	-0.35
Euler:	13.25	2.17	-0.35
Euler:	13.28	2.17	-0.35
Euler:	13.32	2.17	-0.35
Euler:	13.35	2.17	-0.35
Euler:	13.38	2.17	-0.35
Euler:	13.42	2.17	-0.35
Euler:	13.45	2.17	-0.35
Euler:	13.48	2.17	-0.35
Euler:	13.51	2.17	-0.35
Euler:	13.55	2.17	-0.35
Euler:	13.58	2.17	-0.35
Euler:	13.61	2.17	-0.35
Euler:	13.64	2.17	-0.35
Euler:	13.67	2.17	-0.35

Table 3 – Euler Angles

Euler values along with flex sensor's analog values are wirelessly transmitted to receiver connected to the host processor. It analyzes the received values and identifies different gestures. Based on the different threshold of the gesture different actions of the peripherals are initiated.

In this system rover is controlled using different hand orientation its forward and backward movement is controlled by upward and downward orientation of the hand, while right and left movement can be controlled by flexing the index finger and little finger respectively to which flex sensors is attached

VIII. CONCLUSION

Gestural Controlled User Interface has the potential to change the way users interact with computers/machines by eliminating traditional input devices such as remotes, joystick, mouse and even keyboard allowing the body to give signals to the computer through gestures.

IX. FUTURE SCOPE

There is lot of scope for improvement to improvise Gesture Controlled User Interfacing System by making it more user compatible and flexible. The overall system can be improved by interfacing muscle and nerve sensors, which detects nerves and muscle signals used to make different gestures thus using it as a control interface. It is subjected to improvement based on the futuristic improvement of the sensors and research in sensing technologies.

REFERENCES

[1] Thomas G Zimmerman, Jarot Lanier, Chuck Blanchard, Steve Bryson and Young Harvill, " A Hand Gesture Interface Device", VPL Research, Inc, 656 Bair Island Road, Suite 304 Redwood City, CA 1987

[2] William T Freeman and Michal Roth, "Orientation Histograms for Hand Gesture Recognition", Mistubishi Electric Research Laboratories, Cambridge Research Centre Decembe 1994

[3] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 7, JULY 1997

[4] Lementec, and J Bajcsy, P, "Recognition of arm gestures using multiple orientation sensors : gesture classification", Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference, October 2004

[5] Rochelle O'Hagan, and Alexander Zelinsky, "Finger Track – A Robust and real time gesture interface", Lecture Notes in Computer Science Volume 1342, 1997, pp 475-484 July 2005

[6] Kye Kyung Kim, Keun Chang Kwak, and Su Young Ch "Gesture Analysis for Human Robot Interaction", Advanced Communication Technology, The 8th International Conference (Volume:3) 2006

[7] Raheja J L, Shyam R, Kumar U,and Prasad P.B "Real Time Robotic Hand Control using Gestures", Machine Learning and Computing (ICMLC), 2010 Second International Conference, February 2010

[7] Ruize Xu, Shengli Zhou and Li W J, "MEMS Accelerometer based Non-Specific User Hand Gesture Recognition", Sensors Journal, IEEE (Volume:12 , Issue: 5) May 2012

[8] M Shakunthaladevi, and R Revathi. "Real time Hand Gesture Recognition using AVR Microcontroller", International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 5, May 2014

[9] Prasad S, Kumar P, and Sinha K.P, "A wireless dynamic gesture user interface for HCI using Hand data glove", Contemporary Computing (IC3), Seventh International Conference August 2014