International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Automatic Plane Level Maintenance Mechanism for Robotic Vehicles Employing Accelerometer

M. Mahesh Kumar[1], Munagala Venkata Rishivarun Reddy[2], Baggari Naga Kalyan[3], Bandaru Mahesh[4], Gunde Sivakumar5, Chakali Naveen Kumar[6]

[1]Assit Prof., Department of Electronics and Communication, Santhiram Engineering College, India
[2, 3, 4, 5, 6]Department of Electronics and Communication, Santhiram Engineering College, India

ABSTRACT

This mechanism is very similar to self balancing robot mechanism. The major change here is that, the plane level maintenance is done for maintaining the goods carrying platform of the vehicle at a plane level. This can be done by two different mechanisms: by hydraulics or by direct two-axis actuation. In either of the cases, the actual actuation is effected by motor only. A 3D accelerometer is placed on the chassis of the vehicle and the pitch and roll data of the vehicle is collected and dynamically the input for the motors is generated. This is a simple top down approach programming.

Keywords: Accelerometer, Robotics, Arduino

1. INTRODUCTION

The door delivery service by unmanned robotic vehicles and drones has become a practice in these times and it has increased drastically after the current pandemic situation. Robotics is one of the most growing fields in the current scenario with a global market projection of .189 billion USD by 2027. Many companies have already started robotic vehicle and drone based product delivery. Such door delivery of not only luxury and electronic products but also daily needs like groceries can reduce the risk of spreading the virus to a great extent. One problem with the robotic vehicle based delivery for groceries is that there are breakables like eggs and glass jars that may contain jam and other edibles that have to be transported with care. In a lot of developed and developing countries, where the roads are good, there is one more problem that has to be taken into consideration: the speed breakers. The speed breakers cause pitch motion to the vehicle and almost negligible or no roll or yaw motion. Therefore a single axis servo motor based plane maintenance system is introduced that adjusts its position based on a 3D Accelerometer sensor.

2. PROPOSED SYSTEM

In the proposed system, a 3D accelerometer, an Arduino UNO and a servomotor is employed. In vehicles plan level maintenance is essential in order to prevent accidents. The 3D accelerometer measures the acceleration with a nominal full-scale range of +/- 2 g. It also measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration, resulting from motion, shock, or vibration. It can measure the acceleration in three dimensions (3 axes) independently of each other. In this system Arduino UNO is used to process the data received from the accelerometer. When the accelerometer is tilted on one side the servo meter moves on the opposite side and vice-versa maintaining the plane level.

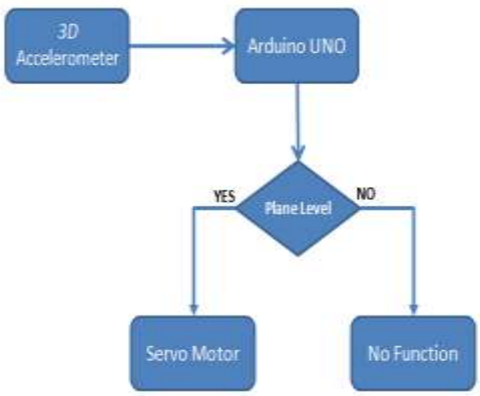

Figure 1. Block Diagram

3. ARDUINO MICROCONTROLLER

“The microcontroller used to implement this project and similar projects are Arduino boards. Presented below is an outline of Arduino family boards.”[2]

3.1 Introduction

“Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. It’s hardware products are licensed under a CC-BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors”[2]

“Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the Arduino language. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) and a command line tool (arduino-cli) developed in Go”[2]

“The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors. The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014”[2]

3.2 Hardware

“Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available”[2]

“Although the hardware and software designs are freely available under copy left licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in –duino”[2]

“An early Arduino board with an RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left.”[2]

“Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lily Pad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.”[2]

“Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default boot loader of the Arduino Uno is the Optibootbootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used. An official Arduino Uno R2 with descriptions of the I/O locations. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.”[2]

“Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent, but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.”[2]

3.3 Software

"A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer)."[2]

IDE:

"The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) that is written in the Java programming language. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2."[2]

"The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware."[2]

Pro IDE:

"On October 18, 2019, Arduino Pro IDE (alpha preview) was released. The system still uses Arduino CLI (Command Line Interface), but improvements include a more professional development environment, auto completion support, and Git integration. The application frontend is based on the Eclipse Theia Open Source IDE. The main features available in the alpha release are:"[2]

Sketch:

"A sketch is a program written with the Arduino IDE.[64] Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension."[2]
"A minimal Arduino C/C++ program consists of only two functions:

setup( ):

This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function main ().

loop ( ):

After setup () function exits (ends), the loop() function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. It is analogous to the function while."[2]

4. INDUSTRY 4.0

"Wikipedia defines Industry 4.0 as thus: The Fourth Industrial Revolution (4IR or Industry 4.0) is the ongoing automation of traditional manufacturing and industrial practices, using modern smart technology. Large-scale machine-to-machine communication (M2M) and the internet of things (IoT) are integrated for increased automation, improved communication and self-monitoring, and production of smart machines that can analyze and diagnose issues without the need for human intervention."[3]

Automation under Industry 4.0 has a particular schema or pattern at its outset. Presented below is how automation in the mass production industry as well as consumer level products are built in today's technological era.

The schema presented in Figure 2 has a lot of other components involved but the generic outline of it stands justifiable for all kinds of automation today.

The software automaton of the conventional automation model, which is the status quo, was built by a human expert or a team of human experts till now. With the advent of machine learning technology, the software automaton was not fully directly designed by human experts. The human experts build the machine learning software and give the real world data set as training information. The machine learning software identifies the pattern between the input and the output parameters of the dataset in the form of a mathematical model. This mathematical model can be downloaded as a working software module to other electronic computing devices. This mathematical model is referred to as the 'trained machine learning module'. The software automaton of all the current digital embedded devices is a mathematical model that gives a numerical output for a numerical input based on arithmetic and logical conditions. This software automaton, as explained above can be either directly developed by a set of human experts by means of setting the boundary conditions themselves based on observation and requirement or can be downloaded as an executable module from machine learning training systems that are trained with relevant dataset. In whatever way the software automaton is developed, it can be loaded onto the relevant embedded computing module that can be used for either sensor based closed loop automation or open loop automation.

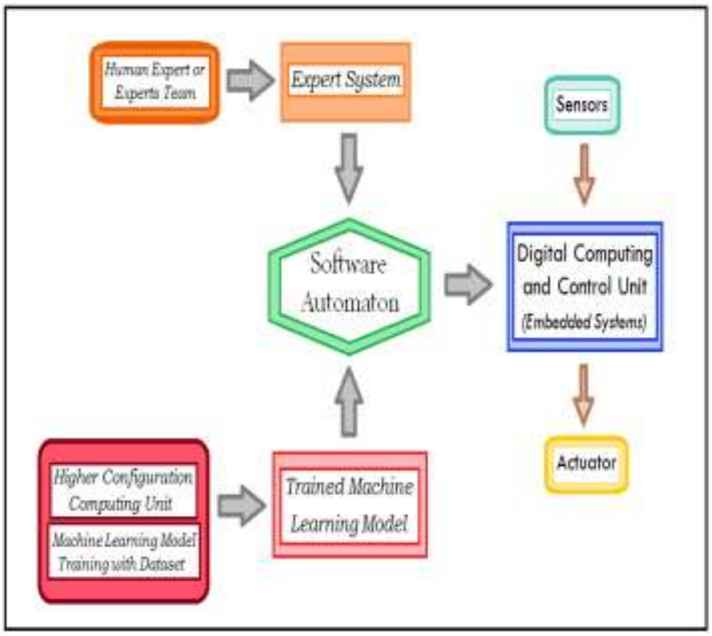

Figure 2. Schema of Automation

The technological components of Industry 4.0 includes IoT, augmented reality, virtual reality, cloud computing, 3D printing, big data analytics, networking, data security, human-machine interaction and others. IoT is a very effective way to collect real world data. Sensors integrated with data acquisition and transmission systems can be placed anywhere and the collected data can be pre-processed if required and used as datasets to train machine learning models.

Cloud computing is employed for optimized utilization of computing resources. There are many third party vendors like Google and Amazon which are very reliable in terms of data security and speed of computation. These services offer companies and organizations a cheap and reliable way to harness the power of artificial intelligence and machine learning.

Big data analytics is the set of technological components involved with collecting, collating and managing large quantities of data for analytics and decision making. When so much data is involved, especially with third party service providers, data security plays an important role.

One of the paramount concerns about Industry 4.0 is the unemployment it can create due to powerful automations. The field of human-machine interactions and co-working has been a very developing field now to mitigate the above mentioned problem.

5. WORKING

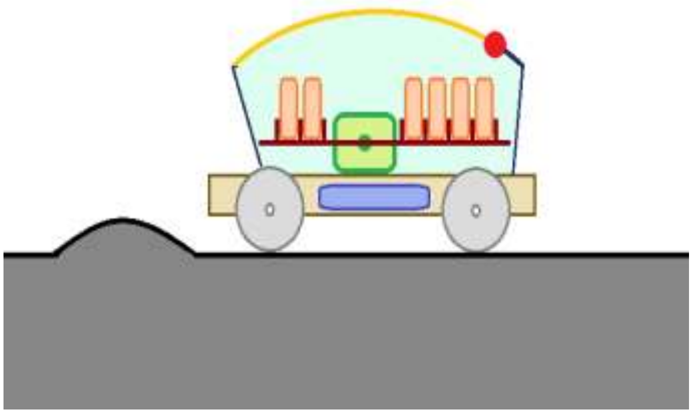

Figure 3. Schematic Diagram

Presented in Figure 3 is a robotic goods delivery vehicle with a single axis plane level maintenance inside. This pitch rotary motion of the vehicle is caused when a speed breaker is being climbed. To maintain the plane balance of the goods in this scenario, a single motor with a platform is provided, on which the goods are arranged. Based on the pitch motion detected by the accelerometer sensor placed on the robot chassis, the motor shaft is rotated to maintain the plane levelness of the platform.

Figure 4. Plane level maintenance 1

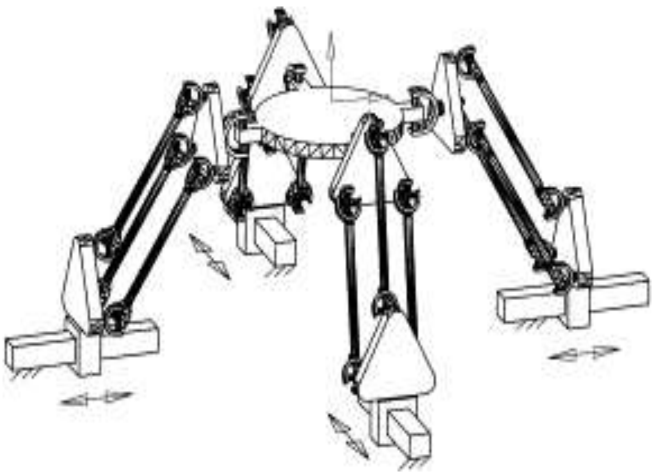

Figure 5. Plane level maintenance 2

Two images of plane level maintenance for three axis systems are presented in Figure 4 and 5.

6. RESULTS AND DISCUSSION

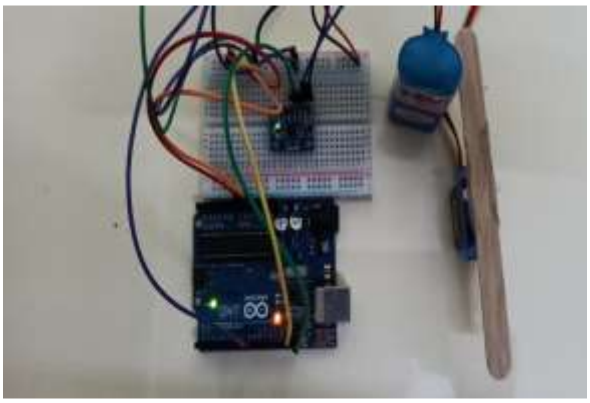

Figure 6. Arduino Setup

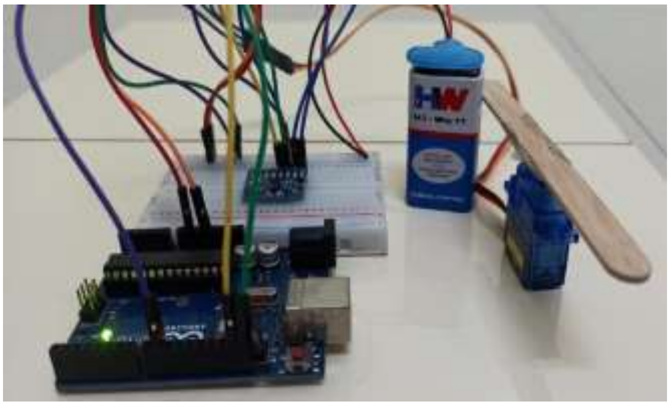

Figure 7. Output Image Top view

As shown in the Figure 6 and Figure 7, the circuit of the system consists of an 3D accelerometer, an Arduino Uno board and a servo motor. This is an ideation level prototype of the product presented in this paper. As mentioned the level maintenance platform is built only for the pitch rotary motion. When the position of the 3D accelerometer is changed, the servo motor shaft rotates correspondingly. This system can be incorporated in robotic vehicles which have to only climb the speed breakers. In this particular ideation level prototype, the model was tested and the results were satisfactory in terms of accuracy and speed. In the future, a system can be designed similarly for maintaining the plane level balance for all three rotary axes, viz., the pitch, the roll and the yaw. A PID controller can be employed to maintain the stability of the system.

7. CONCLUSION AND FUTURE WORK

The door delivery robot presented in this mechanism is designed in such a way that only pitch rotary motion corresponds to the speed breakers climbing. For deliveries in military requirements and also in rescue operations, delivery of medicines and other breakable things have to be done with a three axis automated balancing system corresponding to pitch, yaw and roll. That system can be mechanically complicated to design, fabricate and install. The idea of self balancing began with balancing inverted pendulums. The most common and effectively used algorithm for self-balancing robots is the PID (Proportional, Integral, Derivative) Algorithm. This system has found a great deal of industrial applications. There has to be other integrated technology along with the one presented in this paper to avoid roll motions. The yaw motion can be managed without almost any damage to the contents inside the robot. Even in yaw motion, there needs to be an identified threshold speed with which the robot must turn. The roll motion must be avoided with integrated technology like image processing or non contact reflect technique. The ultrasonic sensor based reflection when fixed on the front sides of the robot can detect an uneven pothole wherein only either the left side or the right side of the robot will get into. The above mentioned scenario has to be sensed beforehand and avoided.

References

[1]. Jun He and Feng Gao, “Mechanism, Actuation, Perception, and Control of Highly Dynamic Multilegged Robots: A Review”, Chinese Journal of Mechanical Engineering

[2]. https://en.wikipedia.org/wiki/Arduino

[3]. https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution