Palestine Polytechnic University

Collage of Engineering

Mechanical Engineering Department

A Mobile Robot that Recognizes Different Terrains

Research submitted to fulfill the requirements of the seminar course in Mechanical Engineering

Prepared By

Ilham Faisal Elayyan

Supervisor

Dr. Hashem Tamimi

2013- 2014

Acknowledgment

I advance great thanks to our great praised Allah who innovates our souls, lights our brains and illuminates the road of tomorrow ...

I advance deep thanks to their dear supervisors and Dr. Hashem Tamimi Who have granted their support, orientation, guidance and advices...

I advance special thanks ...to our, families, dear teachers, lecturers, friends, and to all who contributed in the accomplishment of this project ...

We can only say for their gratitude... Thank You...!"

اهداء

الى منبع الحب والعطاء، من كان لي نعمَ السَّند في السرّاءِ والضرّاء، من يصغرُ امام حِلمِهِ و تضحيته كلّ ثناء ، الى أبي الغالي

الى التي علمتّني أولى الكلمات، وذرفَت حنانا لأُجليَ العَبَرات، الى التي بصبرها وعطائها غدت سيدة السيدات، راحة القلب ومنبع الرحمات ، أمّي

الى فرحة عمري الكبرى ،توأم روحي ورفيق دربي ،وأنيس غربتي ، من لم يكن ليتمَّ هذا الانجاز الا به، زوجي الحبيب ، حسام ، وعرفانُ موصولٌ بالحب لوالديه ، عمي وزوجته" أمي الثانية " الكرام

الى أكبر قلبين عرفتهما في حياتي، من كانا لي بمثابة الوالدين بحنانهما ورعايتهما وعظيم عطائهما ، جدّي زكريّا وجدّتي هدى ، داما لي في هذه الحياة سندًا وذخرا

الى ريحانة فؤادي ومن لازمني في كل حين قبل أن يراني ، من أحببته واشتقت لرؤيته طويلاً ، ولدي بإذن الله عز الدين

الى كل افراد أسرتي وأحبتي الذين أحاطوني بالحب والعون طيلة مسيرة حياتي ، شقيقاتي وشقيقاي ، صديقاتي الصدوقات ، زاد الطريق ،لولاكم جميعا بعد الله عز وجل ما هانت علي الصعاب

وختامًا مسكِيًا ، أهدي جهدي المتواضع هذا الى عظماء هذا الزمان ،الذين اختاروا أن يعيشوا حياتهم أحرارا في زمن العبودية للطاغوت، ودفعوا في سبيل ذلك من اعمارهم واجسادهم أثمانًا ، فلمثلهم تنحني الهامات ، وامام انجازاتهم تصغر الانجازات

الهام

Abstract

The main idea of the project based on that the vibrations that induced in the body of a mobile robot while moving on a ground are different from terrain type to another, these vibrations can be used to make the robot learn to discriminate between different surfaces and to classify the current terrain.

This project aims to study a technique for extracting information about terrain features by having a vibrated body and accelerometer, and without needing extra helping sensors such as cameras. Also this research can be improved to have more precise information about the terrains material.

To implement this project, we use machine learning algorithm that can classify the different vibration values into different terrains. A mobile robot is used to test the performance of the system, and it was able to classify three terrains with a rate of success 81% in the online test.

الملخص

يهدف المشروع الى بناء رجل آلي يستطيع أن يعرف طبيعة الأرضية التي يتحرك فوقها ، وذلك بالاعتمادِ بشكلٍ أساسي على المفارقة بين القياسات للاهتزازات الناشئة فيه اثناء الحركة على اسطح مختلفة لها خشونة مختلفة ، وبناءً اعلى طبيعة السطح عليه يمكنه بمهمة محددة، كأن يغيّر من سرعته.

عندما يتحرك الروبوت على أرضية ما تنشأ في داخله اهتزازات ، هذه الاهتزازات تختلف من سطح لآخر ، هذه الاهتزازات يمكن للروبوت تعلُّمها وايجاد فروقات فيما بينها وبالتالي يتمكّن من التعرُّف على السطوح حال شروعه بالحركة عليها حيث يعمل على ارسال رسالة تبين السطح الى جوال اخر .

التحدّي الأكبر في هذا المشروع هو ايجاد تقنية للتعرف على السطوح ومعرفة معلومات عنها من خلال الاهتزازات الناشئة عن الحركة فقط، وكل ما يلزم لهذه التقنية هو مجس للحركة (accelerometer) و جسم يتحرك وينشأ فيه اهتزازات ، دون الحاجة الى مجسّات اضافية مساعدة كالكاميرا .

Table of contents

1

Chapter One

Introduction

1.1 Overview

1.2 Project Idea Description

1.3 Approach

1.4 Requirements and Assumptions

1.5 Literature Review

1.6 System schedule

1.1 Overview

When an outdoor mobile robot navigates on different types of ground surfaces, different types of vibrations are induced in the body of the robot. These vibrations can be used to teach the robot to discriminate between different surfaces and to classify the terrains.

The main challenge in this project is to find a technique for extracting information about terrain features by having a vibrated body and accelerometer, and without the need to extra helping sensors such as cameras. Also this research can be improved to have more precise information about the terrains material.

This research aims to classify different terrains from our environment. The results then will be analyzed and compared with the known theoretical features for the studied material. The main features that will be studied are those that have a relation with the vibrations (roughness and elasticity).

1.2 Project Idea Description

There is a variety of different terrain types in outdoor environments, in which each posing different dangers to the robot and demanding different driving style. In outdoor environments, a mobile robot typically faces many different terrain types.

Some of them are flat and not slippery, therefore the robot can traverse them at relatively high speed. Other ground surfaces are loose, slippery or bumpy, and therefore dangerous. To prevent accidents, the robot has to traverse these regions slowly and carefully. These examples show that the ground surface itself is a possible hazard to the robot in outdoor environments.

The idea of vibration-based terrain classification is to measure the vibration that is induced in the robot while it traverses the terrain. The vibration can be measured at the wheels, the axes or the body of the robot. Usually, accelerometers are used to measure the vibration perpendicular to the ground surface. As different terrain types induce different vibration signals, one tries to learn characteristic vibration signals for each terrain type from training examples. The learned model is then used for classification of unknown data .

The **disadvantage** of the method is that terrain can be classified only while the robot traverses it, but not beforehand.

Advantages are for example, the independence from illumination conditions and from the specific types of sensors, and the high reliability. Thus, vibration-based terrain classification can be used as a stand-alone classifier or in combination with other sensors.

1.3 Approach

The research started by performing simple experiments using hand pulled cart with accelerometer on different surfaces. The next step is to perform this experiments using an electrical toy car at different velocities and different terrains. Experiments and its analysis and results will be done as the steps shown in Figure 1.1.

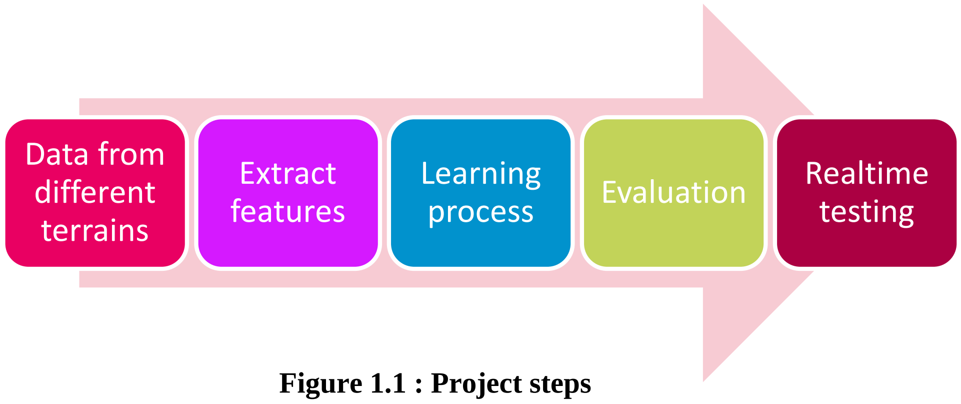


Figure 1.1 : Project steps

1- **Data collection:** start by recording different accelerometer readings for a hand -pulled cart on different terrains.

2- **Features extraction:** samples of theses readings will be analyzed to extract features using SVM kernel models to find the best features that represent the terrain roughness. Features that may be used for example the amplitudes of the output curve that taken by the accelerometer, or the x or y axes values of readings of it.

3- **Learning process:** the features that are extracted from the experiments will be taught to computer (in MATLAB) to be able to identify it when any similar features is given to it, and then it can classify the surface type.

4- **Evaluation**: The learning process will be evaluated by testing it in MATLAB.
5- **Real-time testing:** A mobile robot will be built with accelerometer, this robot will be moved in different terrains and it will give a notification telling that terrain type is guessed.

1.4 Requirements and Assumptions

- We will have a toy car that will move on different terrains and classify the terrain type.
- That will be done after moving for a short distance on that terrain.
- If a terrain change detected while it is moving it will represent a notification.

1.5 Literature Review

Vibration-based terrain classification was first suggested by Iagnemma and Dubowsky[10]. The idea is to measure the vibration that is induced in the robot while it traverses the terrain. The vibration can be measured at the wheels, the axes or the body of the robot. Usually, accelerometers are used to measure the vibration perpendicular to the ground surface (z-acceleration).

As different terrain types induce different vibration signals, one tries to learn characteristic vibration signals for each terrain type from training examples. The learned model is then used for classification of unknown data.

Brooks and Iagnemma examined vibration-based terrain classification for planetary rovers [3, 4]. They use Principal Component Analysis (PCA) to reduce the dimensionality of their data and Linear Discriminant Analysis (LDA) for classification. Sadhukhan and Moore presented an approach based on probabilistic neural networks (PNN) [1, 7]. In [8], Weiss suggested an approach that uses Support Vector Machines (SVM) for classification.

Stevens et al. presented an approach for vehicles driving up to 35 mph [9]. However, they focused on assessing the roughness of the terrain to adapt the velocity, and not on grouping the ground surface into classes

1.6 System Schedule

Table 1.1: *Overall system timing table for the first semester*

TimeLine in Weeks																
Task/weeks	First Semester															
	1th	2th	3th	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	16th
Searching about robot	■	■	■	■												
Collection Data From surfaces					■	■	■	■	■	■						
Preparing software skills for the project			■	■	■	■	■	■	■	■	■					
Build up the software in Matlab							■	■	■	■	■	■				
Learning SVM techniques										■	■	■				
Extraction parameter From Matlab											■	■	■			
Build up the software in Android													■	■		
Testing and simulation in Matlab and in Android Device										■	■	■	■	■		
Documentation writing					■	■	■	■	■	■	■	■	■	■	■	■
Presentation preparing and present															■	■

2

Chapter Two

Theoretical Background

2.1 Overview

2.2 Vibrations (Theory and Sensor)

2.3 Machine Learning

2.4 Mobile Robot

2.5 Android

2.6 PID controller

2.7 Microcontroller

2.1 Overview

This chapter contains basic theoretical information about the theories and technologies used in the project that includes vibrations, roughness, mobile robots, machine learning, and android mobile application.

2.2 Vibrations (Theory and Sensor)

2.2.1 Vibrations

Vibration is a mechanical phenomenon whereby oscillations occur about an equilibrium point. The oscillations may be periodic such as the motion of a pendulum or random such as the movement of a tire on a gravel road.

Vibration is occasionally "desirable". For example, the motion of a tuning fork, the reed in a woodwind instrument or harmonica, or mobile phones or the cone of a loudspeaker is desirable vibration, necessary for the correct functioning of the various devices.

More often, vibration is undesirable, wasting energy and creating unwanted sound – noise. For example, the vibrational motions of engines, electric motors, or any mechanical device in operation are typically unwanted. Such vibrations can be caused by imbalances in the rotating parts, uneven friction, the meshing of gear teeth, etc. Careful designs usually minimize unwanted vibrations [12].

2.2.2 Roughness

Surface roughness, often shortened to roughness, is a measure of the texture of a surface. It is quantified by the vertical deviations of a real surface from its ideal form. If these deviations are large, the surface is rough; if they are small the surface is smooth. Roughness is typically considered to be the high frequency, short wavelength component of a measured surface.

Roughness plays an important role in determining how a real object will interact with its environment. Rough surfaces usually wear more quickly and have higher friction coefficients than smooth surfaces. Roughness is often a good predictor of the performance of a mechanical component, since irregularities in the surface may form nucleation sites for cracks or corrosion, on the other hand, roughness may promote adhesion.

Although roughness is often undesirable, it is difficult and expensive to control in manufacturing. Decreasing the roughness of a surface will usually increase exponentially its manufacturing costs. This often results in a trade-off between the manufacturing cost of a component and its performance in application.

Roughness is typically measured in "RMS" micro inches and is often only measured by manual comparison against a "surface roughness comparator", a sample of known surface roughness’s [11, 13].

2.2.3 Vibrations Sensor

There are several types of vibration sensors. The one that is used in the project is the accelerometer.

An accelerometer is a device that measures proper acceleration. The proper acceleration measured by an accelerometer is not necessarily the coordinate acceleration (rate of change of velocity). Instead, the accelerometer sees the acceleration associated with the phenomenon of weight experienced by any test mass at rest in the frame of reference of the accelerometer device. For example, an accelerometer at rest on the surface of the earth will measure an acceleration g= 9.81 m/s^2straight upwards, due to its weight. By contrast, accelerometers in free fall or at rest in outer space will measure zero. Another term for the type of acceleration that accelerometers can measure is g-force acceleration.

Accelerometers have multiple applications in industry and science. Highly sensitive accelerometers are components of inertial navigation systems for aircraft and missiles. Accelerometers are used to detect and monitor vibration in rotating machinery. Accelerometers are used in tablet computers and digital cameras so that images on screens are always displayed upright.

Single- and multi-axis models of accelerometer are available to detect magnitude and direction of the proper acceleration (or g-force), as a vector quantity, and can be used to sense orientation (because direction of weight changes), coordinate acceleration (so long as it produces g-force or a change in g-force), vibration, shock, and falling in a resistive medium (a case where the proper acceleration changes, since it starts at zero, then increases). Micro machined accelerometers are increasingly present in portable electronic devices and video game controllers, to detect the position of the device or provide for game input [14, 15].

2.3 Machine Learning

Machine learning, is a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. For example, a machine learning system could be trained on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam folders [16].

There are many methods that are used for classification using machine learning like probabilistic neural network (PNN), Brook's Method, decision Trees, k-nearest-neighbor (KNN), Naïve Bayes. The one that will be used in the project is (SVM)[8].

2.3.1 When to use machine learning?

Machine learning is a good method for the problems that are ill defined, and the problem that its data is not ready at design time, also when there is noise in the data, and when the speed and significance is required over precision, also it helps to understand the biological way of learning.

2.3.2 Machine learning system

- Input: many instances of the problem and their expected results (training)

- Process: a general purpose algorithm that can solve the problem only after tainting (brain)
- Results: the solution is usually not optimal.

2.3.3 Machine learning steps:

- Preprocessing (removes noise, adjust light intensity…)
- Segmentation (Remove background, isolate objects)
- Feature Extraction (Length, width, shape, color…)
- Classification

2.4 SVM (Support Vector Machine)

2.4.1. Introduction

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis.

Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible [17].

New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces [17].

2.4.2 Theoretical background:

support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training samples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the samples as points in space, mapped so that the samples of the separate categories are divided by a clear gap that is as wide as possible. New samples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

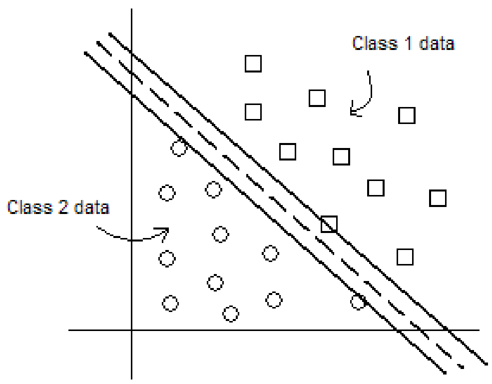


Figure 2.1: SVM margin that separate between classes

A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one “target value” (i.e. the class labels) and several "attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

Given a training set of instance-label pairs (x_i , y_i), i = 1, … ,l where $x_i \in R_n$ and y ∈ $\{1,-1\}^l$ the support vector machines require the solution of the following optimization problem:

$$min_{w,b,\xi} = \frac{1}{2}w^t w + C\sum_{i=1}^{l}\xi_i$$

$$\text{Subject to} \quad y_i(w^t \emptyset(x_i) + b) \geq 1 - \xi_i$$

$$,\xi_i > 0$$

Here training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function $\emptyset$. SVM finds a linear separating hyper plane with the maximal margin in this higher dimensional space. C>0 is the penalty parameter of the error term. Furthermore, $K(x_i,x_j) \equiv \emptyset(x_i)^t\emptyset(x_j)$ is called the kernel function. Though in SVM books the following four basic kernels:

1- Linear : K(x_i,x_j) = $x_i{}^t x_j$
2- Polynomial : K(x_i,x_j) = $(\gamma x_i{}^t x_j + r)^d$, $\gamma > 0$
3- radial basis function (RBF): K(x_i,x_j) = $exp(-\gamma\|x_i - x_j\|^2), \gamma > 0$
4- sigmoid: K(x_i,x_j) = tanh($\gamma x_i{}^t x_j + r$)

Here γ, and d, are kernel parameters.

2.4.3 Procedure of training and testing

In this section the practical steps in doing the SVM training and testing process well be discussed, step by step.

Procedure that proposed in order to get the best results:

1- Transform data to the format of an SVM package:

SVM requires that each data instance is represented as a vector of real numbers. Hence, if there are categorical attributes, we first have to convert them into numeric data. We recommend using m numbers to represent an m-category attribute. Only one of the m numbers is one, and others are zero. For example, a three-category attribute such as (red, green, blue) can be represented as (0,0,1), (0,1,0), and (1,0,0).

2- Model selection:

Though there are only four common kernels mentioned in Section 4.1, we must decide which one to try first. Then the penalty parameter C and kernel parameters are chosen.

3- Use cross-validation to find the best parameter C and γ:

There are two parameters for an RBF kernel: C and γ. It is not known beforehand which C and γ are best for a given problem; consequently some kind of model selection (parameter search) must be done. The goal is to identify good (C, γ) so that the classifier can accurately predict unknown data (i.e. testing data). Note that it may not be useful to achieve high training accuracy (i.e. a classifier which accurately predicts training data whose class labels are indeed known). As discussed above, a common strategy is to separate the data set into two parts, of which one is considered unknown. The prediction accuracy obtained from the \unknown" set more precisely reflects the performance on classifying an independent data set. An improved version of this procedure is known as cross-validation.

In v-fold cross-validation, we first divide the training set into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining v -1 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

The cross-validation procedure can prevent the over fitting problem. Figure 2 represents a binary classification problem to illustrate this issue. Filled circles and triangles are the training data while hollow circles and triangles are the testing data.

The testing accuracy of the classifier in Figures 1a and 1b is not good since it over fits the training data. If we think of the training and testing data in Figure 2a and 2b as the training and validation sets in cross-validation, the accuracy is not good. On the other hand, the classifier in 1c and 1d does not over fit the training data and gives better cross-validation as well as testing accuracy.

It's recommend a "grid-search" on C and γ using cross-validation. Various pairs of (C,γ) values are tried and the one with the best cross-validation accuracy is picked. We found that trying exponentially growing sequences of C and γ is a practical method to identify good parameters (for example, C = $2^{-5}, 2^{-3}, \dots, 2^{15}$, γ= $2^{-15}, 2^{-13}, \dots, 2^{3}$).

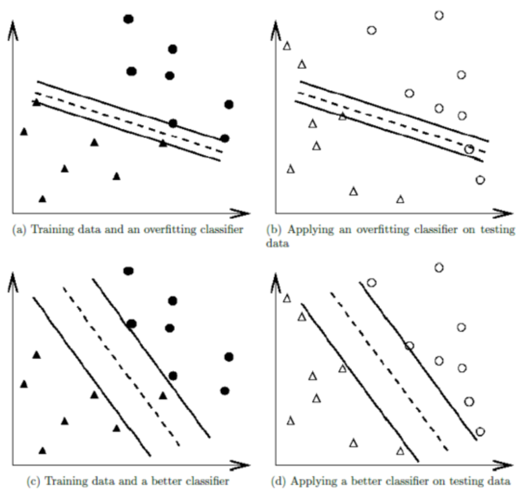


Figure 2.2: An over fitting classifier and a better classifier (● and ▲ are training data O and Δtesting data).

4- Use the best parameter C and γ to train the whole training set.

5- Test.

2.4 Mobile robots

A mobile robot is an automatic machine that is capable of movement in any given environment.

Mobile robots have the capability to move around in their environment and are not fixed to one physical location. In contrast, industrial robots usually consist of a jointed arm (multi-linked manipulator) and gripper assembly (or end effector) that is attached to a fixed surface.

Mobile robots are a major focus of current research and almost every major university has one or more labs that focus on mobile robot research. Mobile robots are also found in industry, military and security environments. Domestic robots are consumer products, including entertainment robots and those that perform certain household tasks such as vacuuming or gardening.

2.5 Android

2.5.1: Android (Operating System)

Android is a Linux-based operating system, designed primarily for touch screen mobile devices such as smart phones and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. The first Android- powered phone was sold in October 2008.

Android is an open source and Google releases the code under the Apache License. This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications ("apps") that extend the functionality of devices, written primarily in a customized version of the Java programming language. In October 2012, there were approximately 700,000 apps available for Android, and the estimated number of applications downloaded from Google Play, Android's primary app store, was 25 billion. A developer survey conducted in April–May 2013 found that Android is the most popular platform for developers, used by 71% of the mobile developer population.

2.5.2: Android (device)

Android devices are powerful mobile computers with permanent internet connectivity and a rich variety of built-in sensors. More properties make the Android system very applicable is that Android uses the Java programming language, which is easy to be familiar with. In addition, the Android API (Application Programming Interface) allows easy access to the hardware components. Interesting for robotics use are the numerous communication interfaces like Wi-Fi, Bluetooth and GSM/UMTS, USB, and the integrated sensors, that is: accelerometer, gyroscope, compass and GPS [18].

2.6 PID controller

A proportional-integral-derivative controller (PID controller) is a control loop feedback controller widely used in industrial control systems, a PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable. The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining $\mathrm{u(t)}$ as the controller output, the final form of the PID algorithm is:

$$u(t) = kp\ e(t) + ki \int_0^t e(\tau)d\tau + kd\ \frac{d}{dt}e(t)$$

Where:

$\boldsymbol{kp}$: Proportional gain, a tuning parameter

$\boldsymbol{ki}$: Integral gain, a tuning parameter

$\boldsymbol{kd}$**:** Derivative gain, a tuning parameter

$\boldsymbol{e}$**:** Error = desired value – output value

$\boldsymbol{t}$: Time or instantaneous time (the present)

$\boldsymbol{\tau}$:Variable of integration, takes on values from time 0 to the present t.

2.7 Microcontrollers

Microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

There are several types of microcontrollers, the most commonly used ones are Microchip Technology PIC, Intel 8051, Atmel AVR.

3

Chapter Three

System Design

3.1 Overview

3.2 General System Block Diagram

3.3 System Main Components

3.1 Overview

This chapter describes the system main parts and the design concepts; it talks about system general block diagram, the system main components.

3.2 General system block diagram

As shown in Figure (3.1) the system contains two parts; offline part and online part. The offline part includes the learning process, starting with data collection. , then the model will be programmed in the robot system to do the suitable reaction according to the type of terrain, classifying the terrain by the robot directly while its moving is the online part of the system.

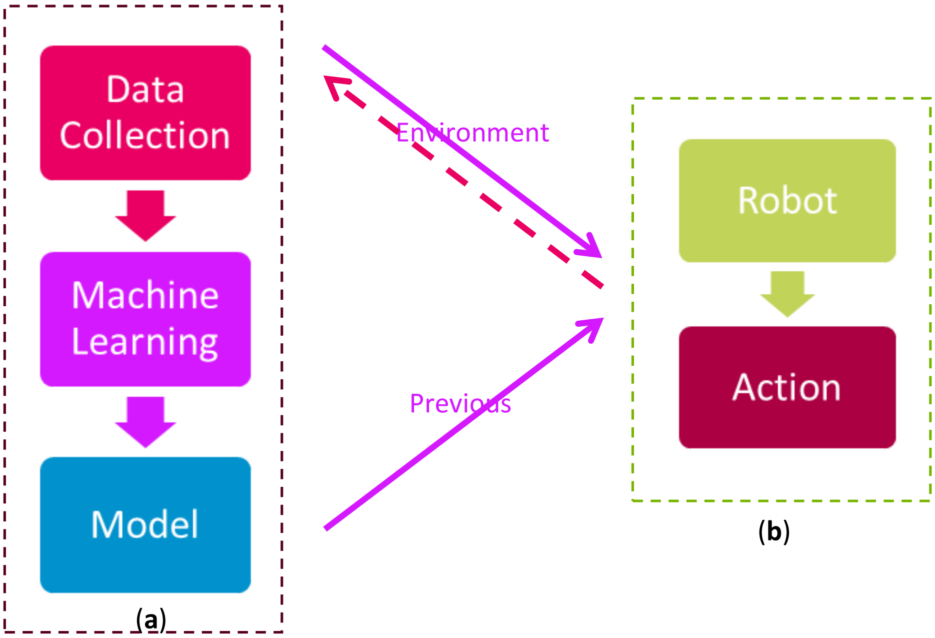


Figure 3.1 : the general system block diagram, (a) the offline part of the project, (b): the online part.

3.3 System main components

The main components of the project are the accelerometer, the mobile robot (for the stand alone data collection stage), PC with MATLAB for machine learning process, then in building all the system an Android device, and outdoor electrical toy car, and a microcontroller. These components and their positions in the project are shown in Figure 3.2.

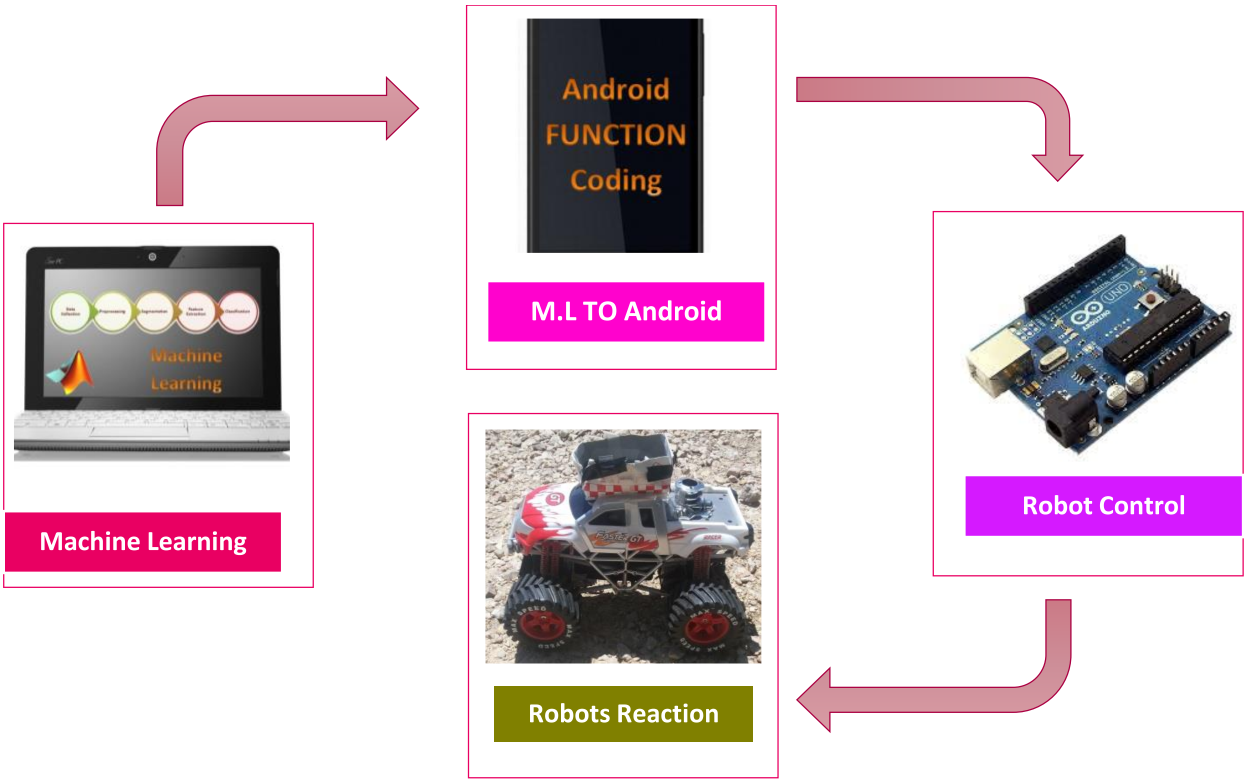


Figure 3.2 : Components general block diagram

4

Chapter Four

Experiments and Results

4.1 Overview

This chapter contains the practical part of the project, MATLAB, and Android work, all the experiments and results, and finally the conclusion.

4.2 Data Collection

Data collection is the first step before starting the machine learning phase as discussed in section 1.4, this section contains an explanation for this step work.

A toy car that simulate a mobile robot was used for data collection, it was moved - using radio remote control- on a terrain while the accelerometer readings being recorded, the readings were taken for long distances on each terrain, this process were done using a mobile phone accelerometer that attached to the car.

Samples of data were collected for grass, limestone, and asphalt terrains, the data were collected using a mobile phone with Android application called Accelerometer Monitor.

4.2.1 Accelerometer Monitor

Android devices have an accelerometer sensor , this accelerometer returns values that describe the changes in acceleration along the three axes of the coordinate system measured in m/x^2 ,t is a sensitive and reports the device vibrations.

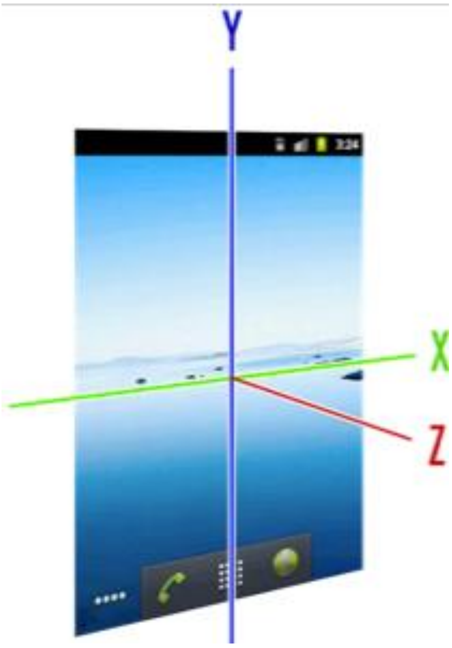
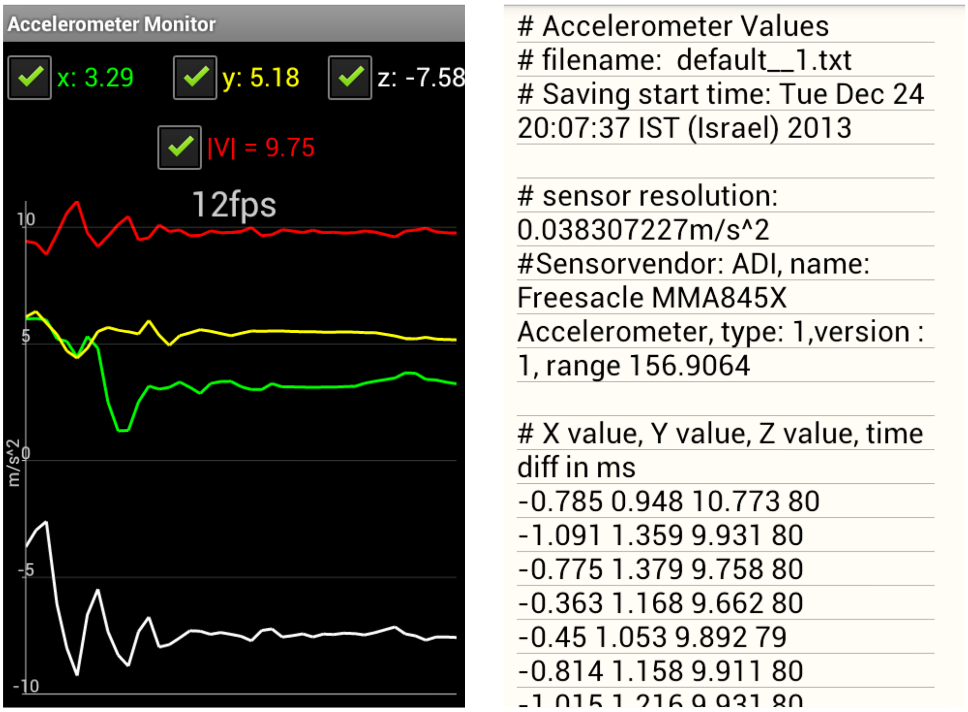


Figure 4.1 :mobile accelerometers axes related to its screen.

Accelerometer Monitor is vibration meter tool to be used in the project since can show vibrations in real time for the three dimensions and save vibration data into SD card as txt file or zip file.



Accelerometer Values
filename: default__1.txt
Saving start time: Tue Dec 24 20:07:37 IST (Israel) 2013

sensor resolution: 0.038307227m/s^2
#Sensorvendor: ADI, name: Freescale MMA845X Accelerometer, type: 1,version : 1, range 156.9064

X value, Y value, Z value, time diff in ms
-0.785 0.948 10.773 80
-1.091 1.359 9.931 80
-0.775 1.379 9.758 80
-0.363 1.168 9.662 80
-0.45 1.053 9.892 79
-0.814 1.158 9.911 80

(a) (b)

Figure 4.2 : Accelerometer Monitor application: (a) :vibration oscillation in real time , (b): Accelerometer values in 3 dimensions.

4.2.2 Data collection results

The data were taken for long distances that gave for each terrain around 10 thousands value for each dimension (x,y,z) , as shown in figures below (figures x-axes are the number of values and y-axis shows the accelerations inm/s^2) :

- **Grass**:

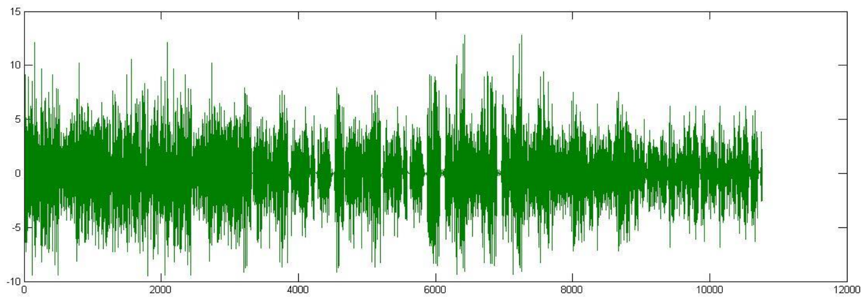


Figure 4.3 :Grass x :Minimum value is -9.4870 , Maximum value is 12.8200

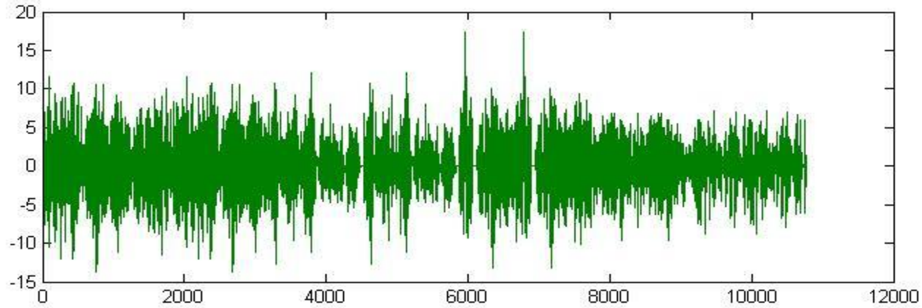


Figure 4.4 :Grass y : Minimum value is -13.66 , Maximum value is 17.3240

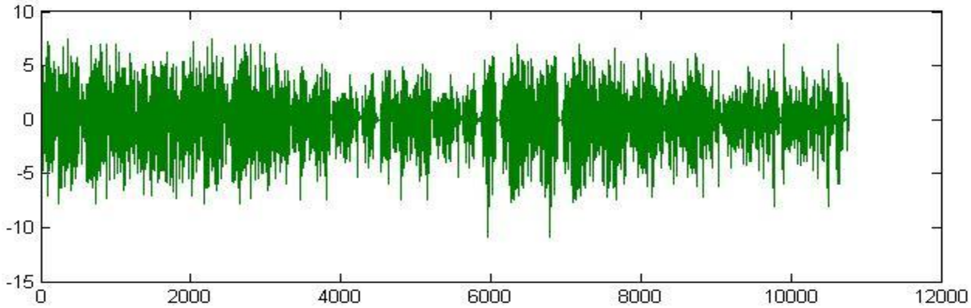


Figure 4.5 : Grass z :Minimum value is -10.8600 , Maximum value is 7.4730

- **Asphalt**:

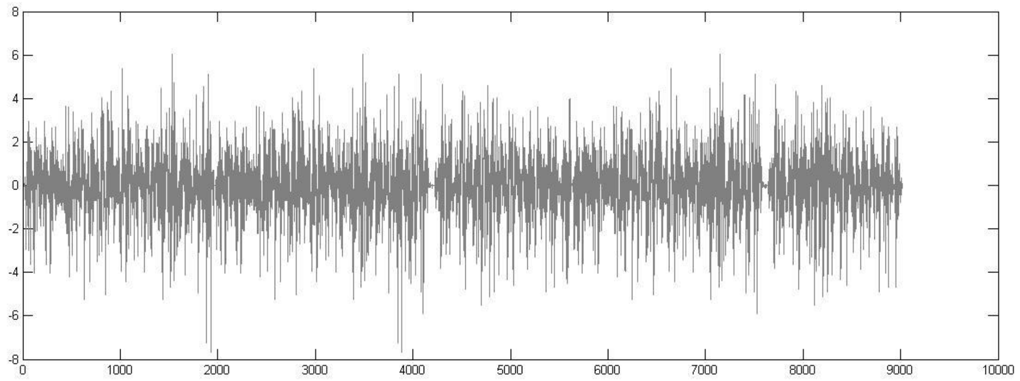


Figure 4.6: Asphalt x : Minimum value is -7.6580 , Maximum value is 6.0060

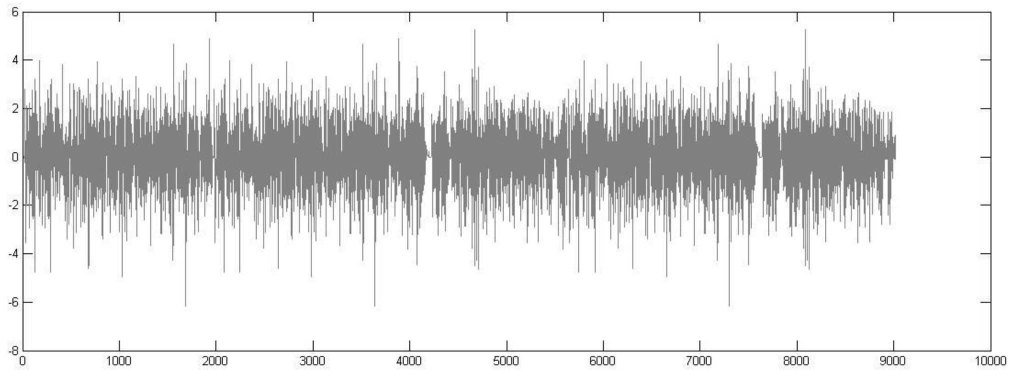


Figure 4.7:Asphalt y : Minimum value is -13.6650 , Maximum value is 17.3240

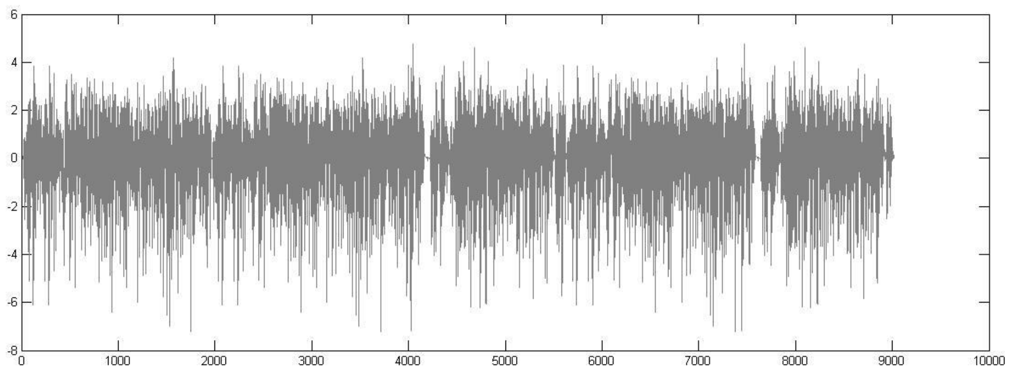


Figure 4.8:Asphalt z : Minimum value is -7.2140 , Maximum value is 4.7640

- **Limestone terrain:**

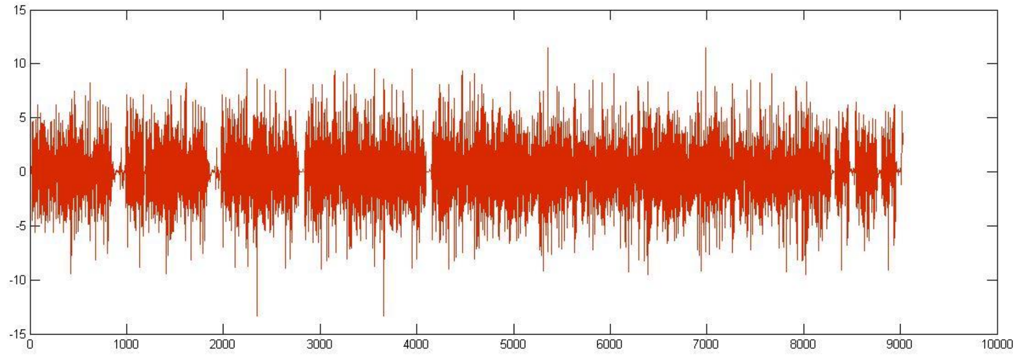


Figure 4.9 : Limestone terrain x : Minimum value is -13.3240 , Maximum value is 11.4450

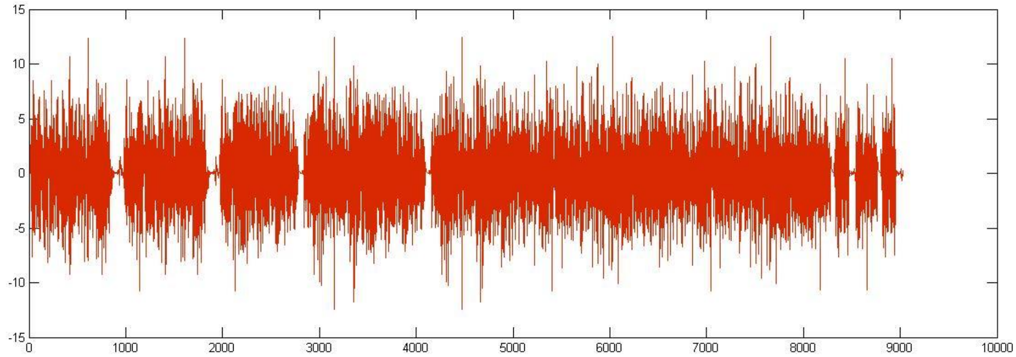


Figure 4.10 :Limestone terrain y : Minimum value is -12.4070 , Maximum value is 12.4810

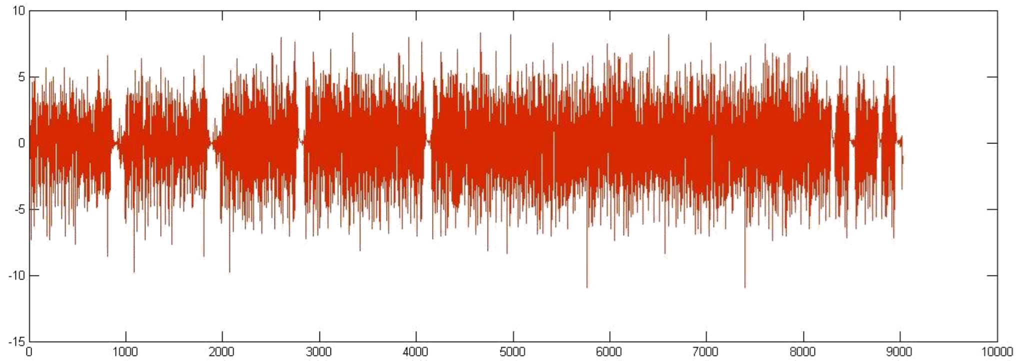


Figure 4.11 : Limestone terrain z : Minimum value is -10.9550 , Maximum value is 8.3210

4.3 Simulation

4.3.1 SVM (MATLAB work)

4.3.1.1 Training

- Transform data to the format of an SVM package:

In this stage of work we took 9000 sample from the data collected, each one have 3 values for the three dimensions (x,y,z), that gave a matrix of dimension (9000*3) , then we reshaped this matrix to have the form

$$\begin{bmatrix} x_1 & y_1 & z_1 & x_2 & y_2 & z_2 & \cdots & x_{300} & y_{300} & z_{300} \\ x_{301} & y_{302} & z_{303} & \cdots & \cdots & \cdots & \cdots & x_{600} & y_{600} & z_{600} \\ \vdots\ \vdots & & \vdots\ \vdots & & & & \vdots\ \vdots & \vdots\ \vdots & \vdots\ \vdots & \\ \vdots\ \vdots & & \vdots\ \vdots & & & & \vdots\ \vdots & \vdots\ \vdots & \vdots\ \vdots & \\ x_{801} & y_{801} & z_{801} & \cdots & \cdots & \cdots & \cdots & x_{900} & y_{900} & z_{900} \end{bmatrix}_{(30*900)}$$

This matrix of data is with (30*900) dimension, we called it the features matrix, it contains a 30 samples each sample contain 900 value as shown, each terrain have its matrix of data , each terrain here is called class and given a label .

The matrices of the three classes together gives a matrix of the dimension (90*900), labeling step is to give the first 30 row of this matrix the value of (1) and the two other 30 rows the value of (0) , and doing that for the second and third thirty rows, preparing to train three support vectors each one learn the features of its data to be able to give a binary decision 1 or 0 .

For the three classes we have (grass, asphalt, limestone) they are labeled respectively as (0,0,1), (0,1,0), and (1,0,0).

4.3.1.2 Testing

Using 10-fold cross validation which means shuffling the data and repeating cross validation 100 times, and then After training the classifier, we measure its accuracy on the validation data, then take the average of the accuracy to get a final cross-validation accuracy.

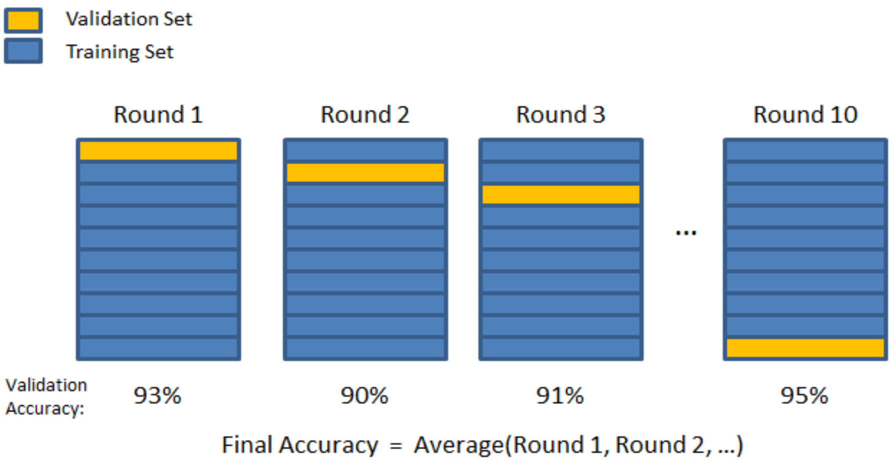


Figure 4.11 : 10-fold cross-validation. The data set is divided into 10 portions or "folds". One fold is designated as the validation set, while the remaining nine folds are all combined and used for training. The validation accuracy is computed for each of the ten validation sets, and averaged to get a final cross-validation accuracy. The accuracy numbers shown here are just for illustration.

- **Model selection:**

We experimented the four common kernels mentioned and took the model that gave the best results, here is the results of testing each model in our data:

1- **Linear** : $K(x_i,x_j) = {x_i}^t x_j$

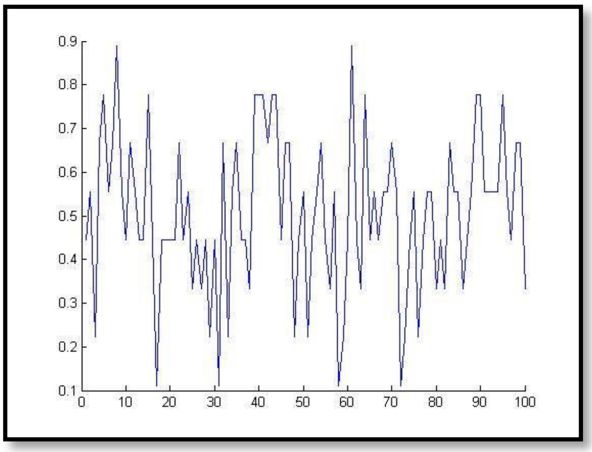


Figure 4.11 : Linear model , average accuracy result is 51%.

2- **Polynomial** : K(x_i,x_j) = $(\gamma {x_i}^t x_j + r)^d$, $\gamma > 0$:

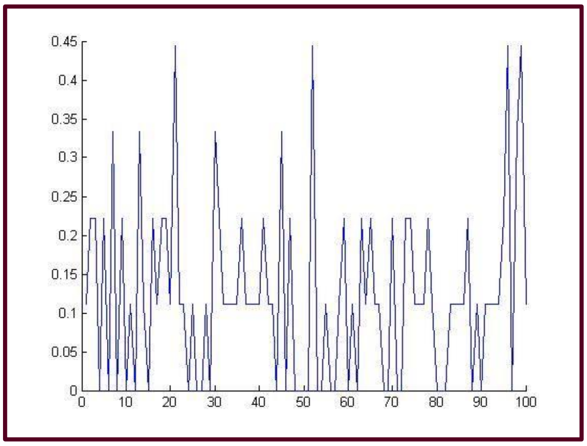


Figure 4.12 :Polynomial model , average accuracy result is 12%.

3- **Radial basis function (RBF):** K(x_i,x_j) = $exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$

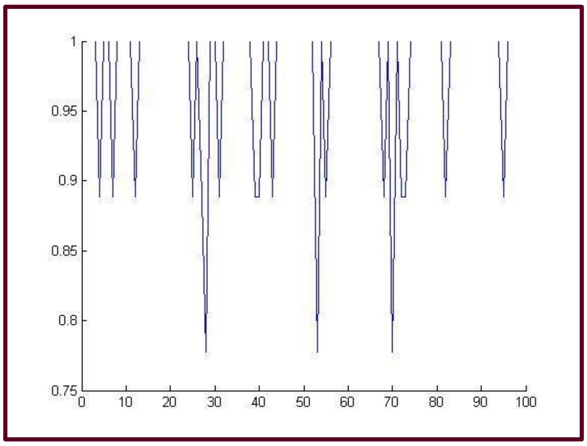


Figure 4.13 RBF model , average accuracy result is 98%.

4- **Sigmoid**: K(x_i,x_j) = tanh($\gamma x_i{}^t x_j + r$)

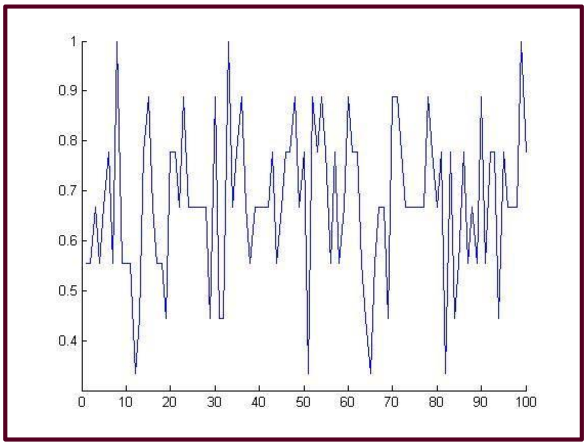


Figure 4.14 :Sigmoid model , average accuracy result is 67%.

As shown above radial basis function (RBF) gave the best accuracy so it was the best model to be selected.

- Selection the values of $\boldsymbol{\gamma}$and **C** :

For the used kernel and in order to have the best accuracy, the parameter C was selected to be 1, and gamma best value was 4.8828e-004.

4.3.1.3 Extracting parameters :

The **'svmtrain'** function in MATLAB returns a model which can be used for prediction, it is a structure and is organized as [**rho sv_coef, SVs**], where,

-**rho**: -b of the decision function(s) wx+b.

-**SVs**: support vectors.

-**sv_coef**: coefficients for SVs in decision functions.

Those three parameters of the model are used to form the decision function:

$$\boldsymbol{s = wt + b}$$

Where:

$\boldsymbol{w}$:is the weight vector, it is converted to a matrix of dimension (m*1)

t: a vector extracted from the training process have a relation with the sv_coef, we took it as a matrix of dimension 1*m.

m: is the number of the support vectors that resulted from the training according to the features of training data, in our case m=60.

Since we have three classes, we got three decision functions, each one can give a decision if the data given to it is belong to its class or not, figure below shows the result of the three s vectors extracted from the code, each one gives a binary result with value that give information about the accuracy, the final decision function gives the decision which have the best value.

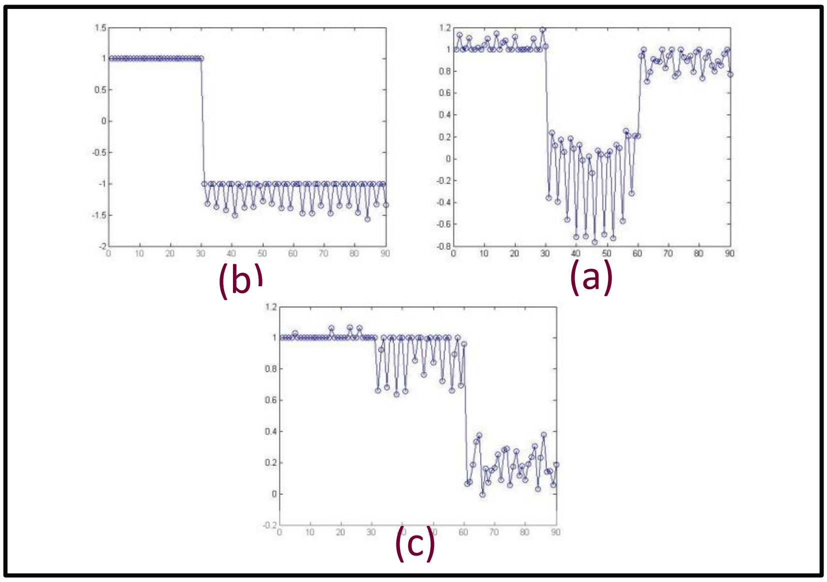


Figure 4.14 : : the three classes decision functions testing results, it is clear that each of them classifying the data binary , belong to class or not (0 or 1).

4.4 Android work

4.4.1 Overview

Now it is the time of applying the results of the successful training operation in a mobile application that sends notification when it detects a change in the terrain that is walking on. This section shows the algorithm that leaded to achieve that and its results.

4.4.2 Program main algorithm

1- Reading the parameters files that extracted from the training process, those parameters matrices are saved as a text files.
2- Online data reading: while program is running it read the data from the accelerometer and store it in a matrix.
3- Applying the decision function: after reading 900 value of the three dimensional accelerometer data, and reshaping it to match the generated model of training process, the data is sent to the decision function as an input to apply the classification model on it.
4- Gives decision: the output of the program is deciding which terrain the given data is belong to and giving a notification telling what is the decision.

4.4.3 Program testing

For testing the program of the application before the online testing we gave it samples of the collected data that used in training to check if it is work correctly, the figure 4.15shows the desktop screen of the application.

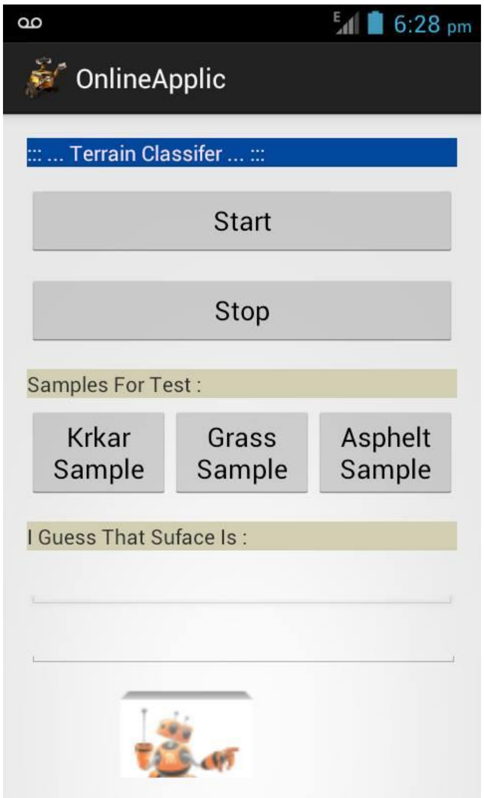


Figure 4.15: : InterFaces of Online Andriod Application

The application work procedure is as follows:

1- Pressing start: when pressing start the application start receiving data from the accelerometer and store it in a matrix.
2- After 30 seconds of data collection the decision function instructions starts to analyze the data matrix and compare it with the three classes support vectors.
3- The application represents the decision as a notification on the mobile screen and by sending a message for another mobile.
4- Backing to step 2, and redo the terrain checking, the application stays in the loop until stop command is pressed.

Figures 4.16 and 4.17 shows how the mobile application was representing the result of detecting the terrain.

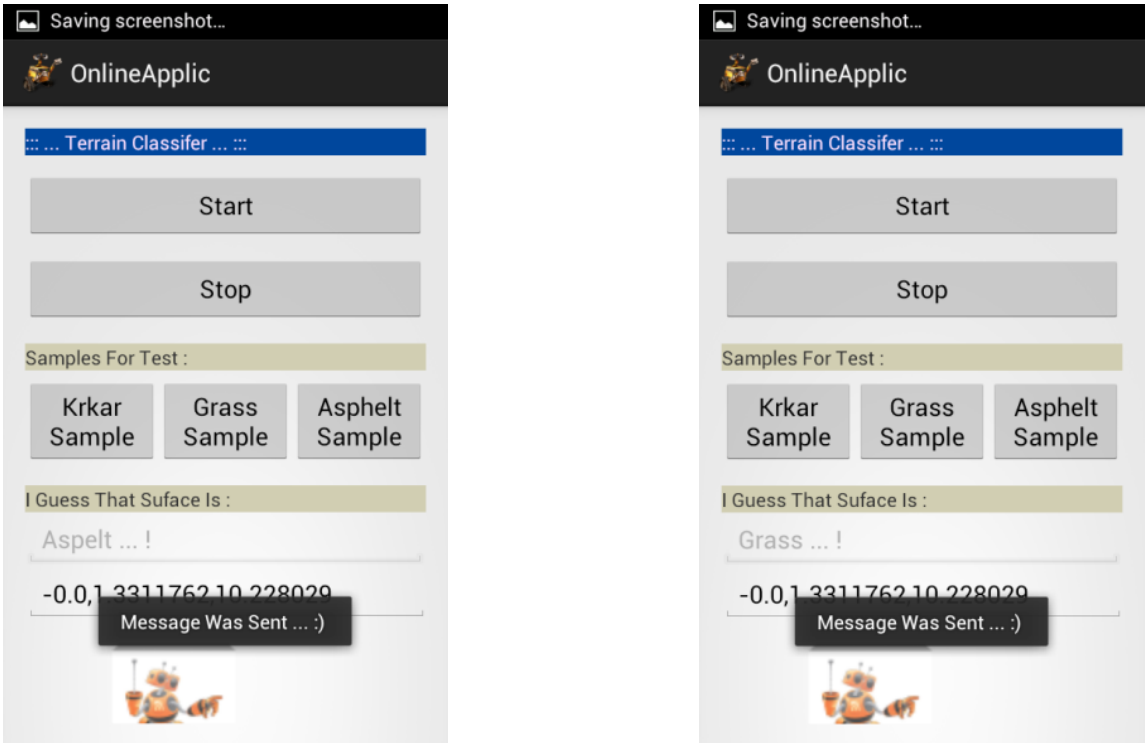


Figure 4.16: : InterFaces of Online Andriod Application

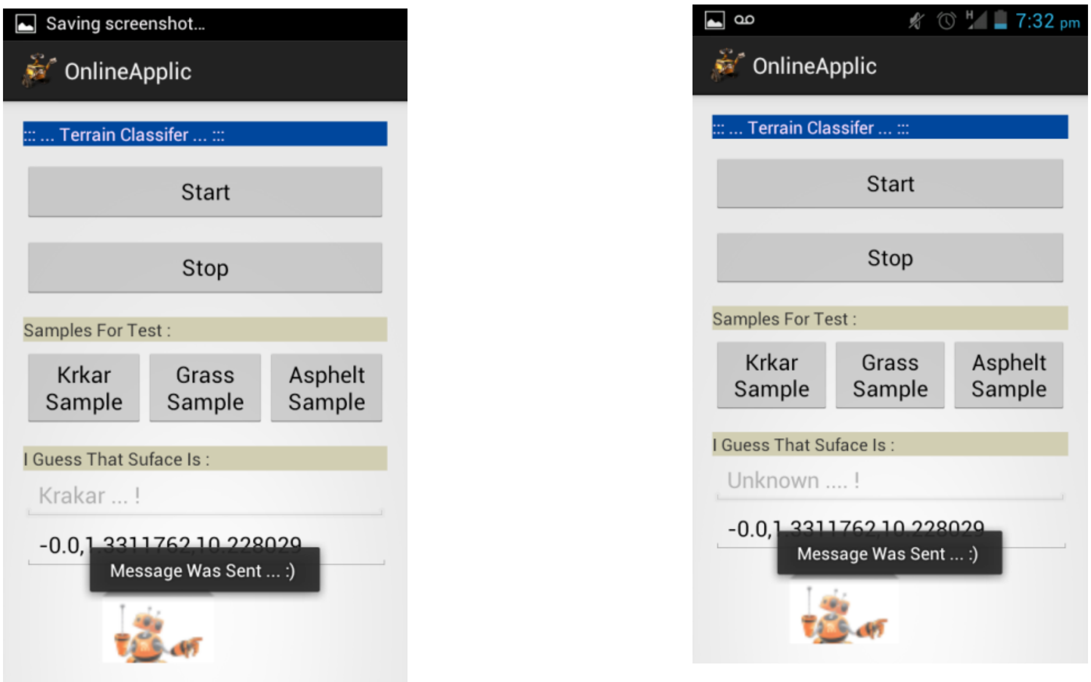


Figure 4.17: : InterFaces of Online Andriod Application

Figures 4.18, 4.19, 4.20 shows a pictures that taken for the system during the online testing.

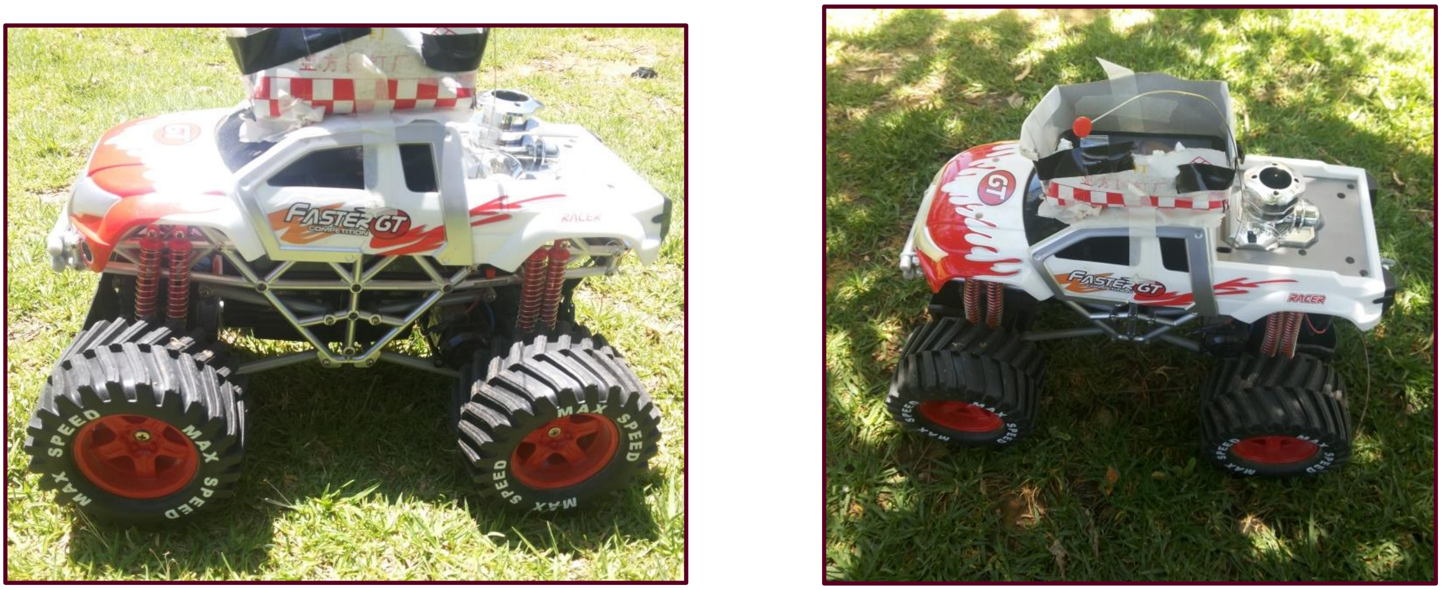


Figure 4.18: : Car In Grass

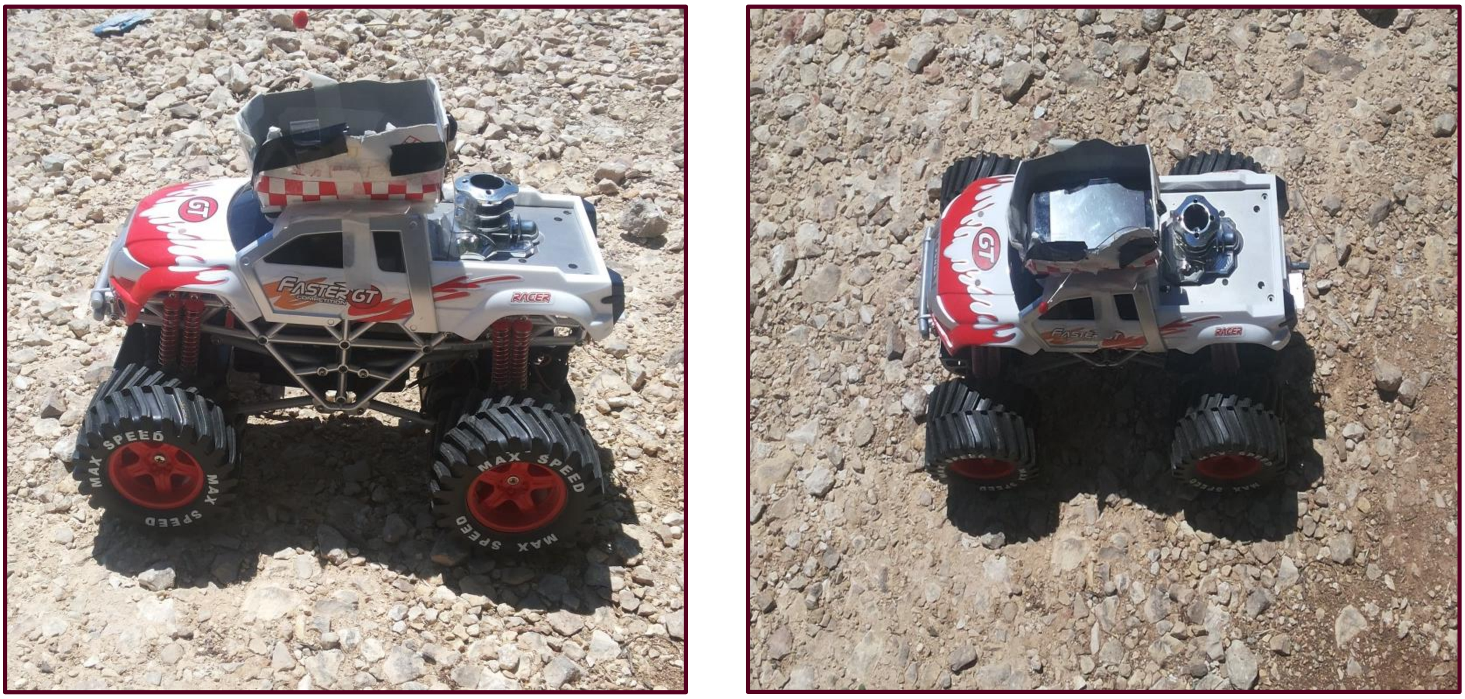


Figure 4.19: : Car In LimStone

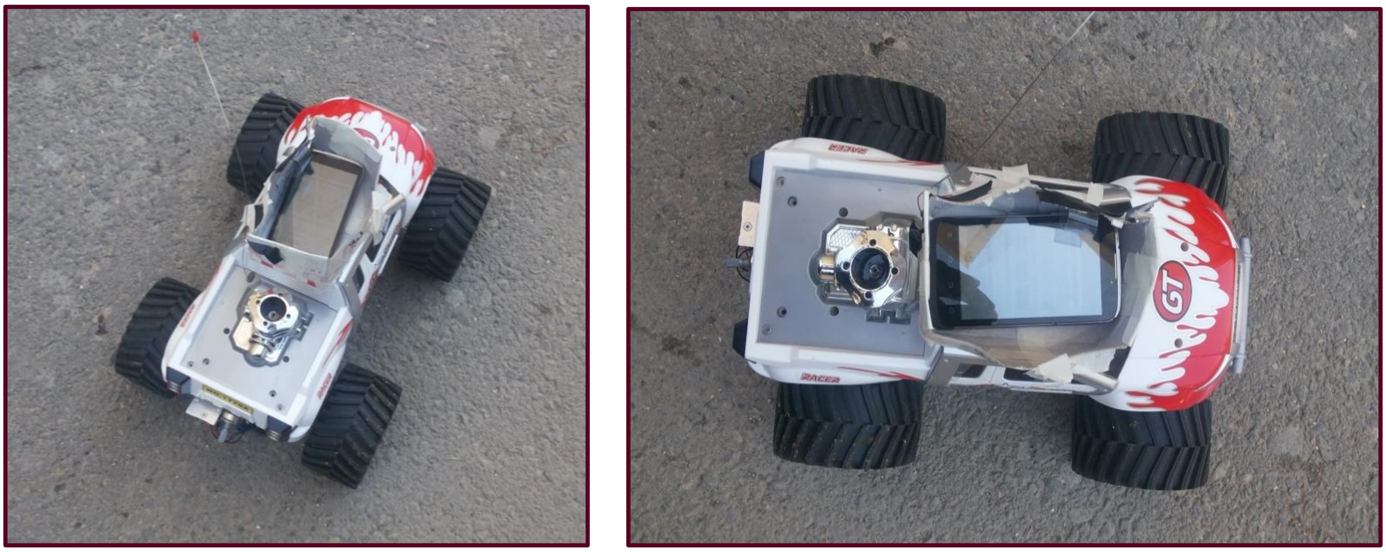


Figure 4.20: : Car In Aspelt

4.5 Online testing results

The mobile application that used in the online testing was set to collect data for 30 seconds and then represent its decision, it was programmed to present it on its screen, and also sends a message to another mobile that change in terrain is detected. We did the online test on different places for the same terrain to see how far the system is successfully works.

Seven online tests were done for each terrain (asphalt , grass, limestone), also we tested the system at different other terrains to see if it can classify it as unknown or not.

Table 4.1 shows the results of online tests done.

	Grass	Asphalt	Limestone
Test #1	✓	✓	☒
Test #2	✓	✓	✓
Test #3	✓	✓	☒
Test #4	☒	✓	✓
Test #5	✓	✓	✓
Test #6	✓	✓	✓
Test #7	☒	✓	✓
Success rate	71%	100%	71%

Table 4.1: the rate of success of the online testing process, total rate is 81%

5

Chapter Five

Feature Work and Recommendation

5.1 Future Work

5.2 Conclusion

5.1 Future Work

5.1.1 Proposed system

A robot that changes its velocity according to the detected terrain

Project results can be used in different applications, in mobile robots, vehicles, and other applications, its decision about terrain can be translated to a reaction to be done by the robot or the vehicle, in our work, we started to build a system that is similar to the system in Fig3.2 and it works as follows:

- While an outdoor robot is moving with the mobile that senses the terrain, THE mobile phone sends a signal that gives information about the detected terrain to the microcontroller
- When the microcontroller receives the signal, he changes its speed according to the terrain by changing the duty cycle of pulses that control its motors speed.
- The robot keeps moving in that speed until a change in terrain be detected, and for that we used a PID feedback controller to have a constant desired speed at each terrain.

5.1.2 Current work

To implement that proposed system, we started with doing the speed control for the car, to be ready to receive the signal of changing its velocity when terrain under it is changed, we started the simulation steps of the speed control process as follows:

5.1.2.1 Difference equation

To find the PID digital difference equation we used the bilinear Tustin method using the transformation

$$c(z) = \frac{u(z)}{e(z)} = \frac{2}{T_S}\frac{1+z^{-1}}{1-z^{-1}} \quad ,, \text{for } s = \frac{z-1}{T}$$

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

For our PID controller the final difference equation that extracted was

$$u(k) = \left(kp + 0.5\,ki + \frac{2kd}{Ts}\right)e(k) + \left(ki - \frac{4kd}{Ts}\right)e(k-1) + \left(\frac{1}{2Ts} - kp\right)e(k-2) - u(k-2)$$

Where:

$\boldsymbol{e(k)}$**:** system error

$\boldsymbol{e(k-1)}$: previous system error

This equation of the feedback digital controller will be programmed in the microcontroller to compare between the desired speed and the measured speed.

5.1.2.2 circuits design

- ✓ The measured speed will be read by the microcontroller by an encoder sensor that its design is shown in the figure 5.1

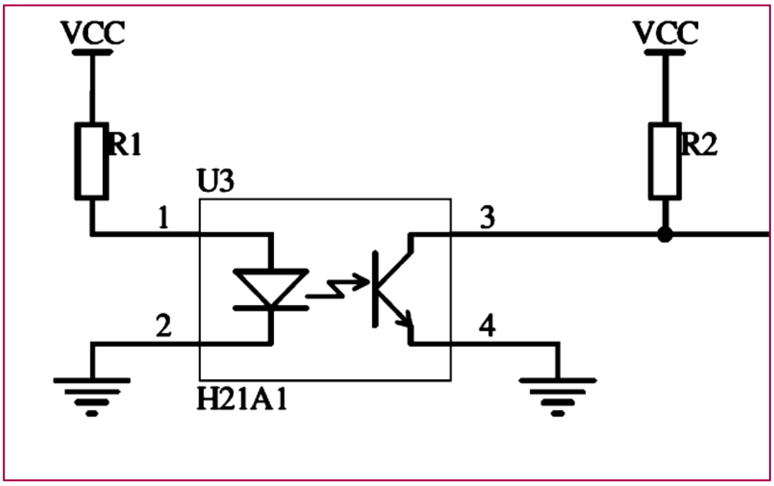


Figure 5.1 Encoder Sensor

✓ The circuit of the speed control system of the car is shown in figure 5.2

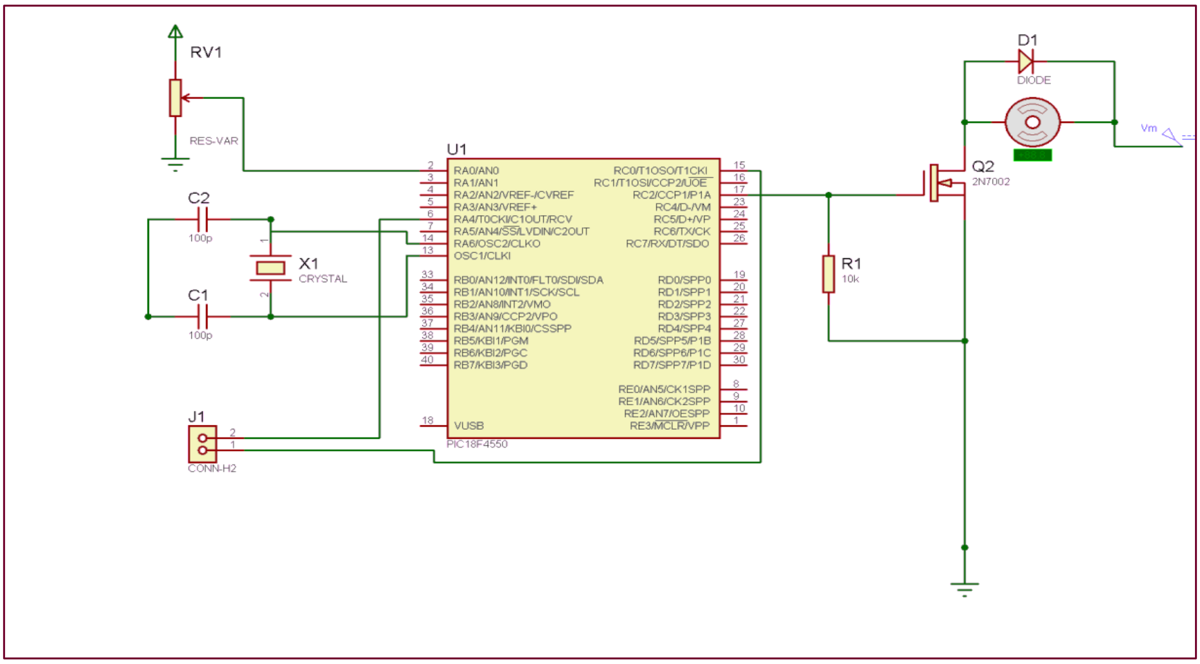


Figure 5.2 Speed Control System

✓ PIC code using Micro C: the code works as the following steps:

1- the car start moving with its maximum speed (100% duty cycle) with controlled fixed velocity by the difference equation.
2- it keeps moving until it receive a signal that there is a change in terrain is detected.
3- the output duty cycle being changed according to the detected terrain.

5.1.3 conclusion of section

This application is in the implementation stage, and we have some challenges to complete it as until now we don't what is the best way to make a communication between the mobile and microcontroller, and for that what is the most suitable microcontroller to be used, whatever we are keeping going on in the project and expect to have a good result in the following few days.

5.2 Conclusion

This project presented a new technology combination for sensing the terrain type. The proposed methodology was by using machine learning tool, and mainly depending on the vibrations that being induced in the body while it is moving instead of depending on camera or other expensive sensors.

SVM is one of the machine learning tools that gave successful results in the case of our project, the system was able to classify three terrains also guess the terrains that is not belong to the trained classes, and that was with a good rate of success.

Working in this project is not stopping here, we will complete trying to have a better classifier that can give more information about the terrains and have more than three classes.

References

[1]: D. Sadhukhan. Autonomous ground vehicle terrain classification using internal sensors. Master's thesis, Dept. Mech. Eng., Florida State University, Tallahassee, Florida, USA, 2004.

[2]: R. O. Dada, P. E. Hart, and D. G. Stork. Pattern Classification. Wiley, 2001.

[3]: C. Brooks, K. Iagnemma, and S. Dubowsky. Vibration-based terrain analysis for mobile robots. In Proc. IEEE International Conference on Robotics and Automation (ICRA), pages 3426–3431, Barcelona, Spain, 2005.

[4]: C.A.BrooksandK.Iagnemma. Vibration-based terrain classification for planetary exploration rovers. IEEE Transactions on Robotics, 21(6):1185 – 1191, December 2005.

[5]: T. T. Mitchell. Machine Learning. McGraw-Hill, 1997.

[6]: C. Weiss, H. Froehlich and Andreas Zell," Vibration-based terrain classification using Support Vector Machines", in International Conference on Intelligent Robots and Systems, 2006.

[7]: D. Sadhukhan and C. Moore. Online terrain estimation using internal sensors. In Proc. Florida Conf. on Recent Advances in Robotics, Boca Raton, FL, 2003.

[8]: C. Weiss,H.Fro¨hlich, and A.Zell. Vibration-based terrain classification using support vector machines. In Proc. IEEE/RSJ International Con- ference on Intelligent Robots and Systems (IROS), pages 4429 – 4434, Beijing, China, October 2006.

[9]: D. Stavens, G. Hoffmann, and S. Thrun. Online speed adaptation using supervised learning for high-speed, off-road autonomous driving. In Proc. International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, 2007.

[10]: K. Iagnemma and S. Dubowsky. Terrain estimation for high-speed rough-terrain autonomous vehicle navigation. In Proc. SPIEConference on Unmanned Ground Vehicle Technology IV, 2002.

[11]: Kohser, Ronald A. (2003), Materials and Processes in Manufacturing (9th ed.), Wiley, p. 223, ISBN 0-471-65653-4.

[12]: [Online] Sinusoidal and Random Vibration Testing Primer

http://www.desolutions.com/blog/2013/04/sinusoidal-and-random-vibration-testing-primer/

[13]: [Online]   Terminology

http://www.mfg.mtu.edu/cyberman/quality/sfinish/terminology.html

[15]: [Online] A beginner's guide to accelerometers

http://www.dimensionengineering.com/info/accelerometers

[14]: Doscher, James. "Accelerometer Design and Applications". Analog Devices. Archived from the original on 13 December 2008. Retrieved 2008-12-23.

[16]: Wernick, Yang, Brankov, Yourganov and Strother, Machine Learning in Medical Imaging, IEEE Signal Processing Magazine, vol. 27, no. 4, July 2010, pp. 25-38.

[17]: Mitchell, T. (1997). Machine Learning, McGraw Hill. ISBN 0-07-042807-7.

[18]: [Online] Android, the world's most popular mobile platform:

http://developer.android.com/about/index.html