Journal of Aeronautical and Automotive Engineering (JAAE)
p-ISSN: 2393-8579; e-ISSN: 2393-8587; Volume 3, Issue 1; January-March, 2016 pp. 10-16

http://www.krishisanskriti.org/Publication.html

Automated Seat Belt Switch Defect Detector

Navoda C.B. Kapukotuwa

Department of Electrical and Computer Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
E-mail: ncbdrck@hotmail.com

Abstract—*Seat belts are crucial when considering about the safety of an automobile. Many lives of passengers in car accidents are saved by the seat belts throughout the history. Generally, the drivers and front seat passengers in cars are required by law to wear a seat belt in almost every country. Therefore, the proper operation of the seat belt in a vehicle is a must. In seat belt harness manufacturing process, the switch of the belt consists of a plastic slider around a PCB where the sensor wires are soldered. The friction between these two components play a major role in the proper operation of the seat belt switch. Usually this is checked using an experienced personal by individually checking the friction of the two components by hand. But this method is outdated and cannot guarantee the proper operation of every seat belt switch. This research is based on an automated way to verify the proper operation of the switch using a pressure sensor based method.*

Keywords*: Embedded Arduino, Force Gauge, Mechanical, Pneumatic, Seat Belt.*

1. INTRODUCTION

Most of the automobile manufacturing companies do not manufacture seat belt harness and they often outsource the manufacturing of seat belt switches to other smaller companies. But these smaller companies do not have the same sophisticated automated machinery as in automobile manufacturing. In seat belt harness manufacturing companies, the assembling processes is usually done by hand and pneumatic tips using human assembly operators. Therefore, these companies have to train the operators to check the proper operation of the seat belt switches by hand without the help of any machinery. This process requires lot of time, experience and it cannot guarantee the accurate verification of all the seat belt switches. Therefore, it is advantages to come up with an automated way to validate the proper operation of the slider seat belt switches.

It is better to understand how the slider seat belt switch works before going detail in to the proposed system. The following Fig. 1-1 is a plastic slider with a PCB assembled. This is the mostly used slider seat belt switch in the automobile industry. The slider is the white colored plastic component while the PCB is the yellowish colored circuit. In between the slider and the PCB, there is a small copper contact in the plastic slider to connect with the assembled PCB. The friction between the plastic slider and the PCB is one of the key component that determines the proper operation of the switch. Therefore, it is used as the parameter to verify the working condition of the seat belt switch. If this switch connects with two wires and a coupler, it becomes the seat belt harness. This proposed machine is to use before soldering the slider with PCB to the wires.

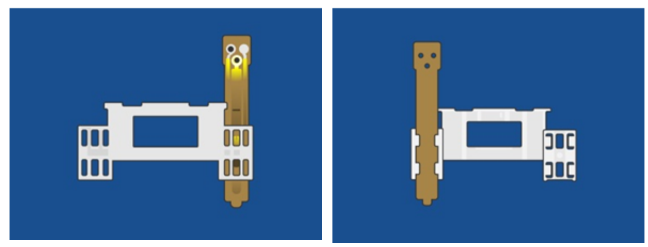

Fig. 1-1: Slider seat belt switch

2. PROPOSED SYSTEM

Most the manufacturing companies use a production line to carry out their operation. Using production line reduces processing time of workpieces, saves costs for temporary storage and controls the entire course of production. In a production line, speed of a single operation severely affects the total production of the company. Therefore, it is beneficial to design a machine which has the ability to confirm the proper operation of two switches at a single time. As a result, it will decrease the verification time for a unit of switches by factor of two.

The purpose of designing the proposed system is to sort out acceptable and no good switches with excessive accuracy in a lesser time. In addition, the proposed system has the ability to be operated by an inexperienced personal without having to be trained before by using the system. Therefore, to design a system which meets all the requirements, a set of samples from proper and no good switches are required to identify their characteristics. Since the experienced assembly operators are trained to identify the adequate friction of the two components, two set of samples of proper and no good switches can be obtained from them. Then the proposed system can be designed by researching on these samples using the notion of checking the friction force between the plastic slider and the assembled PCB to verify the proper operation of the switch.

The basic process of the proposed machine can be summarized as follows,

- Place the two components in the machine and press the 'TEST' button.
- Acquire the force readings of the two switches from the force gauges.
- Process the readings.
- Confirm the operation of the switches.

3. APPROACH & METHODOLOGY

The system is designed according to the required specifications as shown in Fig. 3-1. It contains a moving mechanical section and an electronic section. A moving mechanical section is required in the machine, since the force must measure when the PCB is moving. Which means the system must measure the kinetic force, not the static force.

The Control Unit in the system acquires the user inputs from the Input Interface, the force measurements from the Force Gauges, process the information and output the results in Output Interface.

3.1 Mechanical Section of the Machine

The moving mechanical section of the system is made using pneumatic components. The pneumatic system consists of a 1-inch air piston, 5/2 solenoid vale, air speed controller, small size FRL unit and 4mm tubes. Additionally, the PCB jig (The die of the switches) is mounted on a Linear Slider Pack. After creating the mechanical system according to the above diagram in Fig. 3-1, the whole system including the electronic section and the pneumatic system was placed on a 10mm thick plastic sheet and different sizes of plastic sheets were used for leveling and adjustments.

In the production line, the assemble unit assembles the plastic slider with the PCB then place them in a plastic tray with a particular position (Sliding position) and pass them to the testing unit. Therefore, the PCB jig was created using the same die at those trays, so the operator can easily place the switches on the PCB jig by getting the parts from the plastic tray without concerning about the sliding position of the switch. When a switch is place on the PCB jig, it holds the plastic slider completely still while allowing PCB component to slide freely.

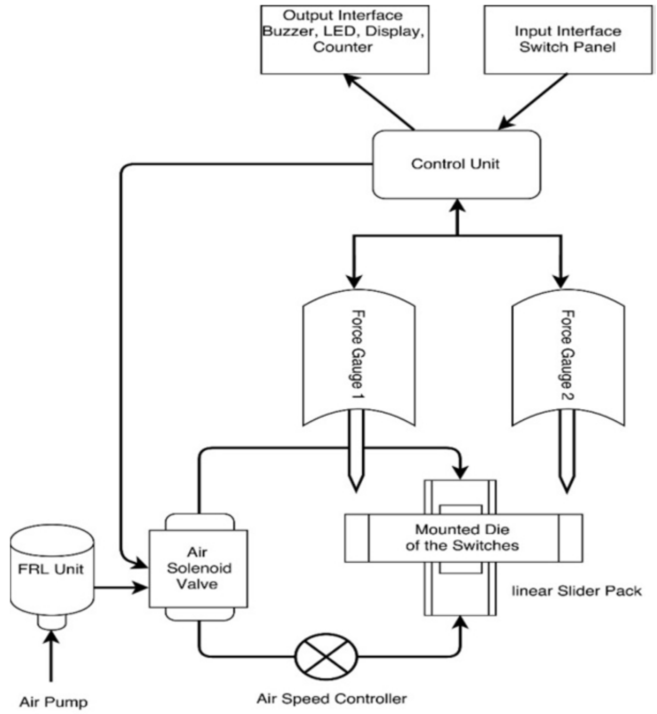

Fig. 3-1: Block diagram of the Defect Detector system

3.2 Force Gauge Analyzation

Force gauge is the most significant component of the Defect Detector machine. It is used in the proposed system to acquire the friction force between the plastic slider and the PCB. One of the main problem when working with a force gauge is its detection value changes rapidly according to the force it reads. Which will make it hard to detect a steady force value with a good accuracy. Therefore, the force gauge in the machine should have very good accuracy and sampling rate as it is used as the only sensor in the system to detect the friction force.

The machine is designed using two IMADA ZTS-5N force gauges for detecting the friction force. This force gauge is specially developed for the industrial applications and most suitable for the proposed system because of its variety of options including high sampling rate, good sensitivity, connection to PC via USB and user-friendly configuration menu. The following Fig. 3.2-1 shows the IMADA ZTS-5N force gauge. [1]

Fig. 3.2-1: IMADA ZTS-5N force gauge

This IMADA ZTS-5N force gauge can be conFig. d to measure and indicate a given range of forces as good forces (OK), the forces that are higher than the range as Higher No Goods (+NG) and lower forces are Lower No Goods (-NG). By adjusting theses "High" and "Low" limits, the force gauge is able to detect the –NG, +NG and OK. The following Fig. 3.2-2 shows the –NG, OK and +NG detections respectively.

Fig. 3.2-2: Force gauge analyzing results

One of the most important step of designing the system is identifying the force ranges for proper and no good switches. Therefore, the previously acquired samples of working and no good switches can be used to identify the High and Low limits. Even though the created mechanical section is designed to be handled by the control unit, it can also be used here by giving external power supply directly to the solenoid valve.

Another crucial part of the system is adjusting the data grabbing time of the machine. That can be challenging since the force value changes rapidly and the data must be grabbed when the PCB jig is at moving state. Therefore, the speed of the pneumatic system has to be adjusted to move the PCB jig in some applicable speed, which enables force gauge to correctly acquire the force data. If the jig moves too fast, the force value cannot be well detected.

IMADA ZTS-5N force gauge package comes with a computer software called ZT Logger. This software can be used to display the force gauge readings in the computer after connecting the force gauge to the computer. These readings are very essential for identifying the required High and Low limits of the force gauge as well as adjusting the data grabbing time of the machine. Since the force gauge is conFig. d to read 10 readings per second, the appropriate speed of the system can be found experimenting with air pressure of pneumatic cylinders while observing the force readings with ZT Logger software.

Then the required High and Low limits of the force gauge can be found using the acquired working samples and not working samples with the help of ZT logger software. The data obtained from ZT logger for one -NG, +NG and OK sample is shown respectively in below Fig. 3.2-3.

Number	Force	Displacement	Go / No GO	Time
1	-11mN	0.00mm	-NG	1:21:35 PM
2	9mN	0.00mm	-NG	1:21:35 PM
3	-6mN	0.00mm	-NG	1:21:35 PM
4	-28mN	0.00mm	-NG	1:21:35 PM
5	10mN	0.00mm	-NG	1:21:35 PM
6	8mN	0.00mm	-NG	1:21:35 PM
7	110mN	0.00mm	-NG	1:21:35 PM
8	104mN	0.00mm	-NG	1:21:35 PM
9	8mN	0.00mm	-NG	1:21:35 PM
10	1mN	0.00mm	-NG	1:21:36 PM
11	0mN	0.00mm	-NG	1:21:36 PM
12	-1mN	0.00mm	-NG	1:21:36 PM
13	0mN	0.00mm	-NG	1:21:36 PM

(a) -NG

Number	Force	Displacement	Go / No GO	Time
1	36mN	0.00mm	-NG	3:05:12 PM
2	-11mN	0.00mm	-NG	3:05:12 PM
3	3mN	0.00mm	-NG	3:05:12 PM
4	39mN	0.00mm	-NG	3:05:12 PM
5	10mN	0.00mm	-NG	3:05:12 PM
6	1156mN	0.00mm	+NG	3:05:12 PM
7	1755mN	0.00mm	+NG	3:05:13 PM
8	1545mN	0.00mm	+NG	3:05:13 PM
9	580mN	0.00mm	OK	3:05:13 PM
10	621mN	0.00mm	OK	3:05:13 PM
11	597mN	0.00mm	OK	3:05:13 PM
12	596mN	0.00mm	OK	3:05:13 PM
13	585mN	0.00mm	OK	3:05:13 PM

(b) +NG

Number	Force	Displacement	Go / No GO	Time
1	32mN	0.00mm	-NG	2:42:19 PM
2	-12mN	0.00mm	-NG	2:42:19 PM
3	-3mN	0.00mm	-NG	2:42:19 PM
4	-39mN	0.00mm	-NG	2:42:19 PM
5	-10mN	0.00mm	-NG	2:42:19 PM
6	-1mN	0.00mm	-NG	2:42:20 PM
7	518mN	0.00mm	OK	2:42:20 PM
8	427mN	0.00mm	OK	2:42:20 PM
9	448mN	0.00mm	OK	2:42:20 PM
10	6mN	0.00mm	-NG	2:42:20 PM
11	1mN	0.00mm	-NG	2:42:20 PM
12	1mN	0.00mm	-NG	2:42:20 PM
13	-1mN	0.00mm	-NG	2:42:20 PM

(C) OK

Fig. 3.2-3: Analyzed force gauge readings using ZT Logger software

3.3 Identification and Verification of the Data Cable Wires

Finally, the force gauges can be connected to the control unit after calibrating them with learned High and Low limits. The IMADA ZTS-5N force gauge consist of a mini USB and a data port. The computer can be connected to the force gauge via the mini USB and the external control unit can get force readings and other information about the force gauge via the IMADA CB-908 data cable [2].

But the system is designed to use the force gauge as a sensor to the control unit. Therefore, IMADA CB-908 data cable cannot directly connect to the microcontroller in the control unit. Hence the data cable has to be cut open to get required data for the microcontroller. Apart from the force gauge readings (-NG, OK & +NG) the cable transmits 'OVERLOAD' to indicate overload, 'READY' for device is ready, 'Zero Reset' for resetting force gauge value to zero. These parameters are given high priority when programing the control unit due to their importance in successful operation of the system.

Furthermore, it is essential to verify the data acquisition form IMADA CB-908 data cable before programming the control unit. Therefore, a small prototype can be made according to the specifications of the data cable using a breadboard and connecting it to an Arduino board. Then a small Arduino code can be written to verify and identify the wires of -NG, OK, +NG, READY, OVERLOAD and 'Zero Reset' in the cable by running the program while observing the readings via the Serial Monitor of Arduino IDE [4].

3.4 Input/ Output (User) Interface

One of the main requirement of the proposed system is have a good user interface so that the machine operator doesn't have to be a veteran and even a beginner is able to operate it without facing any problems. Therefore, the user interface was created to easily insert the inputs and observe outputs. The user interface of the system was prepared as a panel board as shown in the bellow Fig. 3.4-1.

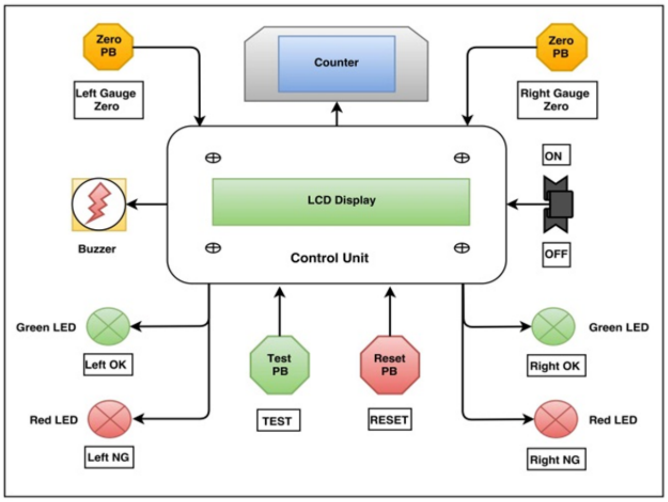

Fig. 3.4-1: User interface of the system

The user interface panel board contains TEST, RESET and Gauge Zero push buttons as inputs to the system while buzzer, counter, LCD display, red and green LEDs as outputs from the system. Every component apart from the buzzer, LCD display and counter is labeled to ease the comprehension of the machine. In addition, the LCD display is mounted on the control unit box since it is a shield of the microcontroller used on the system. The operations of each input and output components of the system are further discussed under the programming of the control unit.

3.5 Control Unit of the System

The control unit of the system consists of two Arduino Mega 2560 boards and a designed shield circuitry. Using an Arduino streamlines the amount of hardware and software development which required in order to get the system running. Furthermore, the Arduino hardware platform already equipped with the power and reset circuitry setup as well as the necessary circuitry to program and communicate with the microcontroller via USB. In addition, the Input/output pins of the microcontroller are already connected to sockets/headers for easy access [3]. The block diagram of the control unit is shown in below Fig. 3.5-1.

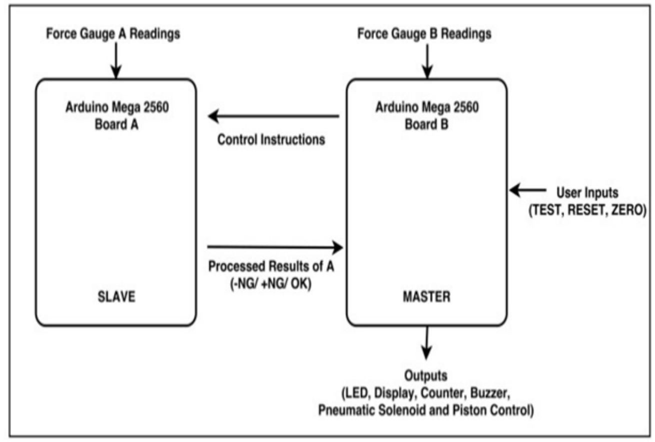

Fig. 3.5-1: Block diagram of the control unit

According to the diagram, the force gauge A is connected to board A and force gauge B to board B. The requirement of two microprocessors is to acquire and process both force gauge readings at the same time. Here the two boards are programmed to communicate with one another in a Master/Slave configuration via the I2C protocol [5]. The Board B acts as the Master (Main Unit) while the board A as the Slave.

In the control unit, board B handles all the user inputs, user outputs and the solenoid valve control. The required control instructions are also sends form board B to board A and the processed results of force gauge A is send back to Board B. Therefore, both of the boards have to be programmed separately. Additionally, a relay module is used to isolate the sensitive processing circuitry from the electromagnetic repulses of the solenoid when sending digital signals to control the solenoid valve.

In addition, couple of small circuits consist of resisters and wires are usually required to connect inputs and outputs to the control unit. To avoid this complication, a shield for Arduino Mega was designed to reduce the space that circuits required and to reduce the number of wires which has to be used. Therefore, the LCD screen, LED outputs, buzzer, and switch inputs were combined in to a shield circuit for Arduino Mega board. After soldering the components to the Mega shield, it was connected to the main Arduino Mega board and the LCD shield was connected on top of the Mega shield. Then the whole control unit was installed in a plastic box as shown in above Fig. 3.4-1.

3.6 Programming of the Control Unit

The programming of the control unit is a vital part of the Defect Detector Machine to process and control the machine accurately. From acquiring data using the input interfaces, force gauges, managing timing of the machine, controlling the pneumatic components, processing all the information to displaying the results in the output interfaces are done by the program implemented on the control unit. Therefore, the programming of the control unit has to be done precisely with utmost care to obtain accurate results from the Defect Detect Machine.

The programming was done using Arduino Software IDE (Integrated Development Environment) as Arduino Mega boards were used in the control unit. The Arduino IDE is an open source programming language and it connects to the Arduino hardware to upload programs and communicate with them. The programming of the Defect Detector machine was done according to the elaborated flow chart shown below in Fig. 3.6-1.

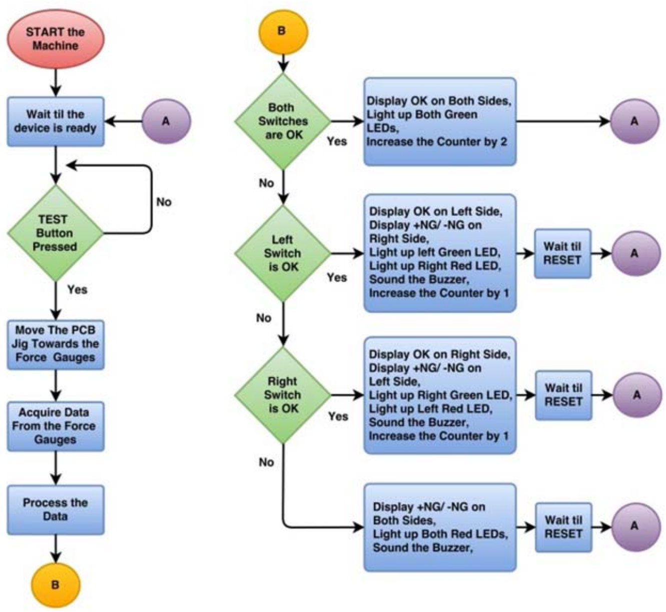

Fig. 3.6-1: Programming steps of the proposed System

The human operator has to switch on the power to start the Defect Detector machine. Once powering on, the display of the machine should indicate whether machine is 'READY' or not. This is programmed by processing the incoming 'READY' data signal from the force gauges to the control unit. And it only displays 'READY' if both of the force gauges are in the READY mode. If the force gauge reading shows a value at the beginning which is not zero, the operator can make it zero by pressing 'Zero Reset' button in the input interface. When this button is pressed, the control unit sends required signal via the data cable to the force gauge for resetting the value of gauge to zero.

The next step after placing the two switches in the PCB jig is to press the "TEST" button. Then the PCB jig should move towards the force gauge, collide with the force gauge sensor and return back to the starting point. This is achieved by programing the microcontroller to control the solenoid valve and make the PCB jig moves forward and backwards in the liner slider pack by depending on the previously found timing requirement.

At the collision, the control unit acquires the necessary readings via the data cable and process them using detection algorithms. If both parts are OK, the left and right green colored LEDs will light up, increases the counter by two and shows the word "OK" in both corners of the LCD display.

If one part is -NG while the other is OK, the red LED will light up in the respective side while green LED in the other, sounds the Buzzer, increases the counter by one and shows the word "- NG" in the respective side and "OK" in the other side on the LCD display. Same process is programed if one part is + NG and the other is OK.

If both parts are any NG type, the red LEDs in both side will light up, sounds the Buzzer and shows the respective NG type in the both corners of the LCD display. Both parts have to be OK for testing process to continue while having any NG type switches requires pressing 'RESET' button on the input interface for continuing the testing process.

3.7 Detection Algorithms

After acquiring the data from the force gauges, the microcontroller is programmed to run the detection algorithms to identify the status of the switches. For that, the detection algorithms for -NG, +NG and OK are derived using the previously obtained data from the ZT logger software for working and no good sample set. Results obtained for one -NG, +NG and OK sample is shown in previous Fig. 3.2-3.

Even though the force gauge provides many force readings, only the readings from 5th to 10th are crucial for the detection process. Therefore, the microcontroller is programmed to consider only six readings start from 5^{th} to 10^{th} to run through the detection algorithms. The considered readings are also highlighted in the previous Fig. 3.2-3.

3.8 Detecting -NG

If all the readings from 5th to 10th are -NG, the component is a -NG switch.

(5th Reading = -NG) AND (6th Reading = -NG) AND (7th Reading = -NG) AND (8th Reading = -NG) AND (9th Reading = -NG) AND (10th Reading = -NG) → -NG detection.

Detecting +NG

If any reading from 5th to 10th are +NG, the component is a +NG switch.

(5th Reading = +NG) OR (6th Reading = +NG) OR (7th Reading = +NG) OR (8th Reading = +NG) OR (9th Reading = +NG) OR (10th Reading = +NG) → +NG detection.

Detecting OK

If any reading from 5th to 10th are OK and none of them are +NG, the component is a OK switch.

[(5th Reading = OK) OR (6th Reading = OK) OR (7th Reading = OK) OR (8th Reading = OK) OR (9th Reading = OK) OR (10th Reading = OK)] AND (NOT) (5th Reading = +NG) OR (6th Reading = +NG) OR (7th Reading = +NG) OR (8th Reading = +NG) OR (9th Reading = +NG) OR (10th Reading = +NG)] → OK detection.

4. RESULTS & DISCUSSION

This Defect Detector machine is capable of detecting the following NGs.

1. The switches with Slider force bellow 0.28N (-NG)
2. The switches with Slider force above 0.81N (+NG)

The switches between 0.28N to 0.81N are considered as "OK" switches. Therefore, the machine was made according to these requirements. After the completion of the machine, a small experiment was done to find out errors and other potentials like speed, accuracy of the Defect Detector machine using an experienced assembly operator and an amateur. They were both given 10 sample units consist of 50 switches in each unit and veteran is to confirm the operation by hand while the amateur is to use the Defect Detector Machine. The results of the experiment are shown below in Table 1.

Table 1: Experiment results

Experienced Human Operator				
Unit	QTY	NG	OK	Time
1	50	-	50	2.46
2	50	-	50	2.45
3	50	4	46	2.42
4	50	-	50	2.44
5	50	3	47	2.44
6	50	-	50	2.41
7	50	3	47	2.42
8	50	1	49	2.38
9	50	-	50	2.41
10	50	-	50	2.43

Defect Detector Machine						
Unit	QTY	Press:	-NG	+NG	OK	Time
1	50	0.5	-	-	50	2.58
2	50	0.5	-	-	50	2.55
3	50	0.5	3	1	46	2.51
4	50	0.5	-	-	50	2.48
5	50	0.5	3	-	47	2.55
6	50	0.5	-	-	50	2.46
7	50	0.5	3	-	47	2.42
8	50	0.5	1	-	49	2.41
9	50	0.5	-	-	50	2.36
10	50	0.5	-	-	50	2.35

According the obtained results, it is evident that machine was able to achieve the same results as the human operator since the number of working and defective switches are same for both human operator and the machine. Additionally, the air pressure is kept at the 0.5 MPa in every session. Since the Defect Detector machine severely depends on the timing of the pneumatic unit, change in air pressure other than the 0.5 MPa can results in poor behavior.

The human operator spent total time of 27 minutes for verifying all the units with an average time of 2 minutes 43 seconds per single unit. The Defect Detector machine operator spent 28 minutes for the complete unit batch with an average time of 2 minutes 47 seconds per unit. The time per unit was calculated using the below equation,

$$\text{Time per Unit} = \frac{\text{time1} + \text{time2} + \text{time 3} \ldots \text{time10}}{10}$$

The Number of Units vs the Time graph for both human operator and the machine is shown in below Fig. 4-1. Here the average time of machine which spent to verify the complete batch is little bit higher than that of human operator. But the time spent on a single unit has gradually decreased along the verification process of the machine and the last few readings are greater than that of the human operator. This is due to the machine operator was not used to the machine at the beginning, but got used to operate the machine properly with time. In addition, the machine has checked 1220 parts per hour. But this checking speed can also be changed according to the machine operator. Therefore, the designed Defect Detector machine can verify the proper operation of a single unit faster than a human operator.

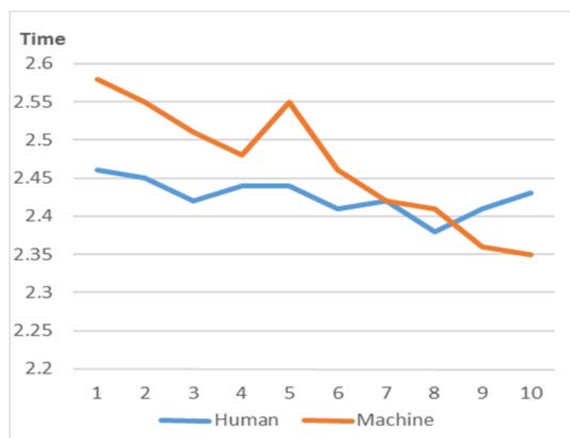

Fig. 4-1: Number of unit vs Human and Machine time graph

5. CONCLUSION

Hence the proposed system is most suitable for the production line of seat belt harness manufacturing process, since the machine operator doesn't have to be an experienced personal, outputs of the machine is very accurate and it reduces the usual verification time of the process. Therefore, the defect detector machine encourages the production to be wilder and labor costs to be less on a per product basis than the equivalent manual operations.

REFERENCES

[1] IMADA, "ZT Digital Force Gauge," [Online]. Available: http://imada.com/products/zt-digital-force-gauge/. [Accessed 2016].

[2] IMADA, "IMADA CB-908 data cable," [Online]. Available: http://www.forcegauge.net/en/cable-en/14249.html. [Accessed 2015].

[3] Arduino, "What is Arduino?," [Online]. Available: https://www.arduino.cc/en/Guide/Introduction. [Accessed 2016].

[4] Arduino, "Serial Monitor," [Online]. Available: https://www.arduino.cc/en/Reference/Serial. [Accessed 2016].

[5] Arduino, "Master Writer/Slave Receiver," [Online]. Available: https://www.arduino.cc/en/Tutorial/MasterWriter. [Accessed 2016].