Int J Adv Manuf Technol (2010) 51:233–246
DOI 10.1007/s00170-010-2626-2

ORIGINAL ARTICLE

Development of an automated testing system for vehicle infotainment system

Yingping Huang · Ross McMurran · Mark Amor-Segan · Gunwant Dhadyalla · R. Peter Jones · Peter Bennett · Alexandros Mouzakitis · Jan Kieloch

Received: 18 December 2009 / Accepted: 12 March 2010 / Published online: 15 April 2010

Abstract A current premium vehicle is implemented with a variety of information, entertainment, and communication functions, which are generally referred as an infotainment system. During vehicle development, testing of the infotainment system at an overall level is conventionally carried out manually by an expert who can observe at a customer level. This approach has significant limitations with regard to test coverage and effectiveness due to the complexity of the system functions and human's capability. Hence, it is highly demanded by car manufacturers for an automated infotainment testing system, which replicates a human expert encompassing relevant sensory modalities relating to control (i.e., touch) and observation (i.e., sight and sound) of the system under test. This paper describes the design, development, and evaluation of such a system that consists of simulation of vehicle network, vision-based inspection, automated navigation of features, random cranking waveform generation, sound detection, and test automation. The system developed is able to: stimulate a vehicle system across a wide variety of initialisation conditions, exercise each function, check for system responses, and record failure situations for post-testing analysis.

Y. Huang (✉) · R. McMurran · M. Amor-Segan · G. Dhadyalla
Warwick Manufacturing Group, University of Warwick,
Coventry CV4 7AL, UK
e-mail: yingping.huang@warwick.ac.uk

R. P. Jones
School of Engineering and IARC, University of Warwick,
Coventry, UK

P. Bennett · A. Mouzakitis · J. Kieloch
Jaguar Land Rover, Engineering Centre,
Coventry, UK

Keywords Automatic testing · Infotainment · Image processing · Modeling and simulation · Hardware-in-the-loop · Robustness · Validation

1 Introduction

An infotainment system provides a variety of information, entertainment, and communication functions to a vehicle's driver and passengers. Typical functions are route guidance, audio entertainment such as radio and CD playback, video entertainment such as TV and interface to mobile phones, as well as the related interface functions for the users to control the system. There has been a large growth in this area driven by rapid developments in consumer electronics and the customer expectations to have these functions in their vehicles. Examples of this are surround sound, DVD entertainment systems, iPod connectivity, digital radio and television, and voice activation.

With this growth in features there has been a corresponding increase in the technical complexity of systems. In a current premium vehicle, the infotainment system is typically implemented as a distributed system consisting of a number of modules communicating via a high speed fiber optic network such as Media Orientated Systems Transport (MOST). In this implementation the infotainment system is in fact a System of Systems (SOS) with individual systems having autonomy to achieve their function, but sharing resources such as the Human–Machine Interface (HMI), speakers, and communication channel [1]. Typical issues with such SOS are emergent behavior as systems interact in an unanticipated manner particularly during some initialisation conditions where it may be possible to get delays and failures in individual systems. These may not be readily observable until the particular part of the system is exercised. During vehicle development, validation of the infotainment system is extremely important and is conventionally carried out manually by engineers who can observe at a customer level but this has limitations with regard to test coverage and effectiveness. The first limitation is the time available to do manual tests, which is constrained by the development time scale and engineer's working hours. The second is in the repeatability of the test, which is subject to human error. Hence, there is a requirement for an automated infotainment test capability, which replicates a human expert encompassing relevant sensory modalities relating to control (i.e., touch and voice) and observation (i.e., sight and sound) of the system under test. This test capability must be able to stimulate the system across a wide variety of initialisation conditions including those seen under cranking, low battery or fault conditions, exercise each function, check for system responses, and record related data, e.g., MOST bus trace, in the case of a malfunction for subsequent analysis. This paper describes the design and development of such a system as part of a UK academic and industrial collaborative project into the validation of complex systems.

In the system, a Hardware-in-the-Loop (HIL) platform supported by a model-based approach simulates the vehicle network in real time and dynamically provides various essential signals to the infotainment system under test. Since the responses of the system are majorly reflected in the display of the touch screen, a machine vision system is employed to monitor the screen for inspection of the correctness of the patterns, text, and warning lights/tell-tales. The majority of infotainment functions are accessed by the user through an integrated touch screen. In order to achieve a fully automated testing, a novel resistance simulation technique is designed to simulate the operation of the touch screen. It is known that voltage transient processes, such as engine start where an instantaneous current inrush can reach 800 A, may result in some failures on the system. To test the system robustness against low voltage transient conditions, a transient waveform generator is developed to mimic three specific transient processes. A testing automation software integrates and controls all devices to form a fully automated test process, which can be run continuously over days or even weeks. The developed testing system not only makes various testing possible, repeatable, and robust, but also greatly improves testing efficiency and eases the task of tedious validation testing.

Model-based testing of functionality of an Electronic Control Unit (ECU) using HIL has been implemented by automotive manufacturers over the last few years [2–5]. Currently, Jaguar Land Rover (JLR) has adopted the HIL technology for automated testing and validation of electronic body systems, powertrain, and chassis control systems [6, 7]. The benefits of this technology include automated testing, earlier testing before physical prototype vehicle build, ability to perform robustness and dynamic testing, and reduction of supplier software iterations. Machine vision systems have been used in many manufacturing applications such as automotive [8–10], robotic guidance [11], and tracing soldering defects [12, 13]. The author also employed machine vision technology for obstacle detection in advanced driver assistant systems [14, 15]. However, no research has been reported using a machine vision system for design validation testing. Validation testing in the design stage is very much different from testing in manufacturing. Firstly, design validation testing requires diverse test cases covering a large number of, rather than a restricted, set to prove proper design. The only way to generate the test cases when the car is in the early development phases is using model-based testing techniques, which simulate vehicle-operating conditions in real time. Secondly, design validation testing requires iterative and repeated tests for robustness evaluation, although it does not require a high volume of parts to be tested. Thirdly, design validation testing needs frequent adaptation of the testing system for different types of cars or for different development stages of the same car. One novelty of this paper is the integration of the machine vision and HIL techniques for complex design validation testing. In addition, the paper proposes a novel pseudo-random concept for generating three voltage transient waveforms, which allows the testing to mimic the random process as seen in real cases, and also enables the testing to be regenerated for further investigations. Furthermore, a common approach to mimic the operation of the touch screen by a human is by using robot arms. In this design, a crafty resistance simulation approach replaces the robot arms to achieve the goal. The approach can be completely implemented in software by using the HIL simulator, therefore eliminating the need of complicated mechanical devices such as robot arms, pneumatic/hydraulic, and solenoid actuators.

2 System configurations

The configuration of the system developed for testing the infotainment system is shown in Fig. 1. The system consists of six vital elements including the unit under test, HIL tester, machine vision (camera), operation of the touch screen, transient waveform generator, and test automation.

The infotainment system under test consists of a number of modules including the radio/CD player, amplifier (AMP), navigation system, blue tooth/telephone/USB, vehicle setup, auxiliary audio interface, and climate control functions. The HMI is based primarily on a 7″ TFT resistive touch screen with additional hard keys on an Integrated

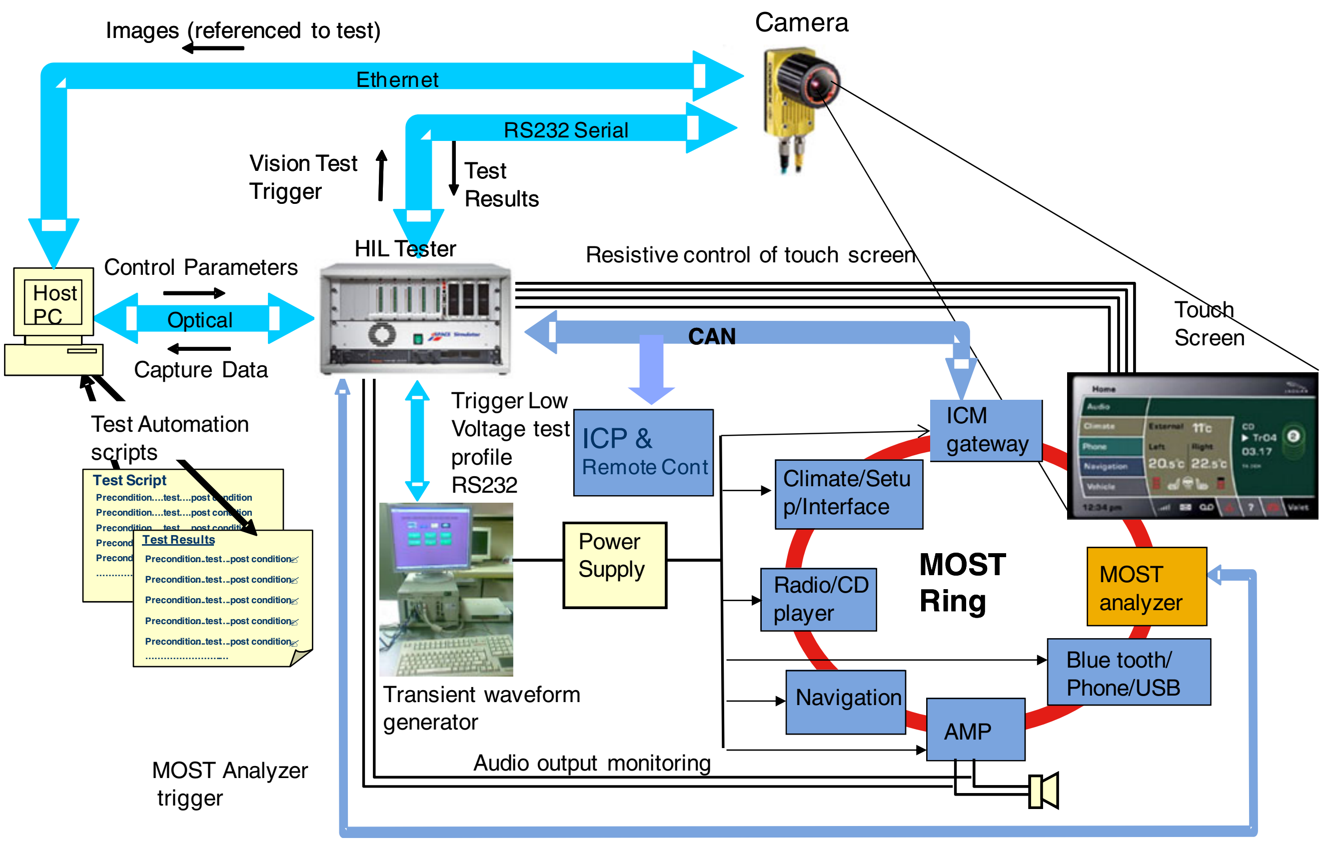


Fig. 1 System configuration

Control Panel (ICP) in the center console and remote controls on the steering wheel. Audio output is via a DSP amplifier. Communication between the modules is through a MOST optical bus carrying control, data, and audio information. The infotainment system is connected to the rest of the vehicle via a module called ICM acting as a gateway between MOST and a vehicle Controller Area Network (CAN) bus. It is worth noting that the ICP and the remote controls on the steering wheel reside in the vehicle CAN bus. In addition, a MOST analyzer was connected in the MOST ring during the testing. The MOST analyzer was controlled by the HIL tester via digital outputs to trigger the logging of the MOST traces when a failure occurs.

Within the testing system, the HIL tester simulates the vehicle network and dynamically provides various essential signals to the infotainment system under test. It also acts as a control center to control other devices. For example, it sends commands via a serial port to trigger the camera and receive the inspection results from the camera. The machine vision system (camera) checks the responses of the system by monitoring the display of the touch screen such as patterns and text. The operation of the touch screen is achieved by using a resistance simulation approach, which is implemented in the HIL tester. By using this approach, the testing system can get access to the majority of infotainment functions. The transient waveform generator produces voltage signals and powers up the infotainment system via a programmable power supplier. The waveform generator, mimicking three voltage transient processes, is used for testing system robustness against low voltage events. The test automation is running in the host computer to integrate and control all devices to form a fully automated test process. In addition, the host PC has been linked with the machine vision system via a TCP/IP Ethernet communication. This link allows the storage of time-stamped images in the host PC so that the behavior of the unit under test can be reviewed offline in terms of the test results. The following sections describe the individual elements of the automated testing system including the HIL tester, vision-based inspection, automated touch screen operation, transient waveform generator, and test experiments.

3 HIL tester

A dSPACE simulator [16] was used to form a hardware-in-the-loop simulation test system. The HIL test system simulates the vehicle CAN bus to provide power mode signals to the MOST Network via the MOST gateway. It also simulates the ICP to operate the infotainment system.

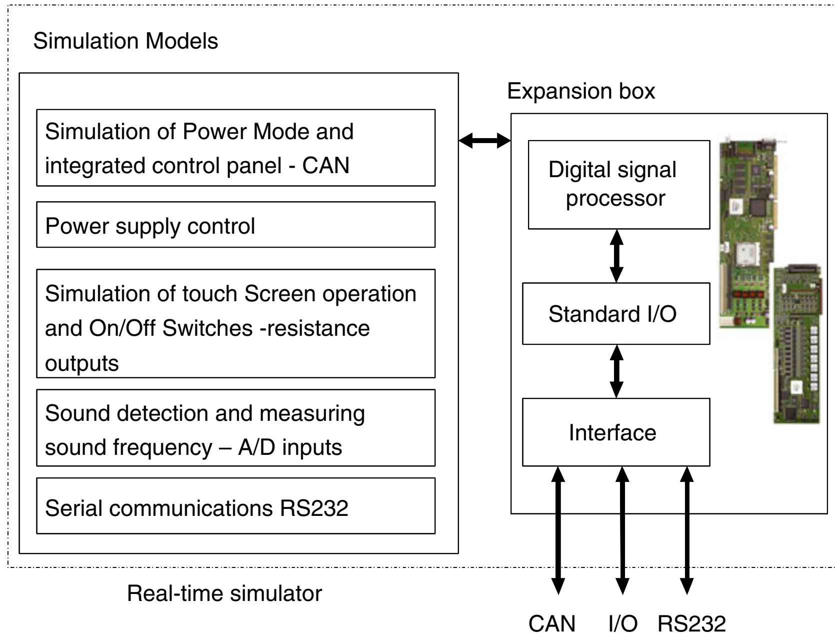


Fig. 2 dSPACE real-time simulator

In addition, the HIL tester also provides RS 232 serial interfaces to communicate with the camera and transient waveform generator, resistance simulation to operate the touch screen, and an A/D interface for detecting sound and measuring sound frequency.

The dSPACE Simulator consists of simulation models and expansion hardware as shown in Fig. 2. The expansion box includes one processor board DS1006 and one interface board DS2211. The DSP board runs the simulation models, while the interface board provides various interface links with other devices, such as CAN, resistance outputs, A/D converters, analog/digital input and output, and RS 232 serial communication to control the machine vision system.

In the HIL system, simulation models are implemented in MATLAB/Simulink/Stateflows and compiled using the auto-C-code generation functions of Matlab's Real-Time Workshop for real-time execution.

3.1 Simulation of power mode

The behavior of the components of the Infotainment system is determined by a CAN signal known as 'Power mode,' which indicates the operational state of the vehicle e.g., 'ignition off,' 'ignition on,' 'engine cranking,' 'engine running,' etc. To test the performance of the infotainment system under cranking conditions, the car under test must be in the 'engine-cranking' state when applying cranking transient voltages to the car. Moreover, any subsequent functional tests must be conducted in the 'engine-running' state after the cranking. In a real car, power mode messages are transmitted by the body ECU connected to the CAN. Since we were testing the infotainment system on a test platform representing a real car sometimes, in order to generate the correct power mode behavior, we utilized CAN simulation of the HIL tester to simulate the body ECU to transmit power mode messages to the MOST gateway.

3.2 ICP simulation

The Integrated Control Panel of the infotainment system provides users with a number of hard keys for operating the system. The functions controlled by the ICP include selection of the audio sources, loading and ejecting CDs, seeking up/down for radio stations and CD tracks, volume controls, and so on. To enable an automated testing of these functions, the ICP must be controlled by the test center, the dSPACE real-time simulator.

The ICP electronic control unit interfaces with a vehicle via the vehicle CAN. Therefore, the ICP unit was simulated by using the CAN simulation of the dSPACE simulator. The models of ICP simulation are shown in Fig. 3.

3.3 Sound detection

Sound detection contains two aspects i.e., detecting sound on or off and detecting the frequency (dominant) of the sound. The sound signal is sampled from the speaker end as shown in Fig. 1, and converted into digital signal by an A/D converter within the dSPACE simulator. The sound on/off is determined by checking the amplitude of the signal. The frequency of the sound is detected by the specific circuit of the simulator. The purpose of detecting sound frequency is to identify a sound source and active CD track. The model is shown in Fig. 4.

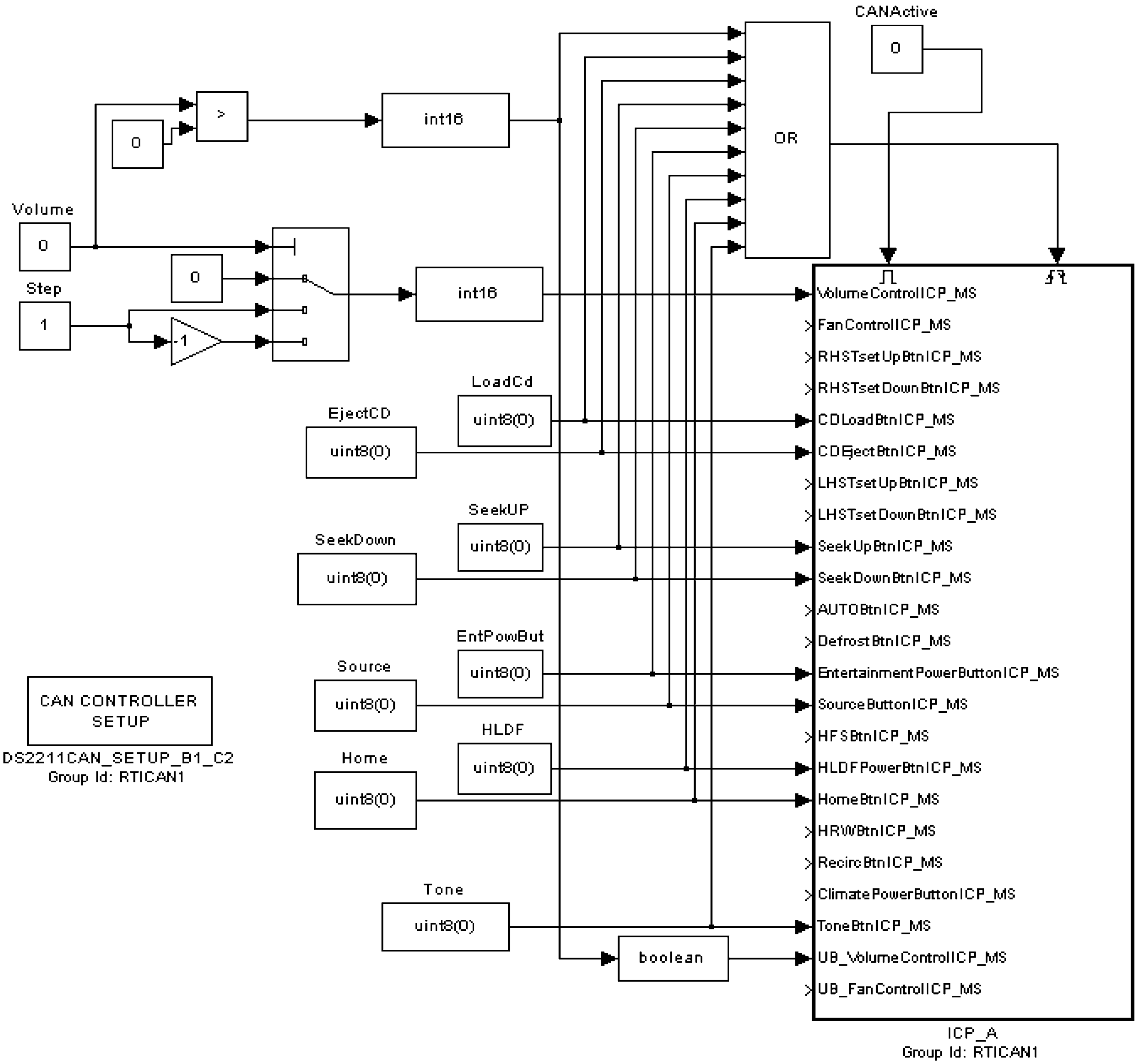


Fig. 3 Model of ICP simulation

3.4 Simulation of serial communications

The RS232 serial communication is used to establish the link between the HIL tester with the camera and the transient waveform generator so that closed loop testing can be performed. During the test, the HIL tester is the control center to command the camera and the transient waveform generator and to obtain the inspection results from them. For example, the camera needs to be commanded to select a specific image processing job file for specific testing. The checking results generated by the camera need to be returned to the HIL tester. The transient waveform generator needs to be commanded to generate a specific cranking waveform for specific testing. The parameters of the waveform resulting in a failure need to be returned to the HIL tester so that this specific testing can be duplicated in the later analysis stages.

A simplified version of the simulation models of the RS232 serial communication is shown in Fig. 5. A transmitted message is ended with a carriage return and has a maximum length of 10 bytes. A received message has a fixed length of 8 bytes. The first 3 bytes gives the result name while the following 5 bytes indicates the result values. For example, the active track number is abbreviated as the result name ATN.

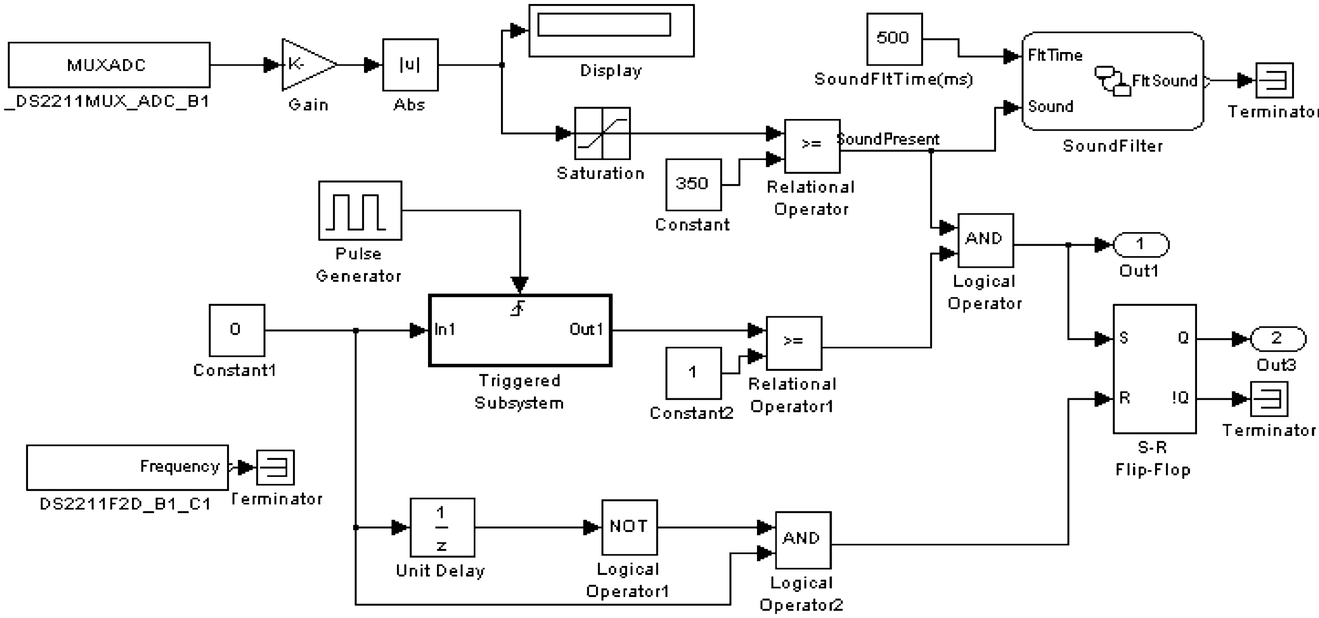


Fig. 4 Model of sound detection

4 Vision-based inspection

4.1 Machine vision system

The machine vision system consists of a camera, lighting, optics, and image processing software. A Cognex In-sight color vision sensor [17] was selected for image acquisition and processing, which offers a resolution of 640×480 pixels and a 32-MB flash memory. The acquisition rate of the vision sensor is 60 full frames per second. The image acquisition is through progressive scanning. The image processing software (In-sight Explorer Ver 4.2.0) provides a wide library of vision tools for feature identification, verification, measurement, and testing applications. The PatMaxTM technology for part fixturing and advanced Optical Character Recognition (OCR) tools for reading texts [17] are available within the software. The primary source of illumination is from a LED ring light with directional front lighting, which provides high contrast between the object and background. The selection of optical lens depends on the field of view and the working distance. In this setting, a lens with a focal length of 8 mm is used. Image processing task can be assigned into different job files stored in the camera flash memory. In this work, the image processing jobs were organized according to the system functions with five job files corresponding to five display pages. Each job file conducts all visual inspections for the page required by the test specifications. The visual inspections conducted in this work can be divided into three categories as follows.

4.2 Inspection of patterns

The majority of infotainment functions are displayed to users through the touch screen. Therefore, the majority of the functional testing is to check if the screen gives correct display as expected. For example, we need to check the page, temperature format, clock format, radio/CD sources, and so on. This checking can be done by inspecting patterns in the specific screen areas. Fig. 6 gives an example of checking the home page display. Initially, the left upper corner pattern of the home page was trained using PatMaxTM tools and stored as a master pattern. During the test, the image captured from the unit under test was compared with the trained patterns. The page was recognized based on the pattern matching score returned from the PatMaxTM tools. As shown in Fig. 6, the home page got the highest matching score of 99.9, while other pages “Audio,” “Climate,” and “Comms” got lower matching scores. The author has also applied the PatMaxTM tools for inspection of vehicle instrument cluster. More detailed information about how PatMaxTM tools work is described in [7].

4.3 Text recognition

The CD track number and some warning messages need to be checked for their correctness. This test aims at recognizing and reporting these texts.

In the image processing library, the OCR tools are designed to recognize a text string. However, this tool can

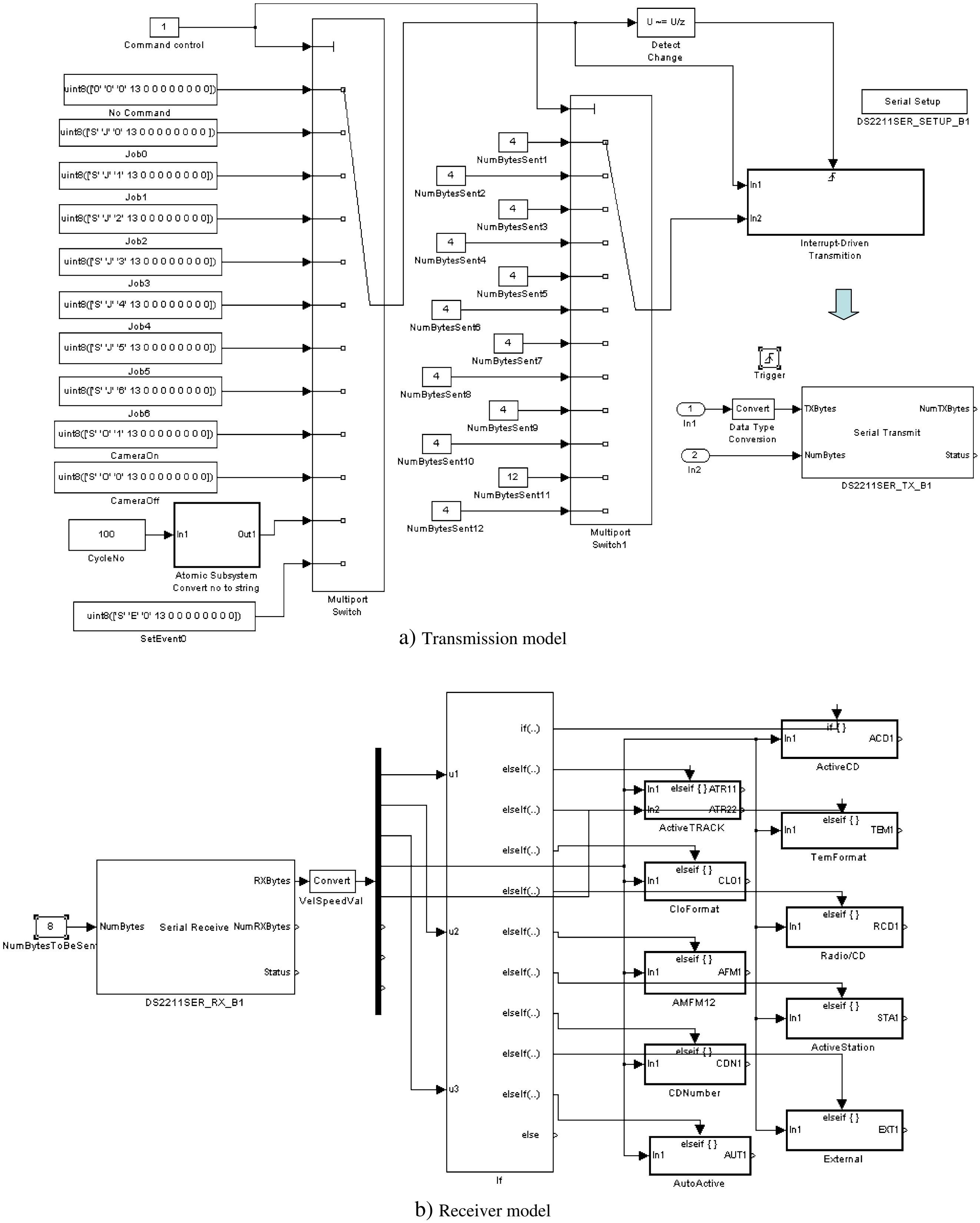


Fig. 5 Simulation models of the RS232 serial communication. **a** Transmission model, **b** Receiver model

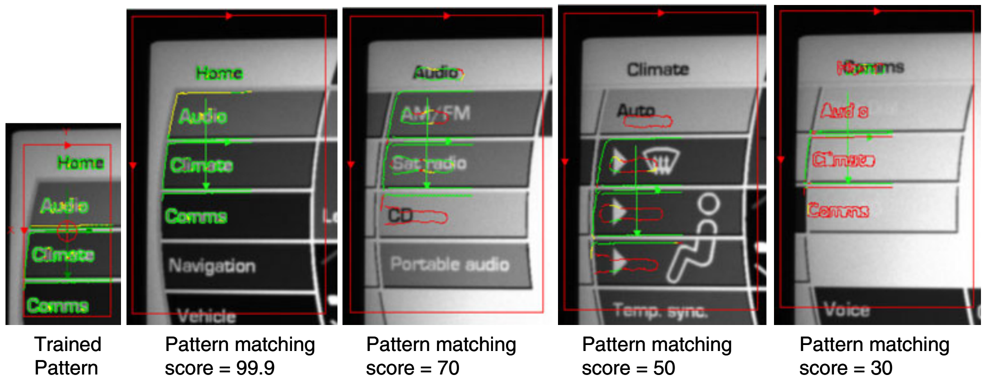


Fig. 6 Pattern inspection

only be used for identifying a text string with a known number of characters. In our case, texts appearing on the screen are distinct with different numbers of characters. In order to solve this problem, instead of identifying a whole text string, each character of the string was recognized separately by treating it as a “blob”. An example is shown in Fig. 7 for identifying a CD track number. We firstly used a blob tool to extract the blobs within the string and sort them in terms of their positions. Since characters were separated from (not linked) each other, each character was recognized as an individual blob. Sequentially, each blob area was located with a red text tool, and the character within the blob was recognized by comparing it with the trained fonts. All digits from 1 to 9 were pre-trained and stored using the “TrainFont” vision tool. Finally, all identified characters were concatenated together to retrieve the entire string.

4.4 Color identification

Since the active button on the display is shown with a different color, color identification was used to check active button, active CD, active radio stations, and so on. The “ExtractColorHistogram” vision tool was used for this purpose, which accesses every whole pixel in an image region to accumulate a color histogram. Each color pixel is composed of three separate color components: red, green, and blue (RGB). Each color component is converted to a value from 0 to 255; therefore, the accumulated histogram contains 256 elements for each color component. The summation of the average values of the three color components within the region represents a specific color. By this method, we can identify a color within a button area as shown in Fig. 8. It’s worth pointing out that the color detection based on the average value of RGB colors is simple but may cause some false detection in some cases. Examples are that the RGB values (255, 0, 0), (0, 255, 0), (0, 0, 255) and (85, 85, 85) indicate different colors but have the same average value. However, the method developed works fine in the application since the color detection is majorly for discriminating an active button, active CD, and active radio with a change from dark blue to light blue. For identifying more colors, we may use a hue, saturation, and intensity (HIS) color model.

5 Automated touch screen operation

As the main HMI of the infotainment system, the touch screen allows the user to navigate through various menus and screens to select and control a desired function, or to configure user preferences and defaults. Fig. 1 also shows a small picture of the premium vehicle touch screen. In order to achieve the goal of a fully automated infotainment test capability which can replicate a human expert, an essential requirement is the ability to replicate the user’s touch screen button presses. Approaches for simulating user button presses were investigated including the use of a robotic

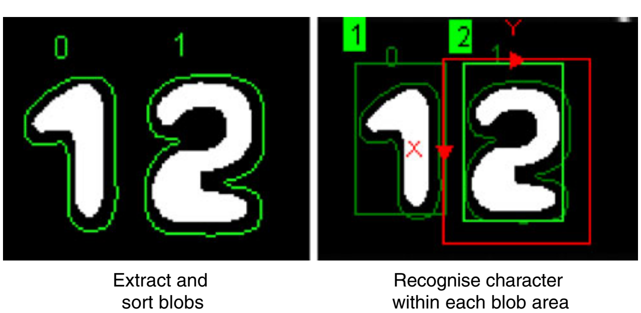


Fig. 7 Text recognition

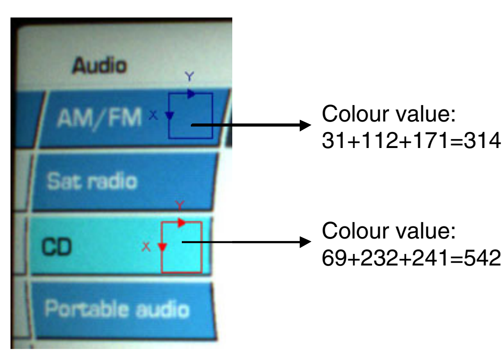


Fig. 8 Color identification

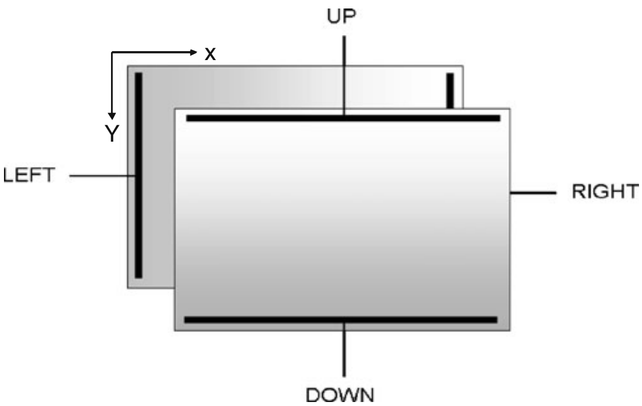


Fig. 9 Four-wire resistive touch screen

arm or finger, and an array of solenoid operated actuators. These approaches were discounted due to issues of either cost, interference with the operation of the machine vision system, or limited upgradeability. As an alternative and more suitable solution, a novel resistance simulation technique was developed for the purpose.

5.1 Touch screen operation

The touch screen consists of two layers including a rigid glass back layer and a flexible polyester front layer, each with a uniform resistive coating, separated by small insulating spacers or dots. One layer is used to capture horizontal coordinates, and the other layer vertical coordinates. Electrodes at either side of each resistive layer are connected to the touch screen controller. This arrangement is illustrated in Fig. 9. When the screen is touched, the front layer is forced against the back layer at the touch point and an electrical contact is made between the two resistive coating layers.

5.2 Measuring touch-point coordinates

The equivalent circuit for such a four-wire resistive touch screen is illustrated in Fig. 10.

The values of resistances R_{X1}, R_{X2}, R_{Y1}, and R_{Y2} will vary according to the position of the touch point. The switch and the resistor R represent the touch point and contact resistance. This means coordinates of a touch point are equivalent to a set of values of resistance, R_{X1}, R_{X2}, R_{Y1}, and R_{Y2}.

It takes two steps to obtain respectively the X and Y coordinates of the touch point. Firstly, to obtain the X coordinates, the touch screen controller applies a fixed voltage across the back layer, and then uses the front layer as a voltage probe to measure the voltage at the touch point. It then repeats this process to obtain the Y coordinates by applying a fixed voltage across the front layer, and using the back layer as a probe. This process is illustrated in Fig. 11.

The resistive coating on each layer creates a potential divider at the touch point. The voltages Vx and Vy equivalent to separated resistances are an analog representation of the X–Y coordinates of the touch point. The touch screen controller then converts these measured voltages into the appropriate screen press coordinates.

5.3 Resistive simulation

Resistance simulation was adapted to mimic a physical press of the touch screen like a human user operation. Five software controllable resistances in the HIL tester were used to simulate the circuit shown in Fig. 11. Four of them simulate resistances R_{X1}, R_{X2}, R_{Y1}, and R_{Y2} while the fifth resistance mimics a short electrical contact as generated by physical press. The simulation outputs were then connected in parallel with the real touch screen.

Calibration must be performed to build up a look-up table listing the coordinates on the touch screen and its equivalent resistances. Accordingly, any point in the touch screen can be virtually pressed by assigning the equivalent resistance values to the resistances. In practice, the touch screen was divided into many small areas, each corresponding to a set of resistance value.

5.4 Benefits of the method

The approach of mimicking screen button press offers the following benefits:

- The approach does not affect the normal operation of the touch screen itself since the simulated equivalent circuit is connected in parallel. The real touch screen can be operated alongside the simulated touch screen. This is particularly useful for experimental setup and manual operation of the Infotainment functions.
- The approach does not obstruct the view of the display. This allows the machine vision system to operate easily and without hindrance.
- This technique can easily and quickly accommodate changes in screen layout resulting from either software changes or changes in display/vehicle. New or changed button coordinates can be introduced by adding or updating the corresponding resistance value mappings.

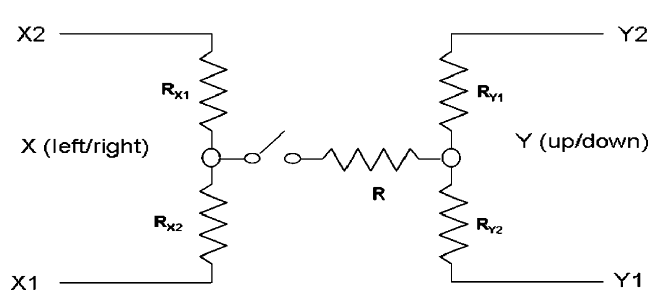


Fig. 10 Touch screen equivalent circuit

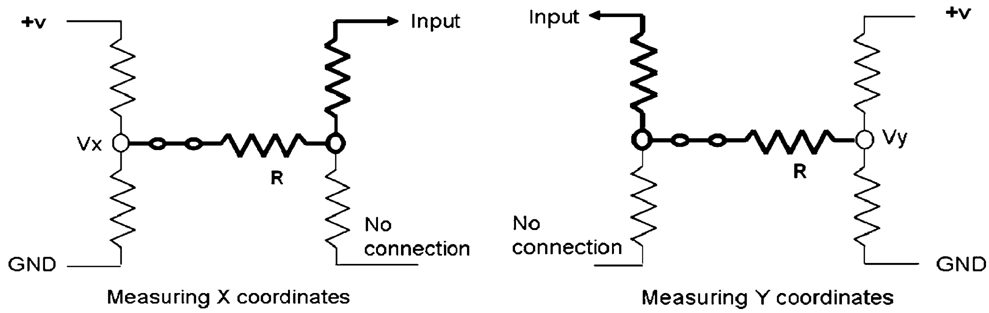


Fig. 11 Measuring touch-point coordinates

6 Transient waveform generator

6.1 Three voltage transient conditions

A transient waveform generator was developed to simulate three voltage transient conditions including Engine Crank, Contact Bounce, and Voltage Ramp (down/up), which have been specified in the JLR Engineering Standard—Immunity to Low Voltage Transients [18]. Engine Crank is the case when the car engine is cranking. A parameterized representation of the voltage seen at the battery terminals is shown in Fig. 12. The values of all parameters T1, T2, T3, T4, V1, V2, and V3 may randomly vary depending on factors such as system configuration, electrical load (impedance), battery state of charge, and ambient temperature.

Contact Bounce reflects the voltage transient caused by poor connections at the battery terminals. A poor connection can be generated by installation errors or damages to the wiring or connectors. Fig. 13 illustrates the parameterized representation of the contact bounce voltage waveform.

Voltage Ramp is the voltage variation caused by a slow battery discharge, dwell at low voltage and subsequent trickle charge. A common example of these cases is: the engine is switched off but some vehicle loads are left on such as the lights or the radio. A parameterized representation of the ramp profile is shown in Fig. 14.

As shown in the figures, each type of waveform can be represented by a set of parameters. In real situations, each parameter can randomly vary. The variation of the parameters creates various combinations representing different voltage scenarios. That is, these voltage scenarios can be mimicked by manipulating these parameters.

6.2 Pseudo-random distribution

Two types of probability distributions, uniform distribution and normal distribution, were selected to generate the parameter sets. For example, if the test was being run for the first time then the test engineer may require getting a broad understanding of where failures may occur. In this case a parameter set with Uniform distribution would be a sensible selection since it will give an equal probability to each value within the test set. Whereas, if after some testing it became evident that failures occurred around a specific value, then, the parameter sets that need to be generated would have to be biased around this point. In this case a parameter set with normal probability distribution where the mean is equal to the failure parameter value would be a sensible selection.

An important requirement for the testing system is the ability to repeat a given test sequence so that any actions taken to address an identified fault can be reconfirmed. This is extremely useful for post-review and analysis in order to isolate the failure causes. The way to fulfill this requirement was to use a pseudo-random number generator to generate parameter values. In this project, a LabVIEW software package (Ver. 8.5) [19] was employed to implement the transient wave generator. The LabVIEW software library provides a selection of pseudo-random number generator functions which support both uniform and normal probability distributions. A test set generated by the pseudo-random number generator depends on a seed value, the range and length of the number, therefore can be regenerated. The seed value can be any value within the range. In this work, test sets for each parameter were individually generated by the pseudo-random generator with the same sample size i.e., the number length. To cover as many voltage scenarios as possible, the number length should be as big as possible. In practice, it can be set according to the testing time allowed and not overrun the computer memory size.

7 System evaluation and experiments

The testing system developed has been deployed and heavily utilized in JLR. Five different carlines, i.e., Range Rover, XF and Discovery, have been tested with different versions of software updates. It has been evaluated that an automated testing using the system developed is four times faster than the manual testing and utilizes approximately 20% of the resource required for manual regression testing once test scripts have been developed. Furthermore, the testing accuracy is dramatically improved with the detected failure ratio increased from 5% by manual testing to 12% by the automated testing for the first issue of the software of a specific vehicle. Fig. 15 shows a real test setup for a

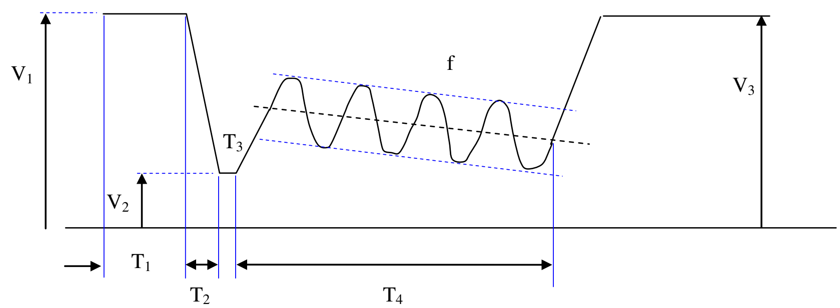


Fig. 12 Parameterized representation of a crank waveform

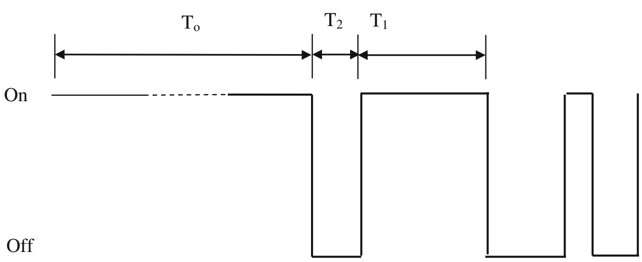


Fig. 13 Parameterized representation of the contact bounce waveform

bench test. In this setting, two cameras are employed for inspection of the instrument cluster and the touch screen display respectively. The transient waveform generator is an embedded version of the system described in Section 6. The test automation software was provided by dSPACE as a part of the package of HIL simulator and runs in the host PC, which is not contained in the picture.

Figure 16 gives a typical test plan for investigating the impact of various cranking waveforms on the infotainment system. The test script is created using dSPACE AutomationDesk (Ver. 1.5) software. The software allows a real-time control of the Simulink models in the HIL and features a drag-pull user interface. Each test cycle corresponds with applying one mode of cranking voltage to the vehicle system. Before starting a test cycle, the testing system and the Infotainment system under test need to be initialized so

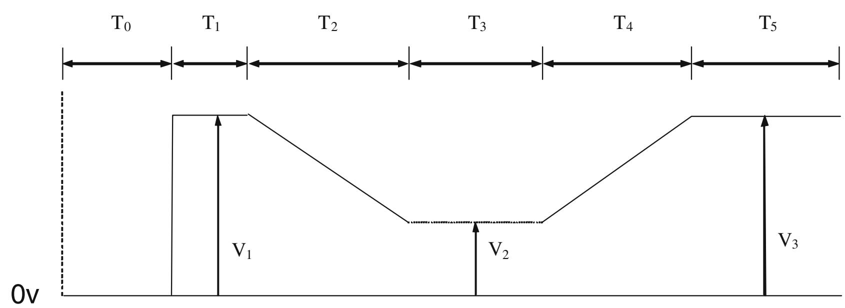


Fig. 14 Parameterised representation of the ramp waveform

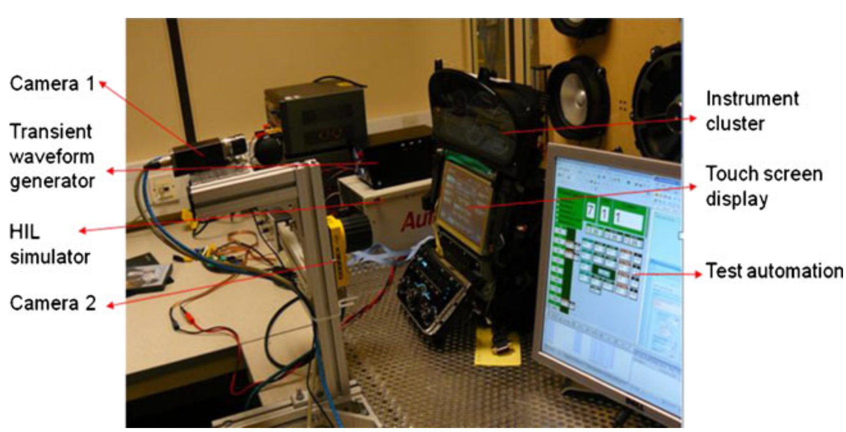


Fig. 15 A real testing setup for a bench test

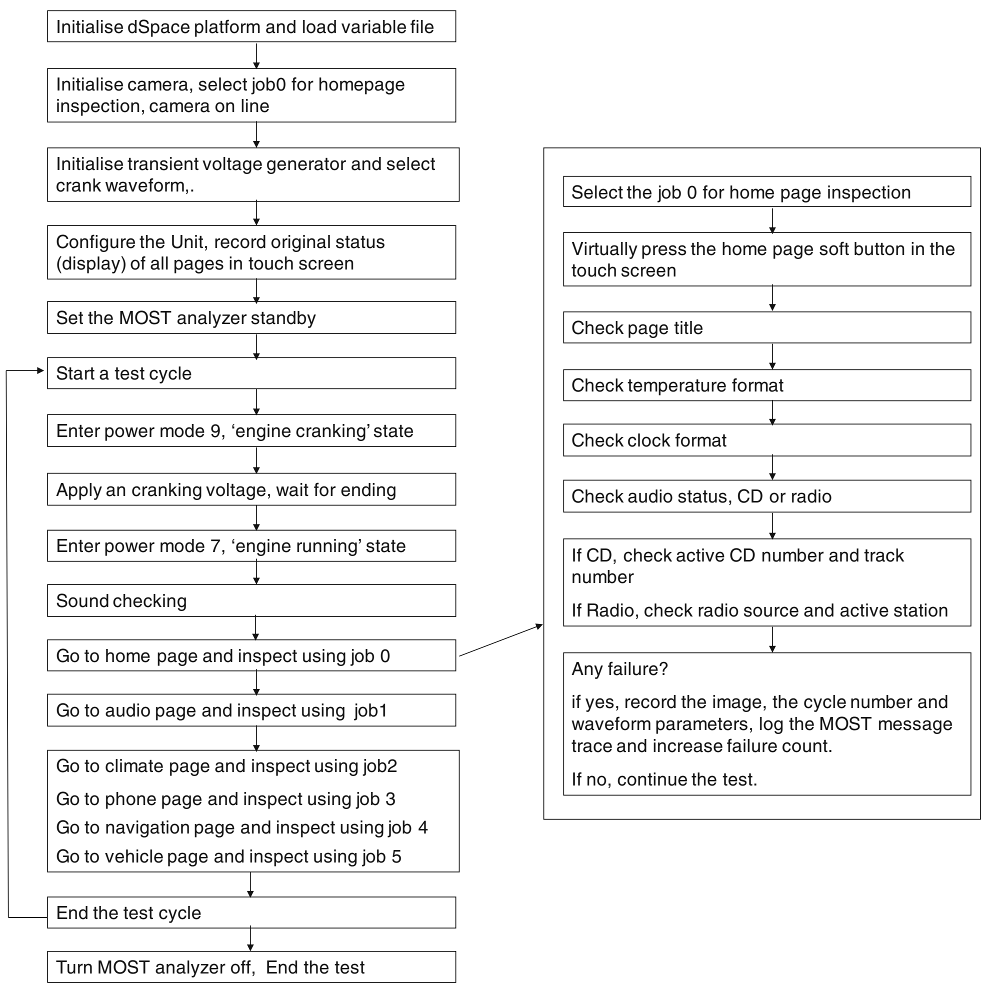
that the testing system is ready and original configuration of the touch screen can be recorded as a reference. In the beginning of a test cycle, a CAN message is issued by the simulator to make the car go to power mode 9 (cranking state) as discussed in Section 3.1. Meanwhile, a cranking voltage is created by the Transient Waveform Generator and applied into the car. The car then goes to normal running state and the testing starts, which includes sound checking and inspections of five different display pages. The right part of the flow chart gives more details on the home page inspection such as checking the page title, temperature format, clock format, and audio status. When an unanticipated pattern occurs, the testing system is trigged to record the failure scenarios including display images, MOST traces, and transient waveform parameters. The information is synchronized by test cycle number and time stamping. These features are extremely useful for post-testing analysis and further investigation. The testing described in Fig. 16 has been continuously run for 100 h with more than 5,000 test cycles generated. The experimental result has shown that the system was highly robust for a long period of the testing.

Figure 17 gives five screen shots to illustrate the operation of the testing system. Screenshot 1 shows the test automation script running in the AutomotionDesk. Screenshot 2 shows the image processing jobs running in the Cognex In-sight camera. Screenshot 3 illustrates how the touch screen are manipulated by the resistive simulation approach, where the touch screen is divided into 40×24 small square areas (grids), each corresponding to a set of resistance value. A button within the touch screen may span a number of grids. Screenshot 4 shows the system control desk where all settings and configurations on the unit under test and the testing system can be done. It can also be used to conduct some manual testing. Screenshot 5 illustrates the status of a transient waveform being applied to the unit under test.

8 Conclusions

A system capable of replicating human control and observation has been developed for automated functional and robustness testing of a vehicle's infotainment system. The system mainly contains four modules including an HIL

Fig. 16 Flow chart of a typical test plan



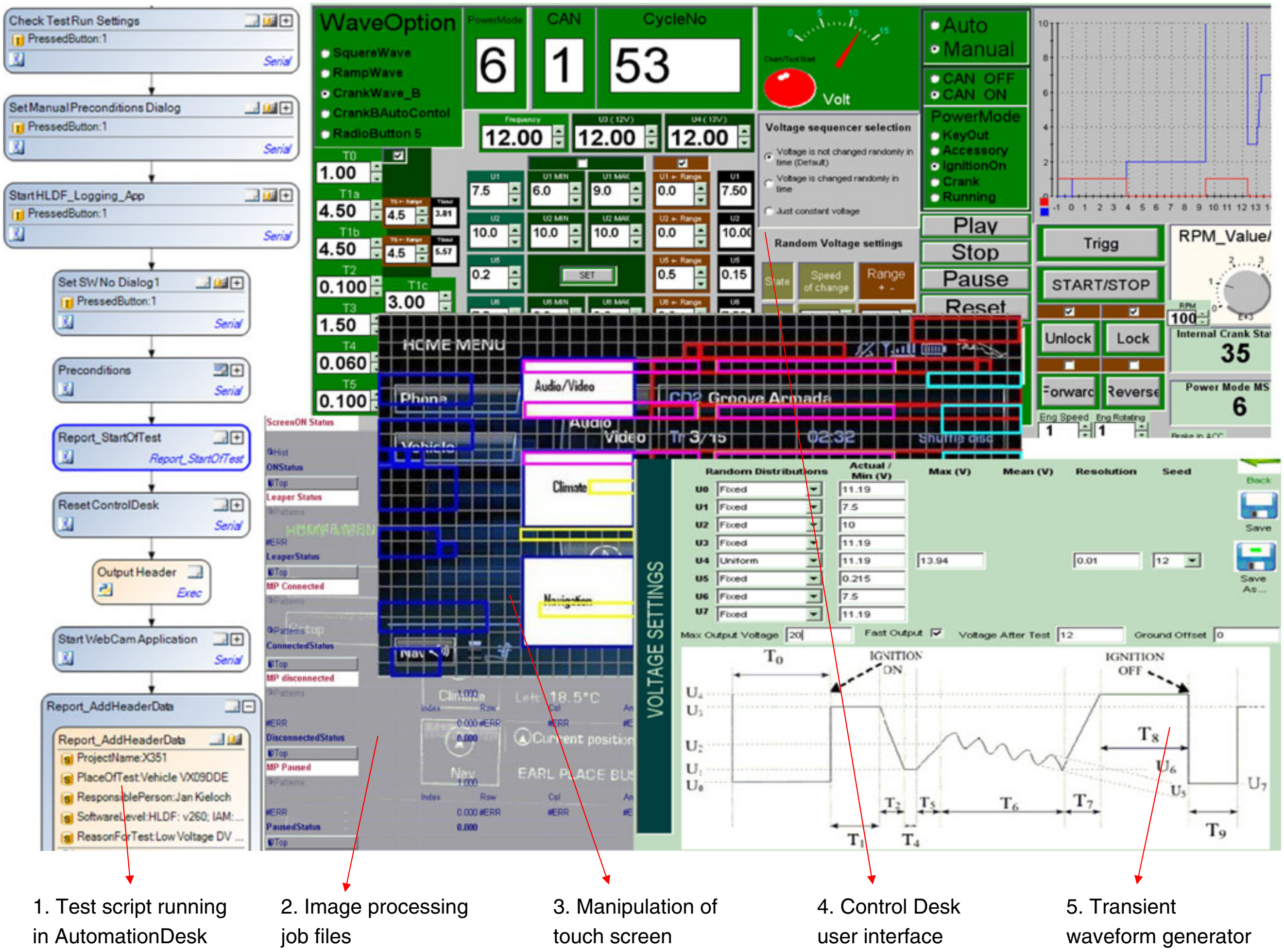


Fig. 17 Screen shot of operation of the testing system

tester, machine vision, automated touch screen operation and transient waveform generator. Serving as a test center, the HIL facility communicates with other modules via RS 232 serial interfaces. In addition, it provides simulation for the vehicle power mode, ICP operation, and the touch screen operation. Furthermore, it also provides A/D interfaces for detecting sound and measuring sound frequency. The machine vision system was designed to perform visual inspection on the display by detecting patterns, text, and color. The transient waveform generator was developed to mimic three voltage transient conditions including Engine Crank, Contact Bounce, and Voltage Ramp. The automated touch screen operation was achieved via a resistance simulation technique, which provides virtue press on the touch screen thereby facilitating the test system. The system developed greatly eases the task of tedious validation testing, increases the test repeatability and reliability, and saves testing time. Moreover, the system developed is able to create various low voltage transient conditions seen in the real world and verify the system robustness under these conditions.

References

1. McMurran R, McKinney F, Tudor N, Milam WP (2006) Dependable systems of systems, *SAE paper 06AE-34*
2. Kendall IR, Jones RP (1999) An investigation into the use of hardware-in-the-loop simulation testing for automotive electronic control systems. Control Eng Pract 7:1343–1356
3. Lee SJ, Kim YJ, Park K, Kim DJ (2003) Development of hardware-in-the-loop simulator and vehicle dynamic model for testing ABS. SAE Technical Paper Series, 2003-01-0858
4. Bertram T, Bekes F, Greul R, Hanke O, Hab J, Hilgert J, Miller M, Ottgen O, Opgen-Rhein P, Torlo M, Ward D (2003) Modelling and simulation for mechatronic design in automotive systems. Control Eng Pract 11:179–190
5. Yeo H, Kim H (2002) Hardware-in-the-loop simulation of regenerative braking for a hybrid electric vehicle. Proc Inst Mech Eng, D-J Automobile Eng 216:855–864
6. Mouzakitis A, Humphrey R, Bennett P, Burnham KJ (2006) Development, testing and validation of complex automotive systems. The 10th Mechatronics Forum Biennial International Conference, Philadelphia, USA
7. Huang Y, McMurran R, Dhadyalla G, Jones P, Mouzakitis A (2009) Model-based testing of vehicle instrument cluster for design validation using machine vision. Meas Sci Technol 20 (6):11. doi:10.1088/0957-0233/20/6/065502, ID 065502

8. Andres NS, Marimuthu RP, Eom YK, Jang BC (2005) Development of a machine vision system for automotive part inspection. Proceedings of SPIE—The International Society for Optical Engineers, ICMIT 2005: Information Systems and Signal Processing 6041: 60412J1–6
9. Hage P, Jones B (1995) Machine vision-based quality control systems for the automotive industry. Assem Autom 15(4):32–34
10. Shafi A (2004) Machine vision in automotive manufacturing. Sens Rev 24(4):337–342
11. Yang S, Cho M, Lee H, Cho T (2007) Weld line detection and process control for welding automation. Meas Sci Technol 18:819–826
12. Qixin C, Zhuang F, Nianjiong X, Lewis FL (2005) A binocular machine vision system for ball grid array package inspection. Assem Autom 25(3):217–222, Emerald Group Publishing
13. Jones CP, Ludlow JE (1995) Machine vision in the electronics industry. Assem Autom 2(2):30–35
14. Huang Y, Fu S, Thompson C (2005) Stereovision-based object segmentation for automotive applications. EURASIP J Appl Signal Process 2005(14):2322–2329, doi:10.1155/ASP.2005. 2322.
15. Huang Y, Ken Y (2008) Binocular image sequence analysis: integration of stereo disparity and optic flow for improved obstacle detection and tracking. EURASIP J Adv Signal Process 2008:10. doi:10.1155/2008/843232, Article ID 843232
16. dSPACE, http://www.dspace.com/ww/en/ltd/home.cfm
17. Cognex (2009) “Cognex Vision Toolbox Manuals”, http://www.cognex.com
18. Jaguar Land Rover (2009) Engineering standard: immunity to low voltage transients—STJLR EES_120.044, Issue II
19. National Instruments http://www.ni.com/labview/