Real-World High-Accuracy Compact Resistive Sensor Simulation Integrated into HIL Automated Validation Test Bench for Fuel Sensor Simulation
Real-World High-Accuracy Compact Resistive Sensor Simulation integrated into HIL Automated Validation Test Bench for Fuel Sensor Simulation

Vladimir Ranchev*
* Visteon Corporation, 2019
** Functional Validation CoC, Sofia, Bulgaria

Abstract
A device under test (DUT) relies on information from a number of different sensors to fulfill required functionality – these may include measurements of temperature, humidity, liquid-level and resistance, for example. A hardware-in-the-loop (HIL) automated validation testbench employed to test the DUT needs to simulate the behavior of these sensors as accurately as possible to ensure that the electronic control unit (ECU) behaves on the test bench exactly as it would when operating in the real world. This means the transition between two values shall follow more like an exponential function rather than switching between them. A standard vision automated testbench uses a digital acquisition card with limited digital output to cover all required resistor chains with accepted accuracy and there is limited space to scale the system which will get a significant impact on the price as well. Motorizing industrial-grade potentiometer and using it for process control applications is an outstanding solution for the given task and constraints.

Author Keywords
Resistive; Sensor; Simulation; Potentiometer; Industrial; Automation; Testbench; Validation; Real-World sensor;

1. Introduction
Today’s vehicles have a high demand for fuel consumption and distance-to-empty calculation. Leaving the engine without fuel may cause costly damage. For proper calculation, there is a need to measure the remaining fuel in the tank with high precision depending on road conditions such as hill climbing, rough road or a harsh driving environment. Therefore at least two sensors have mounted the tank in order to ensure proper measurement of fuel left in the tank.

Faulty measurements can be caused by any component of the potentiometer. This includes deviation of the total resistance from the nominal one (higher or lower value) or measured values from the middle point doesn’t equal the total resistance value i.e.
\[ R_T \neq R_1 + R_2 \] (1)

The software logic is capable of determining the different types of signal failures and report the most accurate fuel level quantity. Visteon’s functional validation center of competence (CoC) is required to test this logic and needs to simulate all types of errors that the sensor can produce. Therefore one sensor is simulated by three different variable resistors with high accuracy. The standard vision automated validation bench needs to support two sensors, i.e. six resistance simulations only for the fuel measurement functionality and other digital inputs and outputs (DIO).

2. Problem Statement
The standard vision automated testbench is installed in the cabinet or rack and there is a dedicated place for DUT, PC full tower, cameras and stands, connector test box (CTB) and power supplies. The CTB connects the PC equipped with DAQ card and DUT itself. One PCI/PCIe slot is utilized for the DAQ card and rest are dedicated to video frame grabbers or interface boards.

Figure 1. Fuel Tank with two sensors.

These sensors are subject to wear and changing their electrical characteristics due to the aggressive impact of the fuel liquids over time or accumulation of dirty sludge. The impact of all those factors should be reduced by the software algorithm of a DUT. The DUT is doing the measurement and it is usually part of the instrument cluster or body controller based on individual vehicle OEM architecture. In most cases, the sensor is represented as a potentiometer shown on the electrical diagram in Figure 2.

Figure 2. Potentiometer Diagram with possible faulty areas (inside Blue is shortened and outside Red is poor/dirty contacts).
The requirements set to test a DUT with the following specification for fuel sensor:

- The DUT shall support up to two 3-wire sensors
- Hardware configuration of the 3-wire sensor is as shown in Figure 3
- The DUT shall achieve accuracy of <1 Ohm of measured fuel level for sensor value within a range of 30 Ohm \( \leq R_{\text{sensor}} \leq 300 \text{ Ohm} \)
- The DUT shall achieve accuracy of <1 Ohm of measured fuel level for dirt resistor value within the range 0 Ohm \( \leq R_{\text{dirt}} \leq 30 \text{ Ohm} \).

This would have the following impact on the testbench:

- PC will not have enough PCI/PCIe slots so it needs to be replaced by PXI
- Extra PXI DAQ Cards
- Extra CTBs with relays
- Additional space to install all the extra hardware

It would have a huge impact on the cost and facility required for the automated testbenches.

3. Design of Alternative Solution

In order to upgrade more than ten HIL automated vision testbenches to support the fuel sensor simulation, a new innovative method is required. This shall implement techniques that enable more efficient usage of HIL testbench interfaces, keeping its compact size. The method shall also support a smooth transition and controlling the sensor as it would happen in real-world usage.

After research made on that topic, the motorized potentiometer (MotPot) appears to be a feasible solution. They are simple to set-up. Easy to operate. Easy to maintain. Never obsolete. There are products already available on the market with a built-in controller, but dimensions and prices wouldn’t bring expected optimization. Moreover, the Validation Framework used to drive the HIL bench, can takeover controlling the motorized potentiometers and remove the need for a costly solution.

A proposed design for the controllable motorized potentiometer is shown in Figure 6.

A single daughterboard contains two industrial-grade multiturn potentiometers, DC geared motor, two relays, and a motor driver. Potentiometers are 10-turns multiturn in order to allow precise positioning. The gear-motor shall provide 250-300ppm which means moving the potentiometers from end to end or to any intermediate position within 3 to 5 seconds (higher ppm may damage the potentiometers and lower will impact the performance of the device). Potentiometer shafts are linked together with the motor shaft so that they are always moving together without slipping. Rout is used as a simulated resistance, while Rsense is used to detect the current position of the sensor. The position is calculated as a percentage and it doesn’t depend on the stability of Vsense:

\[
P_{\%} = \frac{V_{\text{pos}}}{V_{\text{sense}}} \times 100\%
\]
Each sample gets calibrated by attaching Ohmmeter on the $R_{out}$ and gathering 50 points along the used range that gives us the relation between position in percentage [%] and output resistance value [Ohm].

![Figure 7. Connection diagram during calibration](image)

The $R_{sense}$ is purposely fixed at 90 degrees forward while $R_{out}$ is at the minimum endpoint in order to avoid noise in $V_{pos}$ measuring around 0[V] (this can be seen from the calibration graph in Figure 8. where $P$[%] is starting from 5% while $R_{out}$ is 0[Ohm]).

![Figure 8. Typical calibration curve of Integrated MotPot](image)

The potentiometers are linear, while the calibration point gets interpolated with `pchip` interpolation function in order to cover the minimal non-linear area of the potentiometers.

The motor driver is aimed to control motor speed and direction according to the three DIOs control signals in order to set the desired resistive value. Also, upon request, the motor will be turned off and an open circuit/short circuit can be simulated.

### 4. Integration into the Validation Framework and HIL Bench

As shown in Figure 6 only three DIOs are required by this design instead of 12 which is four times the optimization in terms of sensitive DIO usage. The new design allocates an additional two AIs which were not used until now. The problem that arises with this design is the code that will drive the system. It needs to be real-time analyses of the inputs and in case there is a difference between requested and current position, the corresponding signal needs to be activated in order to drive the motor that moves the potentiometer on the desired position. If the requested position gets changed, this triggers an immediate reaction of the system to start the process of adjustment to the new value.

Such applications are typically driven with control loops. Control loops are used almost everywhere nowadays. Anytime we adjust our current work according to the obtained previous results we form a control loop. For example, when we feel cold and turn our heater on, we form a feedback loop, and when we press the accelerator of a car, whenever we are late, we again form a control loop. Whenever we make any change in the environment by sensing the previous results of that process we form a closed control loop.

A PID controller is an instrument used in industrial control applications to regulate temperature, flow, pressure, speed, and other process variables. PID controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controller. A PID controller is a well-established way of driving a system towards a target position or level. It's practically ubiquitous as a means of controlling temperature and finds application in myriad chemical and scientific processes as well as automation. PID controllers use closed-loop control feedback to keep the actual output from a process as close to the target or setpoint output as possible.

PID controllers are the most widely used controllers in industrial settings because of their ease of use and the satisfaction of performance they are capable to provide the user. The cost/benefit ratio provided by these controllers is more than provided by any other controller. Many techniques have been proposed for their design, because of their widespread use, for the tuning of the parameters of PID i.e. $K_p$, $K_i$ and $K_d$ and for the implementation of additional functionalities that improve their performance.

![Figure 9. PID Control](image)

The block diagram of a simple PID controller is provided in Figure 10.

![Figure 10. PID block diagram](image)
Our validation framework supports the integration of Simulink models. In Simulink, a PID controller can be designed using the built-in PID block.

Figure 11. Simulink PID Controller model

A Simulink model that is connected to the corresponding signals from a DAQ card needs to be developed and integrated. The model will be initialized with the interpolation points from the calibration in order to convert the requested value from Ohms to percentage position for the Vsense. The error is then calculated and delivered to the PID Controller. The PID Controller will drive the motor with the corresponding direction and also speed in the form of a PWM signal.

In order to achieve the best performance and avoid overshooting, the PID controller needs to be tuned. For tuning the PID controller we choose the Ziegler–Nichols tuning method. It is performed by setting the I (integral) and D (derivative) gains to zero. The “P” (proportional) gain, Kp is then increased (from zero) until it reaches the ultimate gain Ku, at which the output of the control loop has stable and consistent oscillations. Ku and the oscillation period Tu are used to set the P, I, and D gains depending on the type of controller used.

Table 1. Ziegler–Nichols tuning method

<table>
<thead>
<tr>
<th>Control Type</th>
<th>$K_P$</th>
<th>$K_I$</th>
<th>$K_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.50Ku</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>0.45Ku</td>
<td>0.54Ku/Tu</td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>0.60Ku</td>
<td>1.2Ku/Tu</td>
<td>3Ku/Tu/40</td>
</tr>
</tbody>
</table>

After setting the values we need some manual fine-tuning of the PID Controller. This is done by slightly adjusting the PID Controller parameters and observe the results. Depending on which effect we need to improve, Table 2 gives us direction on what the impact will be and how different parameters on the PID controls will increase corresponding value.

Table 1. Effect of increasing parameter independently

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Rise time</th>
<th>Overshoot</th>
<th>Settling time</th>
<th>Steady-state error</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_P$</td>
<td>Decrease</td>
<td>Increase</td>
<td>Decrease</td>
<td>Decrease</td>
<td>Degrade</td>
</tr>
<tr>
<td>$K_I$</td>
<td>Increase</td>
<td>Small change</td>
<td>Increase</td>
<td>Eliminate</td>
<td>Degrade</td>
</tr>
<tr>
<td>$K_D$</td>
<td>Decrease</td>
<td>Decrease</td>
<td>No effect in theory</td>
<td>Improve if $K_D$ small</td>
<td></td>
</tr>
</tbody>
</table>

We can see how the solution is integrated within the HIL automated vision testbench using PROVEtech as the validation framework in Figure 12. CTB was already designed to support a few signal controls boards. The MotPot daughterboard has been designed in the most efficient way in order to allow the installation of at least six modules with a single CTB.

Figure 12. Integrated Sensor simulators in CTB controlled by Simulink model running under PROVEtech:RE

5. Conclusion

The innovative solution that we have integrated into our HIL automated vision testbench makes efficient usage of available interfaces in order to increase the number of simulated sensors. The increased number of sensors does not require re-architecture of HIL benches or additional expensive equipment. The original request was to upgrade not only one, but more than ten HIL benches, so the provided solution does not require an extra space which would be a problem as well. The results captured from the real HIL bench prove the theory of performance, reliability and accuracy of the device.

Figure 13. Captured transition from real HIL bench

6. Impact of Your Research

With this efficient solution, Visteon’s functional validation team acquires knowledge in the areas of Simulink, sensors, PID controllers and tuning, which are essential and can be leveraged for providing solutions to other problems where this can be applied.

7. References

