Scenario Based Testing Approach for Fusion Algorithm

Kausalya Paila
Embedded division
DMC systems India Pvt. Ltd.
Bangalore, India
kausalyap@gmail.com

***Abstract*—Driver assistance systems like lane departure sensing, forward collision warning, advanced adaptive cruise controller and collision warning system need input data from several sensors. Sensor fusion algorithm adds more accuracy and reduces any ambiguity to the sensor data that is received by the driver assistance systems from sensors. Hence testing the fusion algorithm, which is most of the times scenario based, is the most crucial and non-trivial activity in the process of testing driver assistance systems. Though most of the tools available in market for unit/system/integration testing are efficient enough, these allow the user to test the application based on assertions or logical expressions and hence are not very useful for scenario based testing. This paper proposes a tool for testing the Fusion algorithm based on scenarios. The tool proposed has a user interface and the user will be able to create scenarios and generate test vectors for fusion algorithm. Measurement fusion method which is a Kalman filter technique is derived to implement the fusion algorithm.**

Keywords - Sensor Fusion, Measurement fusion, Kalman filter, Scenario based testing, Model based testing, Driver assistance systems

I. INTRODUCTION

Model based testing includes phases of testing as it is in traditional SDLC. The typical types of testing that are covered under model based testing are Unit testing, Ring testing, Integration Testing, Functional testing.

Unit testing of models though follows the same basic concept of traditional unit testing; it always is laborious and includes lot of ambiguity. Typical steps that are followed as part of unit testing of models are

- Specifications/requirements analysis
- Test Vector generation
- Testing – use of off the shelf tools or custom tools

A. Why Custom tools when there are abundant off the shelf tools available?

There are abundant commercially off the shelf (COTS) tools available for unit testing of models. The biggest and most important drawback of these tools is that the strategic test vector generation is always non-trivial and mostly a generic process is followed for any type of application. Most of the original equipment manufacturers (OEM) develop in-house tools for testing such applications instead of using the off the shelf tools. The reason is that in generic tools,

- The test vector generation is random
- Test vector generation is based on assertions (logical expressions from specifications/requirements)
- Strategic test vector generation is always non-trivial

Also in applications `like sensor fusion and most of the driver assistance systems, the data that is to be handled is huge in size and accurate testing of such applications is very critical as well as very complex.

II. SENSOR FUSION

Advanced driver assistance systems (DAS) like collision avoidance, lane departure warning, and blind spot detection are being offered by many OEMs. Most of these systems require an overlay of data from various sensors like Radars, lasers, GPS, Inertial measurement systems and video instead of just one sensor. In order to improve the performance of each module and improve the accuracy of data, it is required to have an overlay/fusion data from all the sensors rather than individual sensor data. Sensor fusion is a method of integrating data from various sensors to obtain the best estimate of system's dynamic states. It is desired to have sensor fusion

- To improve the reliability of data
- To detect sensor failures and replace the failure data with estimated data
- To estimate parameters which are not measured directly

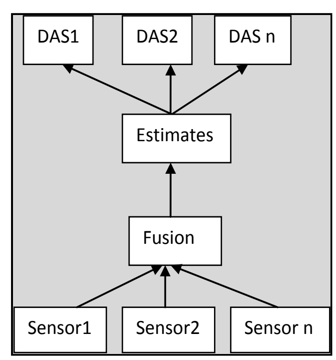

Figure 1. Role of Fusion Algorithm

Few of the reasons to select this application for present work are

- the complexity of application is very high

- It includes tremendous amount of data handling which would be very difficult to test using any of the COTS tools.

A. Scenario Based Testing of Sensor Fusion Algorithm

In the present work, a tool is discussed which can be used to generate test vectors for Fusion application based on scenarios created by the user. The tool can generate test vectors based on user configurations for host, targets and environment. User can run the simulation over time and generate test data in excel format for the entire simulation duration with a specified step time. Animation is provided in the tool so that user can have a feel for the inputs given and outputs generated. User can view the test vectors or outputs while the simulation is in progress and can cross verify the results with the animation. User can configure the output data that is of interest to the user.

The tool at the end of first phase runs in non-real time and has a limitation on the performance when no of targets is more. There can be a slight delay in the animation when compared with the simulation and the reason for this is that the simulation is given a high priority compared to display and animation.

1) Overview of the Tool

A graphical user interface is provided for the user to input all the configuration parameters required to run the simulation. The basic and mandatory inputs required as user input are the Road geometry, Host vehicle and Target vehicle's positions on global coordinates, Host vehicle and Target vehicle's velocity and acceleration, number of targets in the scenario, Simulation duration and simulation step. The physical system behavior is modeled and the sensor outputs are sent as inputs to fusion algorithm. Finally, fusion algorithm is run and the final estimates are displayed on user interface as well as the data is available for the user to save in a data file. The scenarios configured can be saved for future use. A simple animation also is provided for the user to view the physical system behavior.

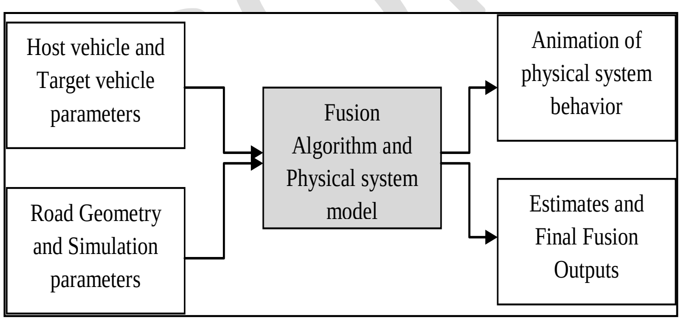

Figure 2. Fusion Testing Tool Overview

2) Environment and Sensors

A basic continuous time model is considered with state variables velocity, acceleration, distance and time (v, a, s, t). Noise, vibration, temperature etc are neglected in the present contribution. The basic equations for modeling the physical system behavior are given in equations 1, 2, 3 and 4.

$$v = \lim_{\Delta t \to 0} \frac{\Delta s}{\Delta t} = \frac{ds}{dt} \quad (1) \qquad a = \lim_{\Delta t \to 0} \frac{\Delta v}{\Delta t} = \frac{dv}{dt} \quad (2)$$

$$v_k = \frac{s_k - s_{k-1}}{t_k - t_{k-1}} \quad (3) \qquad a_k = \frac{v_k - v_{k-1}}{t_k - t_{k-1}} \quad (4)$$

The distance is calculated from the centre of gravity (COG) of the car. The sensors are located on front part of the car. The two SRRs are located on two side corners, the LRR is located on centre and the positions of all sensors are referred throughout the simulation with the calibration values. Confidence of each sensor's measurement is calculated and the sensor measurements and the confidence values are passed on to fusion algorithm.

3) Fusion Algorithm

Sensor fusion is a method of integrating the data from various sensors in order to reduce the ambiguity in data and increase the accuracy of system outputs. In the present work, Kalman filter based approach is considered for developing fusion algorithm and the measurement fusion method is considered from the two commonly used methods (1) state vector fusion (2) measurement fusion.

III. Scope and Further work

The work is under progress. In the present state, the tool runs in non-real time. Future work would be to implement a more efficient fusion algorithm to obtain the estimates, to migrate the process model to simulink[1] platform in order to achieve near real time performance and to achieve sensor interfacing with the tool to test in a near real time scenario using hardware in loop setup. With few modifications, the tool can be generalized to all driver assistance systems.

IV. Analysis and Experimental Results

Several configurations are created to capture different scenarios; test vectors generated for the created scenarios and are compared with the expected results that are calculated manually.

A. Scenario1 – Straight road geometry with single target in host vehicle's view

- Straight road environment is considered with host and one target.
- Host has two short range radars, long range radar and one vision sensor in the front part of the vehicle. No sensors are considered on rear in the present work.

[1] Matlab and Simulink are registered trademarks of the Mathworks Inc.

- Target vehicle's relative longitudinal position at start is 15 meters and relative lateral position is 15 meters.
- Host and target are considered to be travelling with constant velocity of 30 and 40 meters/sec respectively.
- Simulation is run for 28 seconds with a time step of 1 sec.

1) Expected Results

Based on the scenario parameters and manual calculations done, Fusion algorithm should indicate one valid target for 19 seconds and no valid targets after that till the last time step.

2) Actual Results

Simulation is run for the scenario-1 as explained above and the test vectors are saved to an excel sheet, excerpts of which are shown in table 1.

TABLE I. RESULTS - SCENARIO 1

Time Stamp	Target's longitudinal position	No of valid targets				
		Fusion	*Vision sensor*	*RFSRR*	*LFSRR*	*FLRR*
0	15	1	1	0	1	0
1	25	1	0	0	1	0
2	35	1	0	0	0	1
3	45	1	0	0	0	1
4	55	1	0	0	0	1
5	65	1	0	0	0	1
…	…	…	…	…	…	…
18	205	1	0	0	0	1
19	215	1	0	0	0	1
20	225	0	0	0	0	0
21	235	0	0	0	0	0
…	…	…	…	…	…	…

B. Scenario2- Straight road geometry with multiple targets in host vehicle's view

- Straight road environment is considered with host and two targets.
- Host has two short range radars, long range radar and one vision sensor in the front part of the vehicle.
- Target vehicles' relative longitudinal positions at start are 22 and 20 meters respectively. Relative lateral position is 15 meters for both the targets.
- Host and the two targets are considered to be travelling with constant velocity of 35, 45 and 50 meters/sec respectively.
- Simulation is run for 45 seconds with a time step of 1 sec.

1) Expected Results

Based on the scenario parameters and manual calculations done, Fusion algorithm should indicate two valid targets for 2 seconds, one valid target for the next time step, two valid targets till step 37 and one valid target after that till the last time step. Most threatening object would be the first target.

2) Actual Results

Simulation is run for the scenario-2 as explained above and the test vectors are saved to an excel sheet, excerpts of which are shown in table 2

TABLE II. RESULTS - SCENARIO 2

Time Stamp	Target's relative longitudinal position – Fusion output	No of valid targets				
		Fusion	*Vision sensor*	*RFSRR*	*LFSRR*	*FLRR*
0	22	2	0	0	2	0
1	22	2	0	0	2	0
2	22	1	0	0	1	0
3	22	2	0	0	1	1
4	22	2	0	0	1	1
5	22	2	0	0	1	1
…	…	…	…	…	…	…
38	22	1	0	0	1	0
39	22	1	0	0	1	0
40	22	1	0	0	1	0
41	22	1	0	0	1	0
42	22	1	0	0	1	0
…	…	…	…	…	…	…

C. Scenario3- Curved road geometry with single target in host vehicle's view

- Curved road environment with a radius of curvature (ROC) of 200 meters is considered with a host vehicle and one target vehicle.
- Host has two short range radars, long range radar and one vision sensor in the front part of the vehicle.
- Target vehicle's relative longitudinal position at start is 15 meters and relative lateral position is 15 meters.
- Host and target are considered to be travelling with constant velocity of 30 and 40 meters/sec respectively.
- Simulation is run for 28 seconds with a time step of 1 sec.

1) Expected Results

Based on the scenario parameters and manual calculations done, Fusion algorithm should indicate one valid target for 9 seconds and no valid targets after that till the last time step.

2) Actual Results

Simulation is run for the scenario-3 as explained above and the test vectors are saved to an excel sheet, excerpts of which are shown in table 3

TABLE III. RESULTS - SCENARIO 3

Time Stamp	Target's relative longitudinal position – Fusion output	No of valid targets				
		Fusion	*Vision sensor*	*RFSRR*	*LFSRR*	*FLRR*
0	15	1	1	0	1	0
1	27.282	1	0	0	1	0
2	39.412	1	0	0	0	1
3	51.322	1	0	0	0	1
4	62.945	1	0	0	0	1
5	74.216	1	0	0	0	1
…	…	…	…	…	…	…
9	114.563	0	0	0	0	0
10	123.182	0	0	0	0	0
11	131.114	0	0	0	0	0
12	138.313	0	0	0	0	0
13	144.739	0	0	0	0	0
…	…	…	…	…	…	…

D. Scenario4- Curved road geometry with multiple targets in host vehicle's view

- Curved road environment (ROC is 200 meters) is considered with host and two targets.
- Host has two short range radars, long range radar and one vision sensor in the front part of the vehicle.
- Target vehicles' relative longitudinal position at start is 22 and 20 meters respectively. Relative lateral position is 15 meters for both the targets.
- Host and the two targets are considered to be travelling with constant velocity of 35, 45 and 50 meters/sec respectively.
- Simulation is run for 45 seconds with a time step of 1 sec.

1) Expected Results

Based on the scenario parameters and manual calculations done, Fusion algorithm should indicate two valid targets for first step, one valid for next step, two valid targets for next five steps, one valid target for next four steps and no valid targets after that till the last time step.

2) Actual Results

Simulation is run for the scenario-4 as explained above and the test vectors are saved to an excel sheet, excerpts of which are shown in table 4

TABLE IV. RESULTS - SCENARIO 4

Time Stamp	Target's relative longitudinal position – Fusion output	No of valid targets				
		Fusion	*Vision sensor*	*RFSRR*	*LFSRR*	*FLRR*
0	22	2	0	0	1	0
1	34.447	1	0	0	0	1
2	46.695	2	0	0	0	2
3	58.674	2	0	0	0	2
4	70.316	2	0	0	0	2
5	81.553	2	0	0	0	2
6	92.321	2	0	0	0	2
7	102.557	1	0	0	0	1
…	…	…	…	…	…	…
11	143.828	0	0	0	0	0
12	149.768	0	0	0	0	0
13	154.845	0	0	0	0	0
…	…	…	…	…	…	…

V. CONCLUSIONS

In this contribution, a method of testing the fusion algorithm is described. The tool can be specifically configured for different scenarios so that a scenario based testing can be done instead of random test vector generation. This would reduce the time used for testing and also would increase the accuracy of test vectors generated. The ability to create scenarios and displaying the results visually is an advantage in the present tool. The tool though is not tested on any real time data; an effort to create near real time scenarios is done.

REFERENCES

[1] Stephan Matzka and Richard Altendorfer, A Comparison of Track-to-Track Fusion Algorithms for Automotive Sensor Fusion, Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems Seoul, Korea, August 20 - 22, 2008

[2] Lundquist, C., "Automotive Sensor Fusion for Situation Awareness",Thesis No. 1422 Linköping studies in science and technology.

[3] Wilfried Elmenreich and Martin Schlager, "Simulation-based Development of Embedded Sensor Fusion Applications," unpublished.

[4] Shrabani Bhattacharya and R Appavu Raj, "Performance evaluation of multi-sensor data fusion technique for test range application", Sadhana Vol. 29, Part 2, April 2004, pp. 237–247.

[5] Keren Kapach and Yael Edan, "Adaptive weighted average sensor fusion algorithms for mobile robots", IADIS International Conference Intelligent Systems and Agents 2007.