Object Fusion for an Advanced Emergency Braking System (AEBS)
Agenda

1. Rear-end collisions & EU legislation
2. How the AEB system works
3. Object fusion methods
4. Simulink implementation
5. Sensor visualisation and testing tools
Rear-end collisions & Legislation

- Rear-end collisions most common accident types for heavy vehicles
- AEB regulated on heavy trucks and buses in the EU from:
  - Nov. 2013 – for new types
  - Nov. 2015 – for new vehicles

The result of a “trailerback” accident
How AEB works
(Advanced Emergency Brake)

At risk of collision:

- **Collision warning**
  - If no driver reaction:

- **Warning brake** (-3 m/s²)
  - If no driver reaction:

- **Emergency brake** (full brake ~6-7 m/s²)

- Avoidance for **moving target vehicles**
- Attempting avoidance also for **stationary vehicles**.
How AEB works  
(Advanced Emergency Brake)
AEB (Advanced Emergency Brake)
Sensor fusion
Two sensors -> One "truth"

Sensors have different advantages

- Radar
  - Range (longitudinal)
  - Relative velocity
  - **Solid object reflection**
    - No shapes
    - Lateral position

- Camera
  - **Object type**
  - Object width
  - Lateral position
  - Range
  - Optical illusions

Redundancy required for stationary objects
Matching and merging

Ego vehicle

Direction of motion

X

Y

Fusion track props
R1
R2
R3
R4
... 
Rn & Cn

Camera track props
C1. Long distance
C2. Lat distance
C3. Long rel speed
C4. Lat rel speed

R1 <-> C1
R2 <-> C2
R3 <-> C3
R4 <-> C4
Object selection

- Object position compared to a “predicted path”
- Relative speeds from/away from predicted path for cut-in/cut-out
- Different selection zones for AEB and ACC
  - AEB has narrower field of interest than ACC
  - The selection criteria of objects reported to AEB and ACC differ
Model Based Design for fusion

Easy to get nice and readable architecture

For-each systems and Matlab Function blocks, suitable for loops and similar calculations.

MATLAB is a suitable platform for debugging and visualization.

Easy debugging in Matlab Function Block
**Code generation optimization: Solutions**

**Problem:** Data copies of bus-arrays are extremely expensive  
**Solution:** *Signal objects used to force reuse the data*

**Problem:** Execution time too long for ECU  
**Solution:** *Model divided into two ticks – probabilities calculated only every 2 ticks.*

**Problem:** Trigonometric functions are expensive on target HW  
**Solution:** *Trigonometric approximations.*  
*coder.ceval used for hand-coded functions.*
Sensor Visualization

- Sensor visualization tool developed with MATLAB/GUIDE
- Synchronize sensor/fusion data with web camera
- Matlab class used to represent each object
- Tool used offline or online (connected to vehicle network)
Development workflow

1. Gather data
2. Re-simulate
3. Find situations
4. Classify and analyze
Data gathering

- Record **ALL** required data for the system to work continuously together with a reference camera.
- Need for **real traffic data** for negative testing is **massive**.
Re-simulation (1)

Simulink Resim

C-Resim

signal.mat

result.mat

Fusion

AEB

ACC

SIDE*

Custom built emulator for the simulation need.

- Reads and writes mat-files
- Speed is ~150 times faster than Simulink.
- Verifies production code

*SIDE = Situation Identification
Situation detection

Finding situations in the data
- Events from simulation
- Interesting situations
- Problems with sensors
- ...

Compare SW versions
Regression testing

- Test scenario creation tool suitable to design test cases visually.
- Can run same test cases in
  - Simulink
  - C-Resim
  - HIL
Speed is increased by separating the job into parallel tasks and distribute them over the network with the help of MATLAB.

A master node coordinates the job while client nodes offer their computational capacity. New nodes can join in during...

### Simulation method | Relative simulation speed
--- | ---
Simulink-simulation (AEB) | 0.25 x real-time
C-resim (AEB) | 40 x real-time
Distributed C-resim (AEB) | N x 40 x real-time
Thank you!

**Summary**

- Visualization of sensor data – key to understanding any scenario.
- Development by ”Massive simulation” - enhances quality and confidence of active safety functionality.
- (Fast simulations are necessary to achieve the above)
- Mathworks tools have successfully supported this workflow
Target ECU Hardware (2013)

ECU Hardware

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>132MHz (floating point support)</td>
</tr>
<tr>
<td>RAM</td>
<td>64kByte + 512kByte (external)</td>
</tr>
<tr>
<td>Flash</td>
<td>1Mb</td>
</tr>
<tr>
<td>E2</td>
<td>64kByte</td>
</tr>
<tr>
<td>CPU load</td>
<td>~60% before introduction of AEB and fusion</td>
</tr>
</tbody>
</table>

ECU designed for I/O, gateway and simple functions.

Not optimized for massive calculations!