Distributed Embedded Systems: An Automotive Testbed

Michael J. Short
Embedded Systems Laboratory
University of Leicester

Presented At Embedded Systems Show
Birmingham, UK, 14 October 2004

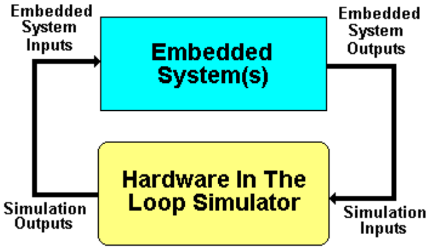
Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Seminar Overview

- Introduction
- Vehicle Dynamics – An Overview
- Motorway Simulation And Driver Models
- Automotive Control Systems
- Test System Implementation / Case Study
- Conclusions

Why Simulation?

- When developing a safety critical system, it is often inappropriate, unethical or even impossible to fully test the system within its operational environment.
- In such cases, a *simulation* of the systems environment allows for a complete and efficient assessment of a system's performance without compromising safety.



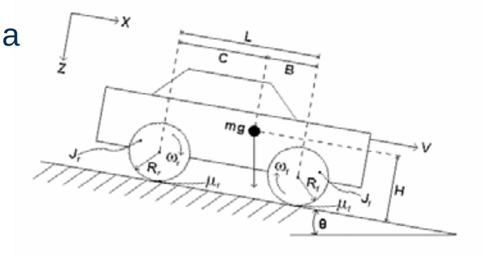
Simulation Requirements

- We require an environmental simulation that represents a motor vehicle travelling down a motorway.
- The embedded equipment under test will represent the 'host' vehicle longitudinal control systems.
- We require a realistic representation of the host vehicle's dynamics, and its interaction with other road users.
- Environmental variables (wind, road gradient, weather etc) must be included.
- A method must be incorporated to allow specific scenario's to be retested, for repeatability.

Vehicle Dynamics – An Overview

- We are concerned with Longitudinal vehicle dynamics only.
- A two-wheel traction model will successfully capture these dynamics without adding unnecessary complication.

- The car, of mass *m*, has a forward Velocity *V.*
- Each wheel has a rolling radius R_i and rotation velocity ω_i.



Vehicle Body Dynamics (1)

- The main equation of motion is obtained by summing the forces acting on the vehicle's body:

$$\dot{V} = \frac{-F_{X_T} + g\sin(\theta) - F_d(V)}{m}$$

- F_{xt} are the tractive forces;
- $g\sin(\theta)$ is the grade force;
- $F_d(V)$ are the environmental forces.

- The environmental forces may be described by two equations;

$$F_a = \frac{1}{2}\rho A C_d (V^2)$$

- Aerodynamic resistance (High speed dominance);

$$F_r = mgC_r(V)$$

- Rolling resistance (Low speed dominance).

Vehicle Body Dynamics (2)

- The Tractive forces are generated by the action of the tyre on the road surface: $F_{X_i} = \mu_i F_{Z_i}$
- The *total* tractive force is then: $F_{X_T} = 2F_{X_f} + 2F_{X_r}$
- The normal forces F_{zi} are a consequence of the vehicle axle loading:

$$F_{Z_f} = mg\left(\frac{C}{L}\cos(\theta)+\frac{H}{L}\sin(\theta)\right) - m\dot{V}\frac{H}{L}$$

$$F_{Z_r} = mg\left(\frac{B}{L}\cos(\theta)-\frac{H}{L}\sin(\theta)\right) + m\dot{V}\frac{H}{L}$$

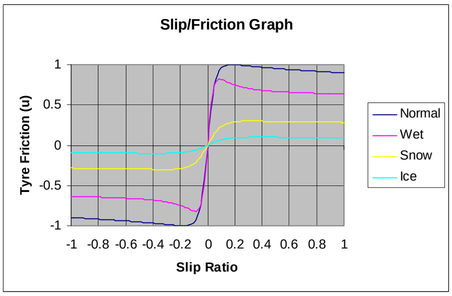
Static Term *Dynamic* Term

The Tractive Interface

- The adhesion coefficient u_i is related to wheel 'slip' λ by a non-linear function. Wheel slip can be defined as:

$$\lambda_i = \frac{V - \omega_i R_i}{\max(V, \omega_i R_i)}$$

- Negative for driving wheel;
- Positive for braking wheel;
- ≅ 0 For 'coasting' wheel.



Weather conditions effect both the shape *and* max of the curve.

The applied wheel torques accelerate and brake the vehicle via this relationship.

Wheel Dynamics

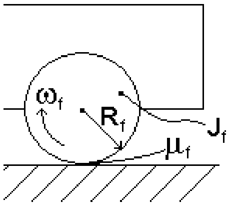
- Again, the main equation of motion for each wheel can be derived by summing the torques around each axle:

$$\dot{\omega}_i = \frac{\tau_{e_i} + \tau_{r_i} - \tau_{b_i} - \tau_{d_i(\omega_i)}}{J_i}$$

- τ_e is the applied engine torque;
- τ_r is the reaction torque;
- τ_b is the brake torque;
- τ_d is the viscous friction torque.

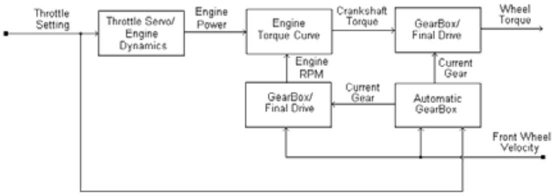
$$\tau_{d_i(\omega_i)} = \omega_i C_{f_i}$$

$$\tau_{r_i} = R_i F_{X_i}$$



- How are the engine and brake torques generated?

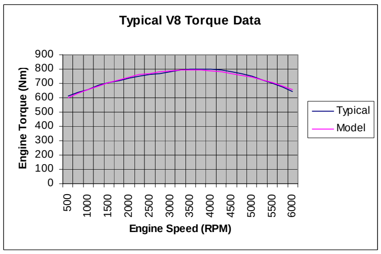
Powertrain Dynamics



The Powertrain consists of the engine, torque converter, gearbox and final drive. The throttle is actuated by a servomotor.

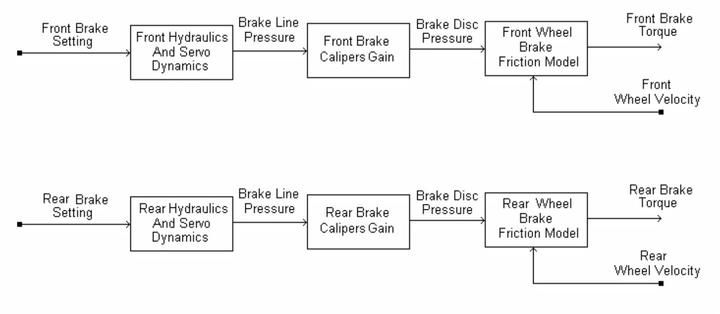
- The maximum torque, T_{max}, depends on the *engine torque curve.*
- We have modelled our engine on typical V8 data:

$$T_{Max} = 528.7 + 0.152r - 0.0000217r^2$$

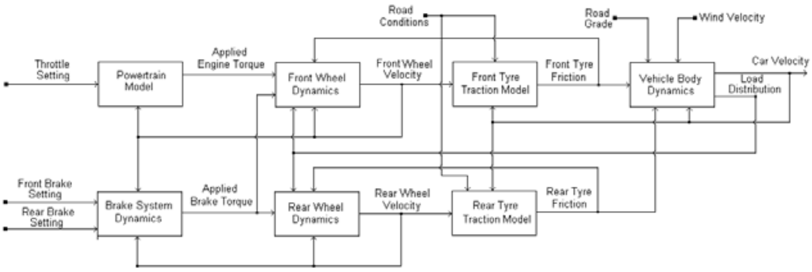


Brake Dynamics

- The braking system consists of an electronically controlled servo valve to modulate each wheel’s brake line pressure, and a simple friction model.



System Block Diagram



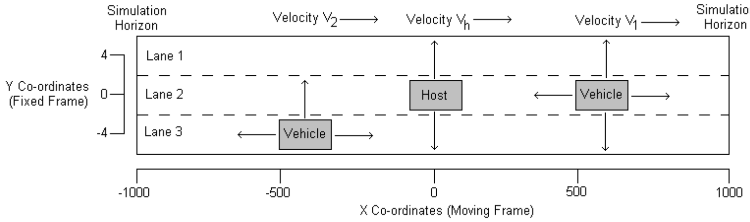
- The vehicle dynamic model is complex, interactive and non-linear.
- The vehicle is front-wheel drive.

Putting It All Together (1)

- With correct choice of state variables and parameter constants, these equations may be solved and integrated to realistically describe the motion of a modern passenger vehicle.
- The first in a series of technical reports – Number *ESL04/01* – contains further details.
- This may be found on the group's website:
- http://www.le.ac.uk/eg/embedded/

Motorway Simulation

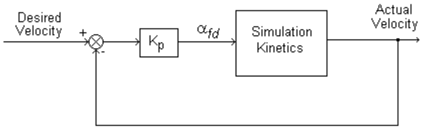
- Additional road users (vehicle/HGV) are represented by a simplified dynamic model.
- A co-ordinate reference frame is attached to the host vehicle:



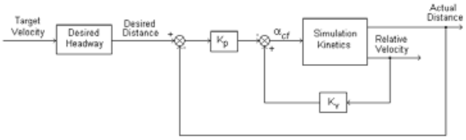
- Additional road users move w.r.t. the host; therefore it is only required to integrate their *relative* velocity.

Driver Models - Longitudinal

- Each road user (including the host) requires an ‘agent’ that decides how much to accelerate/brake etc.
- Free driving model



- Car following model



- How do we switch between the two? $\alpha = \min(\alpha_{cf}, \alpha_{fd})$

Driver Models – Lane Changing (1)

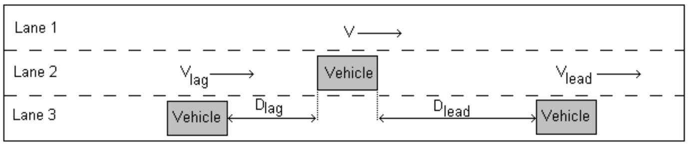
- Each ‘agent’ must also decide when to change lanes.
- Apart from manoeuvring toward an exit or avoiding a lane closure, there are three motivations for changing lanes:

-- Overtaking a slower vehicle;

-- Yielding to a faster vehicle;

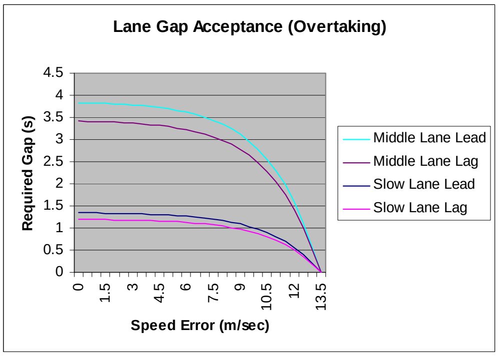
-- Adhering to the KLEWO principle.

- To describe how relevant each manoeuvre is for a given driver at a given time, a non-linear ‘urgency’ model has been developed.
- This model relates the perceived stimulus to the available gaps:



Driver Models – Lane Changing (2)

- When the gap is ‘satisfied’ for a given urgency, the ‘agent’ steers the vehicle into the adjacent lane.
- A typical ‘urgency’ relationship:



- Even at *maximum* urgency, a minimum gap is required for safety.

-- This is *NOT* shown on the graph!

- All lane changes are implemented using a sinusoidal trajectory generator.

Simulation Parameter Selection

- The simulation requires many parameters to be selected, including:

-- Driver model gains (K_p, K_v etc.);

-- Driver desired speed distributions;

-- Driver desired headway distributions;

-- Traffic composition (% HGV’s etc.);

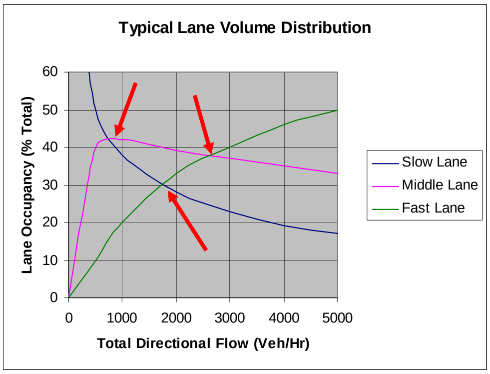
-- Traffic flow rates;

- All statistical data has been obtained from the DFT.

- Can we compare, or even optimise, the simulation to any existing data?

Is It Realistic?

- What can the traffic flow experts tell us about motorway driving?
- A typical lane occupancy diagram*:



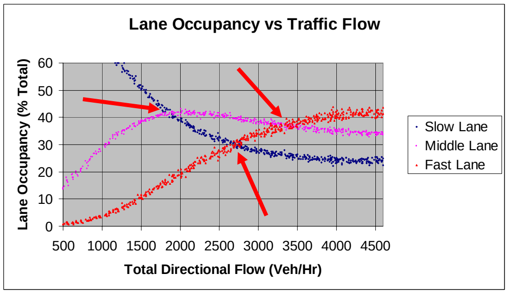
- The lane flow crossover points are important.
- The X and Y values where these points occur distinguish typical data from, say, the M6, the M42 and a German Autobahn.

---

* May, A.D. (1990) *"Traffic Flow Fundamentals"*, Prentice Hall, ISBN 0139260722.

Our 'Motorway'

- A 1994 video study* in this area has given valuable data regarding the crossover points for a section of the M1, south of Leicester.
- This data has been used to optimise the driver model parameters to give a good Macroscopic fit (statistically).



- 1-3 Crossover ≅ 2700 V/H, 30%.
- 2-3 Crossover ≅ 3350 V/H, 37%.
- 1-2 Crossover < 2000 V/H, 42%.

---

*McDonald, M., Brackstone, M. and Jeffery, D. (1994) *"Simulation Of Lane Usage Characteristics On 3 Lane Motorways"*, Proc. 27th ISATA Conf., Aachen, Germany, pp. 365-372, November 1994.

Simulation Control

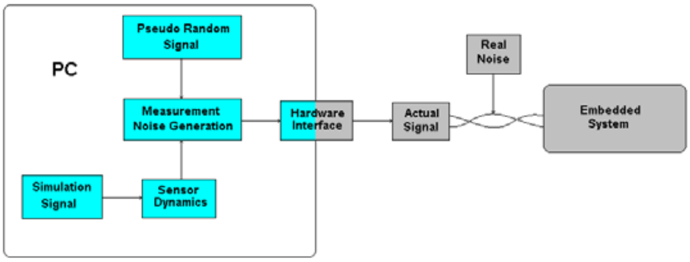
- It is desirable to control the simulation and tailor specific scenario's for particular experiments.
- This is achieved through the use of script files.
- Each script file is a simple text file that contains information such as traffic density, road conditions, wind speed…etc.
- Each simulation variable may be updated according to the passage of time;
- Each variable may also be updated according to the distance the vehicle has travelled.
- The vehicle may 'replay' a typical journey several times.
- In addition, particularly dangerous situations may be replayed to examine different system failure modes/recovery strategies.

Interfacing To The Hardware

- To keep implementation costs to a minimum, all hardware interfacing is done via the standard PC parallel port.
- Due to the nature of the signals, 4 ports were required in total.
- To enable the PC to synthesis analog signals, several high-speed, low-cost DAC's have been used.

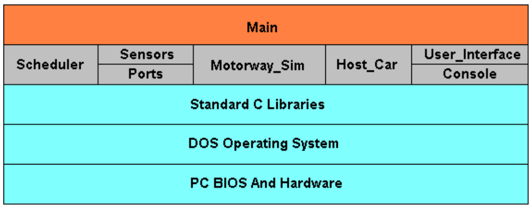
Virtual Sensors/Actuators

- In order to provide a realistic interface to the embedded system, the simulation variables in question are ‘sensed’ by virtual sensors before the value is communicated to the hardware interface.
- The virtual sensors are implemented as simple lags in the time domain, and are injected with measurement noise.
- The actuator settings from the hardware interface are propagated through a similar model before the signals are injected into the simulation.



Putting It All Together (2)

- A DOS (co-operative) scheduler was implemented to satisfy the timing requirements.
- The PC system timer was re-programmed to give 1ms ‘tick’ intervals.

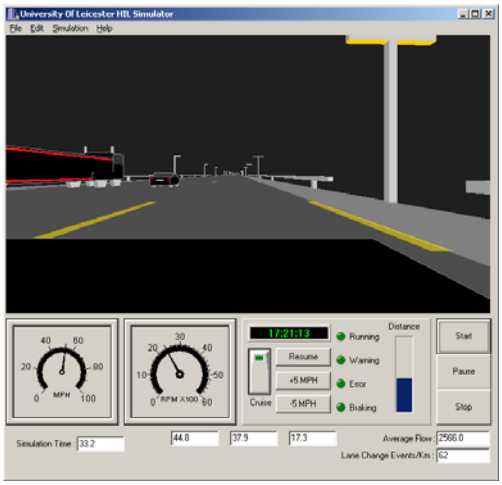


Main				
Scheduler	Sensors Ports	Motorway_Sim	Host_Car	User_Interface Console
Standard C Libraries				
DOS Operating System				
PC BIOS And Hardware				

- The second and third in the series of technical reports – reports *ESL04/02* and *ESL 04/03* – contain further details.
- http://www.le.ac.uk/eg/embedded/

Adding Graphics

- In addition to the DOS application, for display and debugging purposes a Windows-based application has been created.
- The scheduler update function is now enabled by a DPC from a high-priority multimedia timer.
- The Windows environment enables tools such as OpenGL to be implemented to create a GUI.



Simulation Time : 33.2

44.0

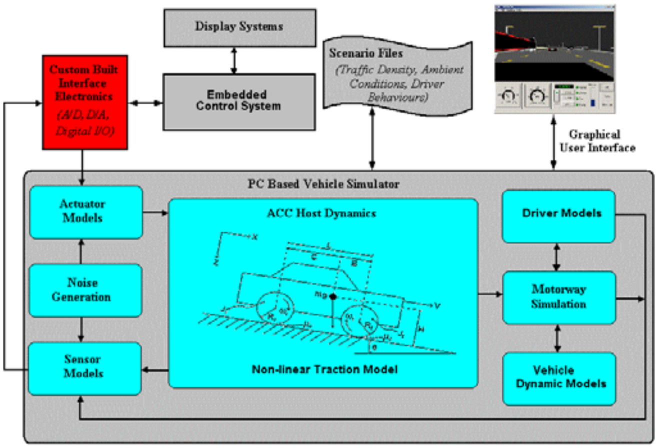
37.9

17.3

Average Flow : 2566.0

Lane Change Events/Km : 62

Simulation Schematic



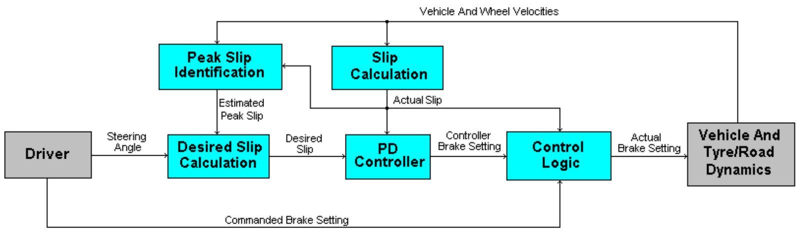
Automotive Control Systems

- In the modern passenger vehicle, up to 70 electronic control units (ECU's) are connected to a variety of sensors and actuators to realise high-level functionality, control and services.
- To reduce cabling costs and improve flexibility, these ECU's are connected to one another via a small number of communication buses.
- To implement such buses, the Controller Area Network (CAN) is a popular choice.
- The longitudinal control systems under investigation in the present study include:

-- Anti-lock Braking Systems (ABS);
-- Traction Control Systems (TCS).
-- Adaptive Cruise Control (ACC);

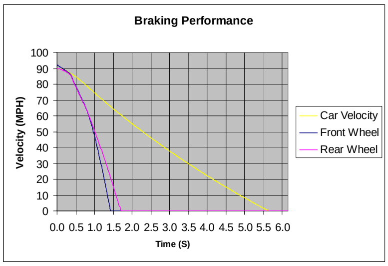
Anti-lock Braking (1)

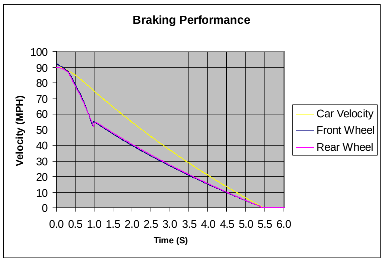
- ABS systems have increases in popularity since their introduction in the 1980's, and are now a common feature in most passenger vehicles.
- With the development of more sophisticated sensor and actuation units, more complex algorithms (such as the limited slip controller) may be employed than the normal bang-bang solenoid controllers that are in common use.
- When incorporated into a drive-by-wire system, these controllers may integrate with steering subsystems and yaw sensors to produce optimum stability when braking around corners or avoiding obstacles.



Anti-lock Braking (2)

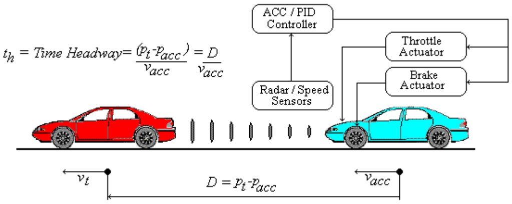
- An important feature of ABS systems – sometimes not well understood – is that it *doesn't* necessarily stop the vehicle any faster (although this is often the case).
- Importantly, if slip is controlled to an optimal value during the braking operation, lateral forces are still present at the tyre contact points.
- This means that the vehicle is steerable – and does not skid – when emergency braking is required.

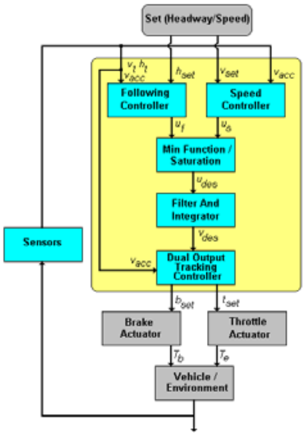




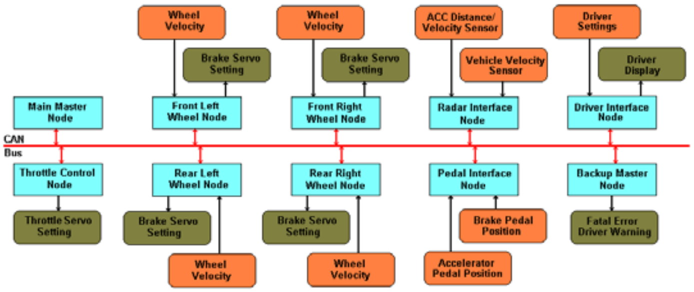
Adaptive Cruise Control (ACC)

- ACC is a relatively new technological development in the automotive field, and is said to reduce driver fatigue and the rate of auto accidents, whilst increasing fuel efficiency.
- A forward looking, Grille mounted 76-77 GHz Doppler radar senses the distance and relative speed of any oncoming vehicles, with a typical range of ≅ 150 m.





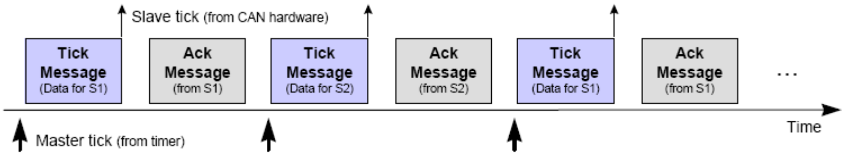
Test System Design



- A distributed drive-by-wire system incorporating ACC has been developed, using 10 C167 16-bit Microcontrollers with on-chip CAN controllers.
- The sensing, control and actuation tasks are distributed across the network. These tasks include filtering, ABS, Traction control and ACC.
- The ACC and ABS systems have been designed in accordance with the relevant ISO specifications.

Shared Clock Scheduling

- CAN is primarily an event triggered bus system, with a priority-based arbitration system.
- As the bus loading increases, jitter and latency in control loops closed by CAN also increase, often in an unpredictable manner.
- With careful allocation of identifiers and bandwidth, a highly reliable time-triggered system may be developed around CAN.
- This approach is know as Shared Clock Scheduling with CAN (SCC) – for further details see notes from the seminar given yesterday by Dr. Michael Pont.

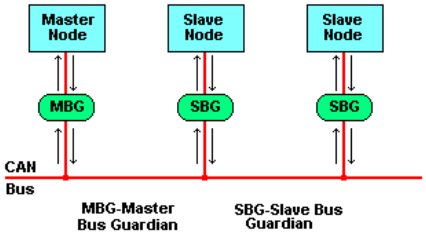


Case Study – Bus Guardians

- So-called ‘babbling idiot’ failures are considered to be the most critical failures in a shared-channel system.
- In the worst cases, such failures occur when a node (or nodes) on a bus start sending out a continuous stream of meaningless messages, making it impossible for other communication between nodes to take place.
- Although providing highly predictable behavior at low cost, shared-clock CAN schedulers (and the CAN protocol itself) have no direct way of handling “babbling idiot” failures.

Case Study – Bus Guardian Design

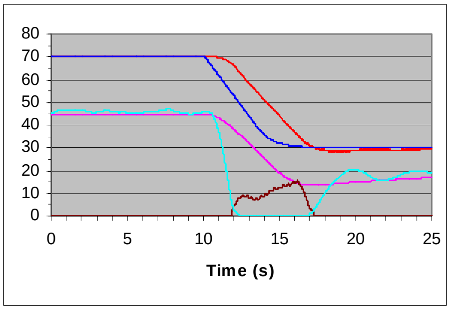
- As a related project, the group has developed simple (yet effective) bus guardians to solve this problem.
- The guardians initialise their operation during the initial exchange of messages.
- The guardians then exploit the deterministic exchange of messages as the system runs to protect the bus against unauthorised access.



Case Study – Failure Mode Analysis (1)

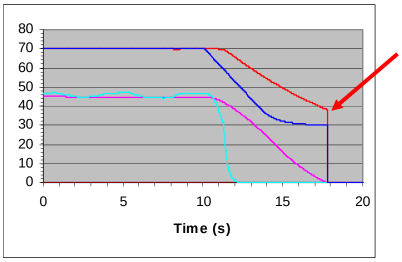
- In order to compare failure modes, a test scenario was created.
- The test scenario involves a following incident down a single lane highway.
- The lead vehicle brakes from 70 MPH down to 30 MPH with a smooth acceleration of no more than 4 ms^2.

- Host Velocity
- Lead Velocity
- Lead Distance
- Throttle setting
- Brake Setting

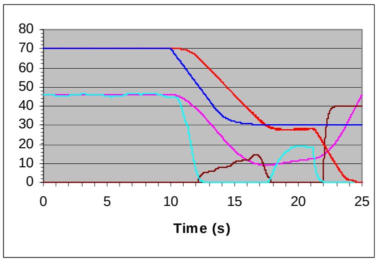


Case Study – Failure Mode Analysis (2)

- In the first test, the master starts ‘babbling’ at t=10 seconds, without bus guardian protection. Even if the driver were able to re-take control at any point, in a third of test cases a crash would still result*.
- In the second test, again the master starts to babble but this time with the bus guardian protection.
- A collision is prevented from happening with the guardians, since the back-up master switches in and applies limp-home control!



Ouch!



---

* Stanton, N. A., Young, M. S. & McCaulder, B. (1997) *"Drive-by-wire: The case of driver workload and reclaiming control with adaptive cruise control"*, Safety Science, Vol. 27, pp. 149-159.

Case Study – System Jitter Analysis

- A system jitter analysis was carried out for the system both with, and without, the guardians.

	No Guardians	With Guardians	Units
Ave. Latency	352.7	973.1	μs
Max. Latency	379.9	1005.6	μs
Min. Latency	333.0	946.8	μs
Max-Min	46.9	58.8	μs
Ave. Jitter	12.3	12.3	μs
Max. Jitter	27.2	32.5	μs

- From the results it can be seen that although the average jitter is not unduly effected, the worst case transmission time has increased by 32.5 μs above the respective average.
- The latency has also increased significantly; this is due to the extra two CAN frames that must be transmitted before the slave clock updates.

Conclusions

- This seminar has reported on work to develop a simulation facility at the ESL, University of Leicester.
- We have used the facility to preliminarily investigate the reliability of a drive-by-wire system with ACC.
- The facility has revealed the effectiveness of the bus guardian design, and also how it may be improved (re-design for improved transmission time etc).
- Next we aim to investigate the implications of choosing a time- or event-triggered design, amongst other things.
- Further work on the simulation may include implementing a 4-wheel traction model (steer-by-wire…??)
- Further reading and source code can be found at the website:

- http://www.le.ac.uk/engineering/eg/embedded

Further Reading (Tech Reports)

- Short, M., Pont, M.J. and Huang, Q. (2004) *"Safety and Reliability of Distributed Embedded Systems: Simulation of Vehicle Longitudinal Dynamics"*. Embedded Systems Laboratory Technical Report ESL04/01.
- Short, M., Pont, M.J. and Huang, Q. (2004) *"Safety and Reliability of Distributed Embedded Systems: Simulation of Motorway Traffic Flows"*. Embedded Systems Laboratory Technical Report ESL04/02.
- Short, M., Pont, M.J. and Huang, Q. (2004) *"Safety and Reliability of Distributed Embedded Systems: Development of a Hardware-in-the-Loop Test Facility for Automotive ACC Implementations"*. Embedded Systems Laboratory Technical Report ESL04/03.
- Short, M., Pont, M.J. and Huang, Q. (2004) *"*Safety and Reliability of Distributed Embedded Systems: *Control Technologies For Automotive Drive-By-Wire Applications"*, Embedded Systems Laboratory Technical Report ESL04/04.
- Short, M., Pont, M.J. and Huang, Q. (2004) *"*Safety and Reliability of Distributed Embedded Systems: *10-Node Distributed ACC System: Co-Operative Implementation"*, Embedded Systems Laboratory Technical Report ESL04/05. (Coming Soon!)
- Short, M., Pont, M.J. and Fang, J. (2004) *"*Safety and Reliability of Distributed Embedded Systems: *10-Node Distributed ACC System: Pre-Emptive Implementation"*, Embedded Systems Laboratory Technical Report ESL04/06. (Coming Soon!)