Deterministic Target Selection - Setting Requirements on Speed and Yaw Rate in Automotive Sensor Systems.

Anders Sandberg; Mecel AB; Gothenburg, Sweden

Håkan Sivencrona, PhD; Mecel AB; Gothenburg, Sweden

Prof Martin Törngren, Prof; Royal Institute of Technology, Stockholm

Keywords: Index Terms--- yaw-rate sensor, target selection, active safety functions, safety critical software, automobile safety

Abstract

A major challenge in the design of active safety systems for vehicles is to approximate the reality collected by sensors into a set of reliable and useful properties. Inability to do so can cause the active safety system to perform in a way, that will put the vehicle in a situation where the driver behaves in an unpredictable way, Unpredictable behavior in these systems can cause the vehicle to enter an unsafe state for itself or others. In order to achieve highly dependable applications it is important not only to know the behavior of sensor data, it is also important to put feasible limitations of the input data to guarantee a safe and predictable function.

This paper presents methods to improve and assess the decision material of active safety functions such as driver support systems with autonomous braking. The methods are introduced to increase the dependability of the function but also to set more accurate scope and also avoid so-called slightly out of specifications, *SOS*, fault.

Introduction

In today's active safety systems there is a need for accurate and deterministic target selection to confidently reduce the number of accidents. Given correct target selection the risks associated with high authority driver support functions are minimized. With this in mind the requirements on target selection are equally important and require the same design attention as the function itself. However target selection is a complex process and depends on several sensors. Recognition of the dependencies between raw sensor data and target selection in order to make it as deterministic as possible is a fundamental requirement.

Control applications of particular interest for this article are collision mitigation functions and other active safety systems that take surrounding traffic into account. The intricate control issue comes from coping with incorrect target selections due to sensor noise, SOS data or latencies; either undetermined or varying, between data sources. One example is if an unwanted target selection from correct to incorrect occurs. This could be when a target is suddenly considered as moving due to timing issues between range rate and host vehicle speed. This could have the effect that a stationary roadside object is sent to an application control algorithm for a safety critical system, full authority braking as an example, and cause a spurious control event that would be highly improbable from a drivers point of view. This event could even lead to accidents as the vehicle is halted in its path.

Target selection uses host movement related data and in conjunction with target related data a relevant target is selected. A typical scenario from the automotive world is that the host movement data comes from several independent sensors and the target data come from one sensor. Feeding multiple sensor data elements into a decision algorithm requires deep knowledge of the data, data sources and the processing of the sensor data in order to assess any faults that the decision algorithm can generate. This understanding is fundamental in order to be able to assess the implications on a user function relying on the decision. Hence, application knowledge is critical to understand the modes of failure that an erroneous decision can create. As active safety algorithms are in place to reduce the risk for and limit damage in case of accidents, one critical issue is if a decision failure causes an accident, an accident that would not have happened had the function not existed. If a feasible answer could be given to this question we have increased our understanding of the criticality of the decisions we make on our set of sensor data.

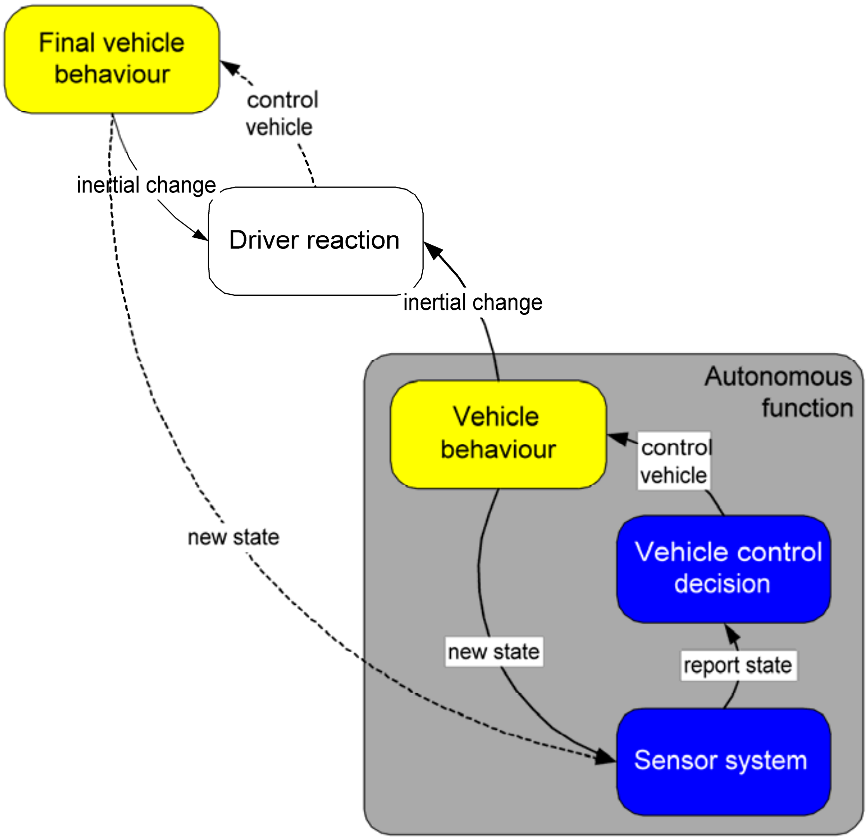

Figure 1 - Cause and effect of driver support functions.

As stated above it is important that the target selection is deterministic. In this context deterministic means: for similar or identical traffic situations the same target selection shall always be made. If this is not met it will be difficult to make user functions that behave in a deterministic way. As shown in Figure 1 different 'control vehicle' actions in similar situations could cause a unwanted ‘driver reaction’ that could have been avoided. A typical sensor structure used in target selection systems is visualized in Figure 2. Certain aspects of indeterministic behavior is hard to avoid due to the nature of the sensors used in the applications, but where possible as much of the indeterminism should be removed using algorithms and system knowledge.

Host state estimation consists of measuring speed and yaw-rate, derive road curvature and host acceleration from these. This set of data is then used throughout the algorithms for different purposes, either on its own or in combination. There are a number of other parameters that is of interest for other applications but for target selection host speed and yaw-rate are the primary ones. We will relate this to target properties using a range sensor that we assume delivers synchronized data regarding range, range rate and angle.

Speed is required for the operational mode of some sensors. As an example radars are not allowed to operate at standstill. Speed is also needed for estimating target acceleration and classification, in this case if target is or has been moving or not. Speed and yaw-rate gives two measures. We can calculate lateral acceleration and the host radius of curvature, ROC. Lateral acceleration is sent directly to an application and ROC is used in target selection. In Figure 3 the different properties of interest are shown.

To measure speed the typical solution is to measure wheel rotation. For most applications the knowledge of absolute speed is not very critical. Conversion between wheel rotation and speed is done by applying a constant wheel radius and calculating the speed with a simple formula. Yaw rate sensing relies on inertial sensors providing an analog value.

For range sensors there are on systematic faults that can and cannot be compensated for. On hardware level there is non-linearity in the modulation giving range and range-rate errors, timing inconsistencies giving angle errors. The sampling of the data provides a limited dynamic range of the data, and limited run-time performance also limits the possible data bandwidth to process. In software we find problems with associating the correct data points with each other as tracking requires some kind of track history as these types of software are systematic in their nature.

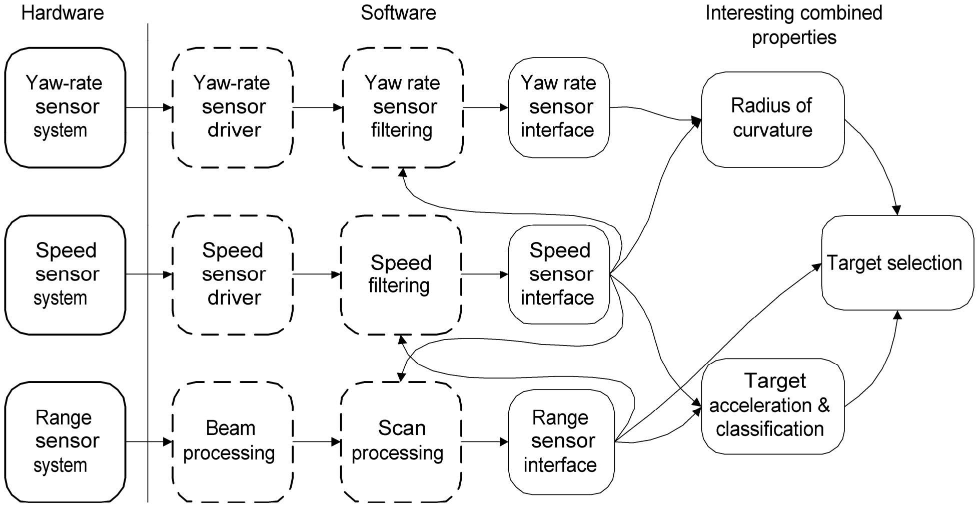

Figure 2 - Sensor architecture for driver support systems.

The criticality depends mostly on the user application. An ASIL analysis, as explained in ISO WD 26262 or SIL analysis as defined in IEC61508 will place these applications in ASIL C or SIL 2-3 in criticality. Future applications where for example full authority braking and active steering is allowed could make them even more critical. For SIL classes a 'catastrophic event' cannot be generated but an ASIL D function cannot be ruled out.

The article starts with an Introduction trying to show the reasons for the article and why the target selection as applied to the automotive domain for active safety systems is critical for application success. The Related work section lists work relevant to software fault tolerance, path prediction and target selection in the automotive domain and The Methodology section explains the host and target properties needed for target selection and how the data is calculated. It also presents secondary sources for data relating to the interesting host properties. Finally the paper presents a discussion on system failure modes, their faults, and detection methods that can be used to make the selection more robust. The Results section presents the findings on latency effects on target properties and target selection. The article ends with a Conclusion.

Related work

From the work of Kopetz (ref 5) the fault hypotheses for the sensor system can be made. In this work he also points out that one important aspect of the system is to minimize the time the system operates in a faulty state, the error detection latency. Other important concepts from a system architecture and software fault point of view is the notion of Fault Containment Regions, *FCR*, introduced to limit the propagation of a single fault condition to other independent parts of the sensor system. The work of Ammar et al in (ref 6) gives an insight in failure modes and provides a good presentation of the concepts of fault, error and failure. It also provides a deeper presentation of fault detection which is to some extent covered in this paper.

Extensive work has been made on target selection and host state estimation. Many articles present work relying on multiple data sources(refs 7-10). The focus is mainly on modeling vehicle behavior and does not model low-level sensor issues. From an application point of view Ferrara(ref 10) only looks at application aspects and does not take target selection into account, An interesting view in (ref 8) is to fuse two views on host movement, one looking at host data and a second assessing position on the road with a camera. Our paper uses a very simple model of host movement.

In the industry there are many projects aiming at developing both applications and basic frameworks for building applications on. Referenced in (ref 5) is the DECOS project that aims at developing a system architecture suitable for safety critical systems. The X-by-wire project had the same scope. The article tries to solve the target selection issue on a current architecture, as available in today's production vehicles, and does not make assumptions on synchronous systems, sensor redundancy and other safety mechanisms more suitable in other industry domains.

Methodology

For target selection we need to estimate the movement of the host and a set of target properties. Data required to correctly selecting an appropriate target depending on the traffic situation. We can distinguish between host and target vehicle properties. Host vehicle properties are, speed; yaw rate. Target vehicle properties are range; range rate; angle. All the data have in common that they originate from measurements made on the host vehicle and thus are based on the host geometric system at a specific time. We can distinguish between physical sensors measuring wheel rotation, inertial sensors measuring yaw-rate and statistical sensors measuring target properties.

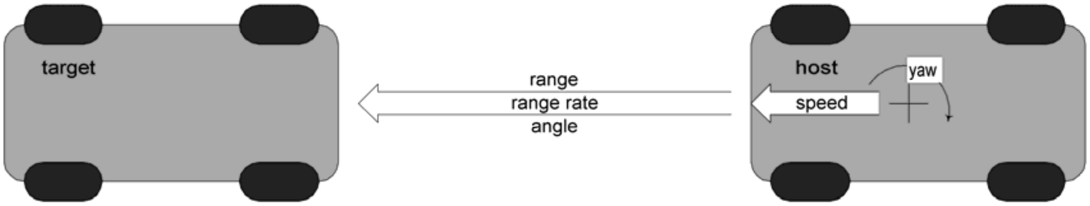

Figure 3 - Target selection properties

For target properties we typically rely on a single source sensor supplying data for range, range rate and angle. Even in multi sensor systems the actual fusion capabilities can be limited due to the available data accuracy from the sensors. In a radar/camera equipped sensor system, the camera could improve or replace the angle property provided by the radar but not necessarily provide a useful improvement of range and range rate. For host movement a typical scenario is that vehicle speed is supplied from the brake system as most modern has anti-locking brakes that supplies a measure of wheel rotation. To measure the turning speed a yaw-rate sensor is needed. The data is then used by stability control functions and the target selection part of collision mitigation systems. Since we are looking for deterministic target selection using data from independent sensors we will need a few examples of decisions based on the independent data. There are no target selection decisions made by comparing yaw-rate so we will look at speed based and speed/yaw-rate based decisions used in target selection.

Speed: Vehicle speed is used directly for the following support operations, sensor mode of operation, target acceleration and target movement classification. Speed sensing is an indirect process as measurement is primarily done by measuring wheel rotation and performing calculations based on a known wheel diameter. The example in Figure 4 shows the situation where vehicle speed is in the range between 1 and 70 m/s and tire circumference is 2.0 m. All but the low speed range are typical values for standard vehicles, the lower speed range could be discussed if we have applications needing decisions determining very low speeds or if the vehicle is stationary.

The calculation process in Figure 4 shows how speed calculation can be performed. Two basic methods are shown, pulse count and pulse period, that makes speed calculation possible. Depending on the requirements both methods will work and provide different accuracy, latency and resolution. Without references to the environment we have no physical relation to vehicle movement to rely on. In case of skidding or wheel slip the approximation that the distance the wheel travels is the same as the vehicle travels fails and all the calculations, per wheel basis, is inaccurate. This means that each method needs plausibility checks based on multiple wheel measurements, either on one axis or using data from multiple axes.

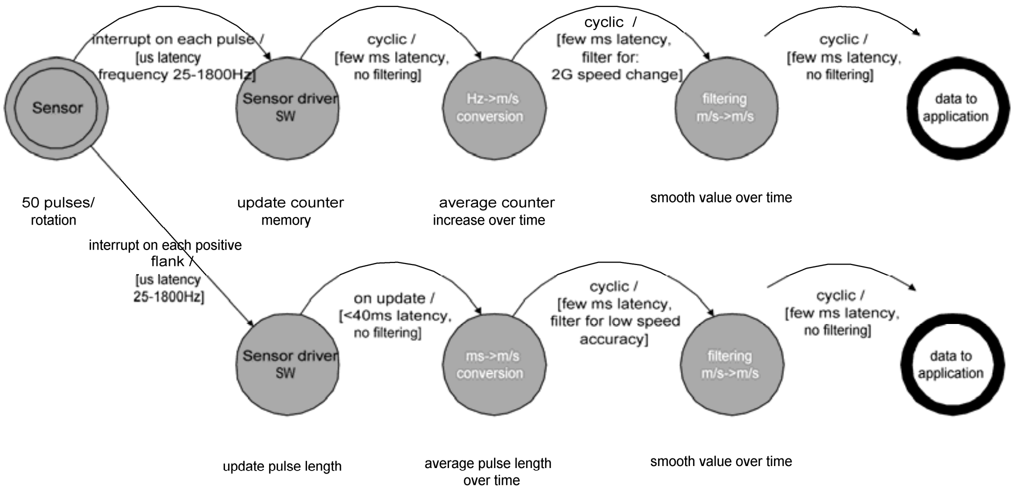

Figure 4 - Speed calculation process

For low speed measurements a longer measurement period is needed as the time between pulses is longer. The data rate to the application is considerably lower than the pulse time, at least if we are assessing if a vehicle is stationary or not. A control engineer would be better off assessing what filtering rate is suitable for a data rate of 25 Hz or lower if we need to assess lower speeds than 1 m/s as shown in the example. An indication of stationary would be lower than 0.5 km/h (0.14 m/s) for a certain time and it would be quite easy to calculate the necessary filtering to make this possible. For high speed measurements a shorter measurement period is possible as we have more frequent data to work with, here the throughput requirements can limit the possible cycle time to the order of millisecond(s). Data rates to the application are typically in the range of 5-10 ms.

Radar operation: How a radar is allowed to operate and radiate[1] is controlled by vehicle speed. To meet this we need to compare vehicle speed with a threshold and shut-down operation at a certain speed.

$\text{radarRadiating}(t) = \text{hostSpeed}(t) > v_{\text{radarOn}}$

Target speed: The estimation of target speed is used in many ways. Target acceleration can be calculated and it is used in target classification as is shown below. The initial calculation is simple as it depends directly on hostSpeed and targetRangeRate.

$\text{targetSpeed}(t) = \text{hostSpeed}(t) + \text{targetRangeRate}(t)$

To be valid the targetSpeed estimation must not be affected by host vehicle dynamics. But host vehicle dynamics will affect both hostSpeed and targetRangeRate an equal amount with a possible separation in time, if this data availability latency is t_{lx} the equation for targetSpeed will actually be:

$\text{targetSpeed}(t) = \text{hostSpeed}(t - t_{l1}) + \text{targetRangeRate}(t - t_{l2})$

Target acceleration: Target acceleration is a useful criterion when assessing a traffic situation. A drastic change in target acceleration can mean that a situation goes from normal to dangerous as the time-to-collision is lowered fast. Estimation of target acceleration is made by differentiating the change in absolute target speed.

$\text{targetAcceleration}(t) = \text{dtargetSpeed}(t)/dt$

---

[1] FCC regulation limits emitted power densities at standstill and when vehicle is moving. For cost reasons power control of the radiating element of the radar is not implemented. Instead the legislative requirement is fulfilled by turning radiation on and off.

Target classification: In this use case the definition is if a target is moving or stationary at an instance in time.

$\text{targetMoving}(t) = (\text{targetSpeed}(t) > v_{\text{isMoving}})$

From the targetMoving property it is possible to derive if a target is moveable, if a target has been moving, but is currently stationary. As we are interested in targets that have moved or are currently moving.

$\text{moveable} = \Sigma\ \text{targetMoving}(t) \geq 1$

If the targetSpeed is above a few km/h a target is deemed as moving. On common limitation on user functions is to disregard stationary objects as classification without vision causes a large risk for misclassification.

Speed and yaw-rate: Yaw-rate is typically sensed with an inertial sensor of some kind. Two examples of yaw-rate sensors are given in ref(3,4). An example of a yaw-rate calculation process is shown in figure 5. The value needs a bias correction both initially and during driving. Normal ranges for yaw-rate is in the order of ± 3 deg/s during high speed driving and up to ± 30 deg/s at low speeds. As sensors typically provide a dynamic range of ± 75-100 deg/s the resolution is an issue at low absolute yaw-rates. The temperature bias needs to be compensated for, but the resolution problem is more of a hardware constraint and cannot really be surmounted.

In **Error! Reference source not found.** the steps needed to process the sensor into actual yaw-rate is visualized. The process includes a temperature bias part as the typical cheap sensors used have a significant temperature bias. How this bias is applied is application specific but it needs to be in place. To enable bias correction we need reference points where the yaw-rate is known. One suitable such state is that yaw-rate is zero when host speed is zero. During driving the correction is more problematic, but a common approximation is that the sum of the yaw-rate = 0 over longer times.

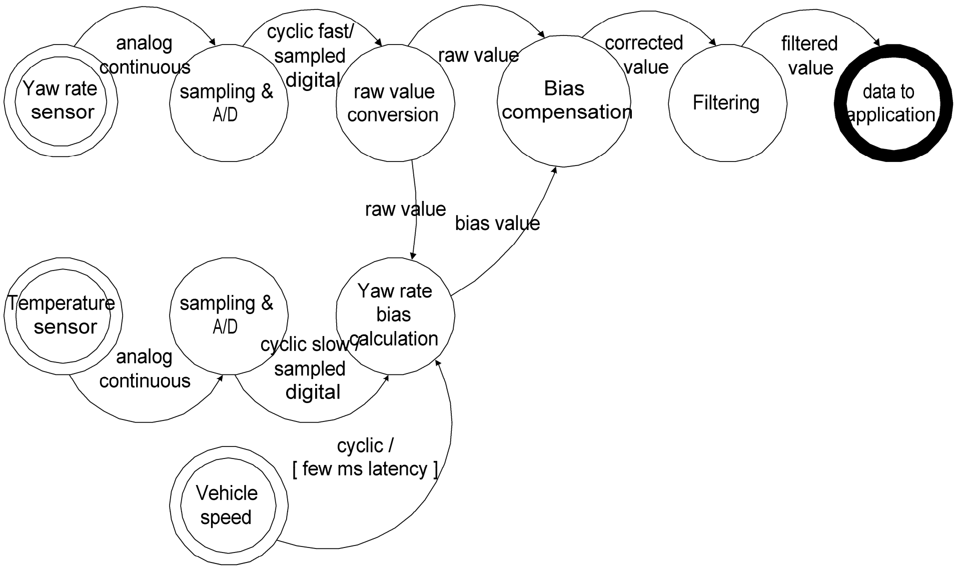

Figure 5 - Yaw-rate calculation process

Some decisions are made with a combination of speed and yaw-rate on the host side. This is needed when we estimate road geometry or assess probability of host movement.

Target selection: Target selection is the selection of which target that should be reported to a user function. To estimate road geometry we use the yaw-rate to calculate the road curvature, ROC.

$\text{ROC} = (360 \cdot \text{hostSpeed}) / (2\pi \cdot \text{yaw [deg/s]})$

We need to take both straight line and curves into consideration but from a yaw-rate point of view we can as an example use a simple threshold: if ROC > 2000 m we estimate the road as being straight. The target selection is based on finding the target that is the largest concern to an application. This selection can be based on many parameters but one common selection is to use the closest in-path vehicle using the distance from the estimated host path, y_{lc}, as the threshold.

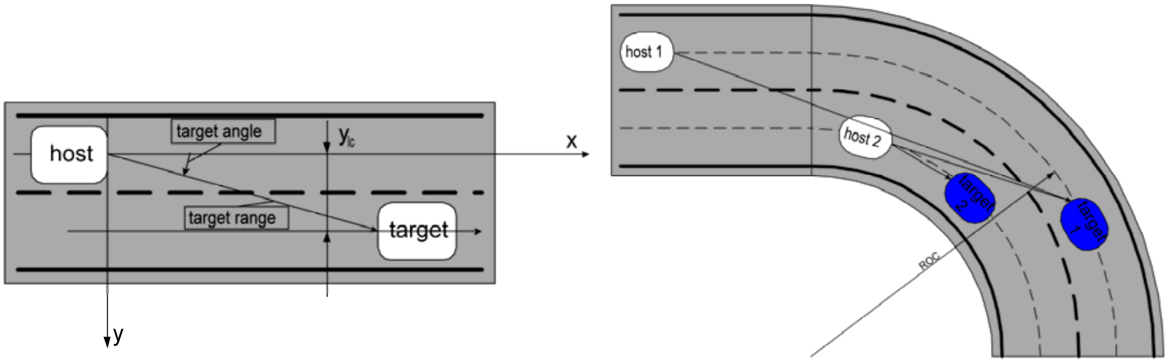

Figure 6 - Target selection on straight and curved road

For a straight road, defined by a radius of curvature over a certain threshold, the distance is easy to calculate using normal trigonometry.

$y_{lc} = y_c = \text{targetRange} \cdot \sin(\text{targetAngle})$
$x_c = \text{targetRange} \cdot \cos(\text{targetAngle})$

For a curved road we can calculate the radius from the curve centre to the target and then the distance to the path curve is a simple subtraction. The target position uses the formulas for the straight road conditions.

$x = x_c$
$y = \text{ROC} - y_c$
$\text{radiusTarget} = \text{sqrt}(x^2+y^2)$
$y_{lc} = \text{ROC} - \text{radiusTarget}$

The in path decision, in both straight roads and curves, is a simple threshold.

$\text{inPath} = y_{lc} < \text{threshold}$

Secondary control sources: To make the values used in the algorithms better quality wise we need comparative data that can be used to augment and validate the data from the main sensors.

Speed: Secondary sources for vehicle speed exist but with different availability, latency and accuracy. For active safety systems equipped with a range sensor we can estimate which targets are stationary and compare the vehicle speed with the range rate of these. This makes it possible to remove static errors in the vehicle speed estimation due to tire changes, wear and pressure changes over time. As this would be a correction factor it would only change the value not the timing of it. The time constant of the correction also needs to be small to make it robust to host dynamics and a time period much larger than the latencies is required. As the change process in itself is slow this should be of little consequence.

Yaw-rate: There are limited possibilities to assess yaw-rate apart from the main sensor. Steering wheel position is an indication that can be used at low speeds where the turning speed is low and to a lesser extent at higher speeds. A second possibility is to use differential speeds of wheels which could be useful at lower speed, this value is dependant on both geometry and actual wheel size and in essence needs its own calibration method to be useful.

<u>Faults and their detection</u>: When using vehicle speed, calculated according to Figure 4, we have assumed that the data is correct, apart from the timing issue. We need to consider the effects of faults and errors in the speed sensing system. To do this some definitions from ref(6) are used:

Definition 1: A fault is a feature of a system that precludes it from operating according to its specification.
Definition 2: An error is a deviation of a computation at computation step N compared to the expected value at step N.
Definition 3: A failure is if the system output differs from expected output with a specific input to the system.

For speed calculation the failure is to report a speed that is inconsistent with the speed associated with the actual movement of the vehicle. When looking in Figure 4 the errors associated can come from any intermediate calculation and the fault likewise be associated with both sensors and algorithms. Faults include: spinning wheels; wrong tire circumference; worn tires; wrong sample times in the data; wrong time measurements.

Detection methods: Spinning wheels, given no secondary source, there may be no stationary targets available, no secondary acceleration gyro we only have the wheel rotation as speed information. This implies no detectable fault on sensor level as the sensor functions according to specification. In this case we must rely on algorithm detection of implausible changes in host speed. Wrong tire circumference will provide a systematic fault where there is a constant difference between actual and estimated vehicle speed. With a compensation algorithm based on stationary targets it is possible to, on long term, correct this and reduce the error to a minimum. Systematic execution level faults where pulses are missed or wrong time between pulses is measured can be detected in the same way as the 'wrong tire circumference' measurement by comparing to reality. Transient misses should be possible to detect using the same methodology as for the spinning wheel and slip type fault as the vehicle will not make transient changes in speed.

The error detection latencies, as described by Kopetz in (ref 5), differ for the different types of faults. A slip type fault could be detected very quickly as the error value can be compared internally to the previous value and a mismatch could be detected. For long-term corrections we need to have limits outside of which we flag an error on the correction. If the tires are changed twice a year and there is a 2% change in circumference we need to make the correction and in conjunction make the algorithms that base their calculation on the actual value robust enough to cope with in-spec and SOS type changes in tire size. In the extreme cases there might be a need to shut operation down as an unrealistic correction of tire size is detected. One example would be using 13" rims where the norm is 16-17" rims.

The failure mode applicable for the yaw-rate sensor sub system, as visualized in **Error! Reference source not found.**, is that the reported value that does not match the actual movement of the vehicle. As the typical scenario is to have a one sensor system there are no practical back-up possibilities. We have two parallel computing tracks, one fast with conversion between raw sensor data and actual yaw-rate, and one based on history where the correction for sensor bias is made.
Any fault in the actual sensor will be forwarded through the system. Since we rely on vehicle speed a fault in the speed sensing can cause a failure in the yaw-rate by the generation of an error in the bias calculation. The temperature sensor is mainly in place to detect when the temperature change is lower than a threshold. As bias correction due to road geometry only makes sense if the change in sensor value only depends on vehicle dynamics.

The error detection latencies depend on the type of algorithm. For sensor failures a typical failure mode is a stale value. This fault has a short detection latency. For bias correction errors the latency is much longer as no good second source of vehicle state is available. Adding a steering angle sensor value to the filtering block in **Error! Reference source not found.** makes it possible to validate the actual yaw-rate against steering angle. For actual yaw-rate calculations the steering angle, especially in modern vehicle where steering is becoming adaptive is difficult as the conversion between steering wheel angle and steering angle is speed dependent.

Results

Speed: Given the methodology for data sources and secondary sources we will have a compensated speed estimate to be used as the host speed in our sensor system. This means that accuracy of data should be well known with respect to a situation where the speed is constant and we have stationary road side targets available.

Radar operation: A faulty decision will likely not cause any significant problems as the thresholds can be set to the lowest possible value. This makes an erroneous decision short lived and quickly corrected. The effect of delayed data is a delay in the decision as there is no dependency on other dynamic entities.

Target acceleration: Considering scanning mechanisms and update rates in the order of 10 Hz the targetRangeRate could have a considerable delay, orders of 100-200 ms, compared to the illumination of the target before being available for any usage. Given that a vehicle can decelerate by around 1g this equals to 1-2 m/s delay effect on target range rate, but even milder autonomous deceleration of around 0.3 g would still result in a delay effect of 0.3 to 0.6 m/s.

To be able to compensate for the measurement and data transfer latencies we need to delay the decision by a time: max(t_{l1}, t_{l2}) to be able to do the measurement. And we also need to be able to delay the earliest part by the time: abs(t_{l1} - t_{l2}) to correlate the measurements in time. If we believe that the targetRangeRate is later the equation for targetSpeed will be:

$$\text{targetSpeed}(t+t_{l2}) = \text{hostSpeed}(t + t_{l2} - t_{l1}) + \text{targetRangeRate}(t)$$

which can be reported to an application at time $t + t_{l2}$.

As an example, consider the scenario where the host vehicle brakes with 10 m/s^2 for 0.5 seconds and the target vehicle maintains speed. The used sensor latency is 200ms which can be seen in the left figure. The host acceleration curve is shown in the right hand figure 7, the indicated error shows the applied filtering on the host acceleration to give some indication of vehicle dynamics. The result in Figure 7 shows the acceleration error as a function of the used latency and the estimated acceleration in real time. We can see that it is easy to find the latency if we have access to measured data with different latency as the acceleration error is directly related to the error in latency. We can see that the whole host change is transferred to the target.

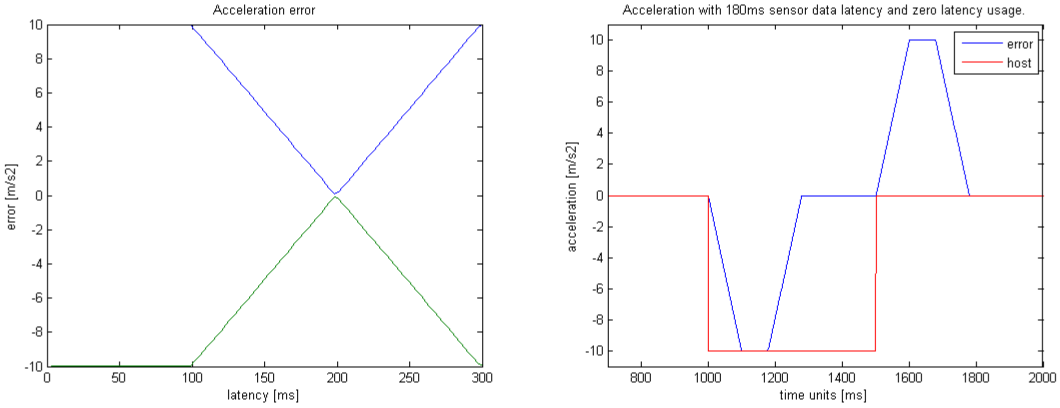

Figure 7 - Target acceleration estimation errors.

Target classification: In this scenario it is essential that the error in target speed due to data determinism is not larger than a certain part of this error budget. The error budget must be maintained under all target and host dynamic events. This puts requirements on uniform latencies in data filtering in related sensors. The criticality of the moving/stationary decision is high as a target that has been moving once must be considered as moving even if they go back to stationary again. For queue assist functions this feature would otherwise lead to frequent dropped targets as most targets are likely to become stationary prior to final control decision. The same applies to the incorrect classification where a stationary roadside object is being considered as moving. If this happens to be a fence in a curve or some other dead-ahead object it is a likely in-path target and the risk of incorrect functional actions is high. This leads to the fact that it is essential knowledge to know all delay paths on data that leads to latched status bits on a target track in order to remove the risk of taking the wrong decision on the track.

From the acceleration errors in Figure 7 we can calculate the error in target speed estimation by integrating the acceleration error. The error is directly related to host deceleration and latency error.

To minimize the target acceleration and speed error we need to calculate the latency between host and target data. As the data shares dynamic reference this is possible by applying a known dynamic event to the vehicle and measure the response. A step change in acceleration will provide a peak deviation where both value and time are dependent on the error. With this value we can correlate the response with the host change and continuously update the timing until the latency value is so near correct that the peaks in acceleration error are below a threshold.

Conclusions

A major challenge in the design of active safety systems for vehicles is to approximate the reality collected by sensors into a set of reliable and useful properties. Inability to do so can cause the active safety system to perform in a way that will put the vehicle in a resulting situation where the driver, due to inexperience, behaves in an unpredictable and outside the specified region, a situation that eventually can cause the vehicle to enter an unsafe state. In order to achieve highly dependable applications it is important not only to know the behavior of sensor data, it is also important to put feasible limitations of the input data to guarantee a safe and predictable function. This paper presents methods to improve and assess the decision material of active safety functions such as the selected use case, an adaptive cruise control with mitigation by braking. The methods are introduced to increase the dependability of the function but also to set more accurate borders and also avoid so-called slightly out of specifications, *SOS*, fault.

In the paper the results show that the latency aspect of speeds on target and host can play a major role in how an application will treat the target. With higher authority to applications the risks associated with target selection will increase. Hence a selection will likely be bound tighter with respect to target selection parameters. We have shown a method to calculate the latency using a simple host maneuver where the estimation error directly gives the latency. As indicated by figure 7 it is important to minimize the latency error in order to reduce the systematic errors in the derived target selection properties. It is of less interest to know latencies on vehicle internal properties as the latencies are smaller, however it is still important to know the data properties in order to estimate the effects of faults in the system, be they systematic, transient or in other forms that might appear. Work remains to assess parameters associated with the actual target selection, estimating noise and latency errors between actual position and estimated position in relation to the host path.

References

1. IEEE SA, IEEE Std 528-2001 - IEEE Standard for Inertial Sensor Terminology, IEEE 2001
2. IEC, IEC Standard 61508 - Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC, 1998
3. Panasonic, Datasheet, Compact Quartz Angular Rate Sensor EWTS82
4. Silicon Sensing Systems Japan, Datasheet, Angular Velocity Sensor CRS03
5. H Kopetz, On the fault hypotheses for a safety critical real time system, Springer, 2006
6. Ammar et al, A comparative analysis of hardware and software fault tolerance: Impact on software reliability engineering, Annals of Software Engineering 10 (2000), Baltzer AG Science Publishers.
7. Caveney, Numerical Integration for Future Vehicle Path Prediction, Proceedings of the 2007 American Control Conference, IEEE, 2007
8. Polychrounopoulos et al, Sensor Fusion for Predicting Vehicles' Path for Collision Avoidance Systems, IEEE Transactions on Intelligent Transportation Systems, IEEE, 2007
9. C.F. Lin et al, Vehicle Dynamics and External Disturbance Estimation for Vehicle Path Prediction.

10. A Ferrara, Collision avoidance strategies and coordinated control od passenger vehicles, Nonlinear Dyn (2007), Springer, 2007

Biography

Anders Sandberg., Graduate Student, Mecel AB and Royal Institute of Technology, Mölndalsvagen 36, S-41236 Gothenburg, Sweden, telephone – (46) 31 720-4400, facsimile – (46) 720-4490, e-mail – anders.sandberg@mecel.se.

Mr Sandberg is affiliated with Mecel AB since 1994 working mainly with active safety systems with sensor and integration issues. Since 2006 he is a PhD student at KTH, Stockholm, Sweden in mechatronics focusing on requirements and requirements management for distributed electronic systems.

Håkan Sivencrona, PhD, Mecel AB, Mölndalsvagen 36, S-41236 Gothenburg, Sweden, telephone – (46) 31 720-4400, facsimile – (46) 31720-4490, e-mail – hakan.sivencrona@mecel.se.

Dr Sivencrona, is affiliated with Mecel AB since 2005 when he left his work at SP Swedish Technical Insitutute. He is working mainly with design of dependable systems. He is currently involved in the functional safety standardization for vehicles, ISO 26262 and also AUTOSAR the automotive initiative to establish a common middleware platform for future E/E systems. He received his PhD from Chalmers University of Technology

Martin Törngren, Prof, Royal Institute of Technology, Brinellvägen 83, Stockholm, Sweden, telephone – (46) 8 709-6307, facsimile – (46) 8, e-mail – martin@md.kth.se

Prof. Törngren is with KTH, school of Industrial Engineering and Management, Mechatronics division. His research interests are in the areas of model based development - in particular model and tool integration, architectural design, systems verification and integration in the context of of mechatronics and embedded control system.