Modeling Feature Interactions Using Safety Interfaces

**Challenges**
- Feature interactions: features may unexpectedly interfere with each other through both cyber and physical channels.
- Dynamic criticality and mode switches: components can switch modes and criticality levels at runtime in response to changing operating conditions.

**Case Study: Adaptive Cruise Control (ACC) and Lane Keeping Control (LKC)**
- Physical interaction: a vehicle’s lateral motion can be impacted by the longitudinal speed controlled by ACC; as a result, ACC may seriously disturb the operation of LKC under certain conditions (e.g., curvy road).
- Cyber interaction: ACC and LKC may share computation platforms (ECUs) and communication resource (CAN bus).

**Scheduling Control Co-Simulation of Automotive Features**
- A Simulink model that is used to simulate how physical and cyber interactions affect the performance of ACC and LKC.
- Different vehicle longitudinal & lateral motion dynamics models and ACC & LKC control algorithms can be plugged into the simulation.
- CPU scheduling and bus communication are simulated using the TrueTime library.

**Simulation Example**
- A coordination algorithm running on a separate ECU will reduce the ACC cruising speed when it detects that the road curvature is above a threshold (where the red arrow points to the figure).
- The coordination logic mitigates the significant lateral motion disturbance introduced by high longitudinal speed (reduced spikes in lateral acceleration when the car enters a curvy section).

**Interface-based Analysis**
- Model both cyber and physical aspects.
- Hierarchical analysis by interface composition.
- Benefits: component-based development; separation of concerns; facilitate system integration, customization, and virtualization.

**Cyber-Physical Interaction Loops**
- The interactions in cyber and physical domains can form a positive feedback loop that may drive the features further away from the “good” operation region.
- A positive feedback loop of cyberphysical interaction example:
  - Some ACC tasks miss deadlines.
  - The ACC control performance degrades, which results in greater variation of the longitudinal speed.
  - When the car is running on curvy roads, overshooting longitudinal speed incurs significant disturbances on LKC and causes the LKC performance degradation, i.e., the lateral lane tracking error increases.
  - The increasing lane tracking error triggers the LKC internal mode switch from “tracking” mode to “searching” mode.
  - In the “searching” mode, the LKC demands a higher criticality and more computation resource.
  - The platform scheduler has to allocate more resource to LKC, and as a consequence, the ACC tasks may miss even more deadlines.

**Real-Time Virtualization for Cars**
- Consolidate 100 ECUs/10 multicore processors.
- Integrate multiple systems on a common platform; virtualization.
- Portability to Linux or Android.
- Safety-critical control on AUTOSAR.
- Example: COQOS, Integrity/Verification, Xen automotive branch.
- Must preserve real-time guarantees on a virtualized platform.

**RTXen (Real-Time Patch for Xen Platform)**
- Support real-time applications in a virtualized environment.
- Latency guarantees to tasks running in virtual machines (VMs).
- Real-time performance isolation between VMs.
- Support real-time communications.
- Xen: type-1, baremetal hypervisor.
- Dom-0 (Manager Domain): drivers, tool stack to control VMs.
- Dom-Us (Guest Domains): paravirtualized / full virtualized OS.
- Xen: type-1, baremetal hypervisor.

**Safety-Critical AUTOSAR Application**
- RTE middleware provides communication service.
- Safety-Critical Feature Examples:
  - Door Lock Control
  - Anti-Brake System
  - How to support such middleware-aware applications in a virtualization platform (e.g., RTXen)?

**Goals for a Retargetable Virtualization Platform**
- RTX — Real-Time Substrate (a middleware for middleware(s))
  - For Application Developers
    - Real-Time Portability (flexibility without losing performance)
      - Location transparency
      - Real-Time guarantees are maintained end-to-end
      - Inter-operability with other environment (e.g. non-virtualized)
  - For System Integrator/Developers
    - Path Reconfigurability (flexibility to optimize performance)
      - Pluggable environment (schedulers, ISR parameters, OS, etc.)
    - Component (“filter”) fusion/encoding/migration/placement
      - Inter-operation/compatibility interfaces can convert/encode another protocol format that is used internally.

**RTX Architecture**
- Retargetable Middleware Interface (front-end) implements a lightweight version of a target middleware (e.g., RTE).
- Signals from theretargetablemiddleware interface are encapsulated to RTX service packets.
- Adaptor (back-end) unwraps a RTX service packet and processes the target middleware signal.