MÜLLER-BBM
ACTIVE SOUND TECHNOLOGY

Integration of Active Noise Control in future IVI system designs

How to deal with low latency requirements?

Motivation

- Active acoustic technologies, influencing vehicle sound via electronic means are becoming a mainstream feature of automotive In-vehicle Infotainment (IVI) systems
- In parallel, IVI systems are moving towards highly integrated SoC's with high performance standard cores and software infrastructure

- Coming from an acoustics & application background, how can we make both match for future systems?

Outline

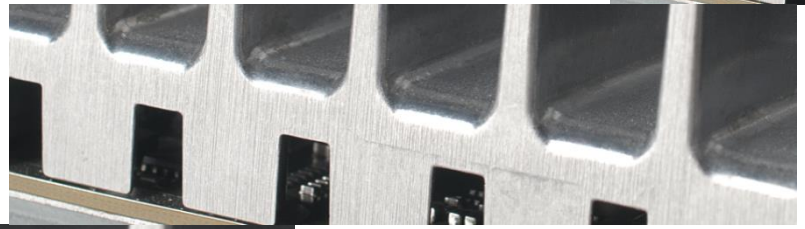
- Short introduction into Active Noise Control
- In-Vehicle Infotainment development trends
- How to integrate ANC into future IVI systems
- Summary

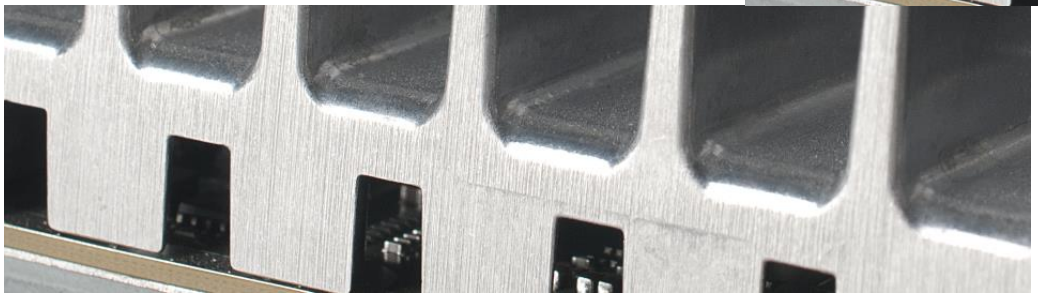
Müller-BBM Group

- Engineering services, special technical products and system solutions at the highest level
- Active employee participation model guaranteeing independency (company is fully owned by the employees)
- An open company culture that motivates and encourages performance, creativity as well as personal development
- Since more than 50 years continued growth and success worldwide

Müller-BBM Active Sound Technology

A growing high-tech spin-off with broad experience bringing advanced tech into automotive vehicles.





End of 2017 in production

- with 8 Audio Tier 1 suppliers
- for 9 global OEMs.

Our role in automotive industry

We deliver

- Radio & amplifier software to be integrated in Tier1 hardware
- Tuning tools for the development process
- System integration know-how and services

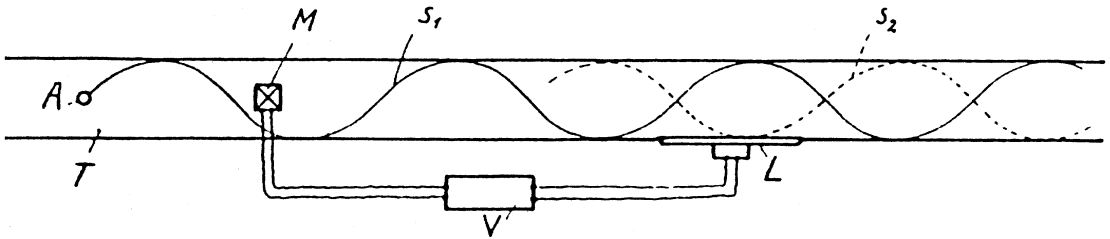
We do not deliver

- Production parts
- HW develpment and manufacturing

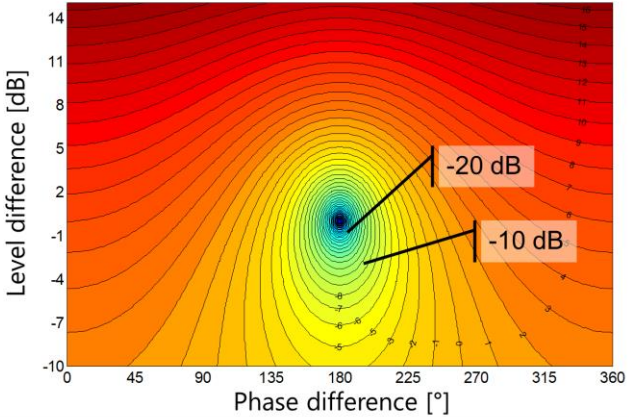
We partner with OEMs and Tier 1 suppliers.

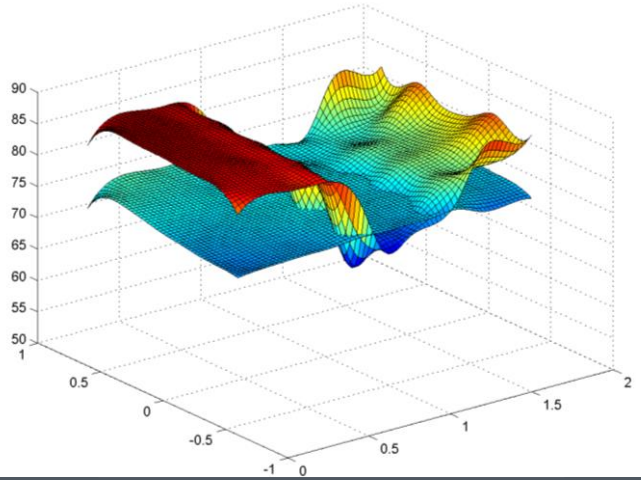
Short introduction to Active Noise Control

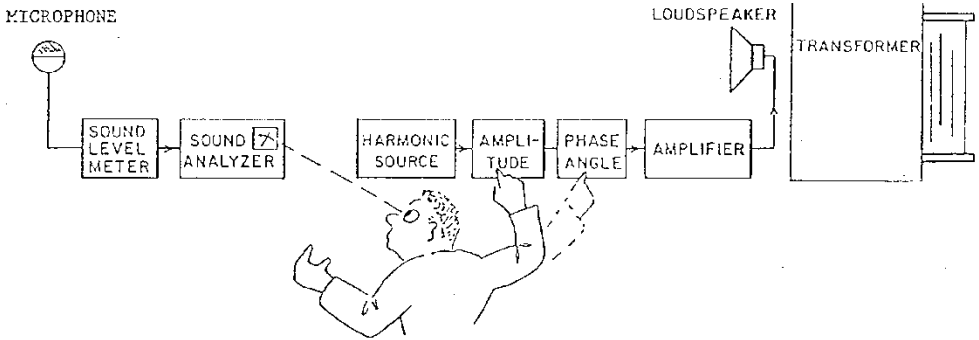
...with a special focus on control theory



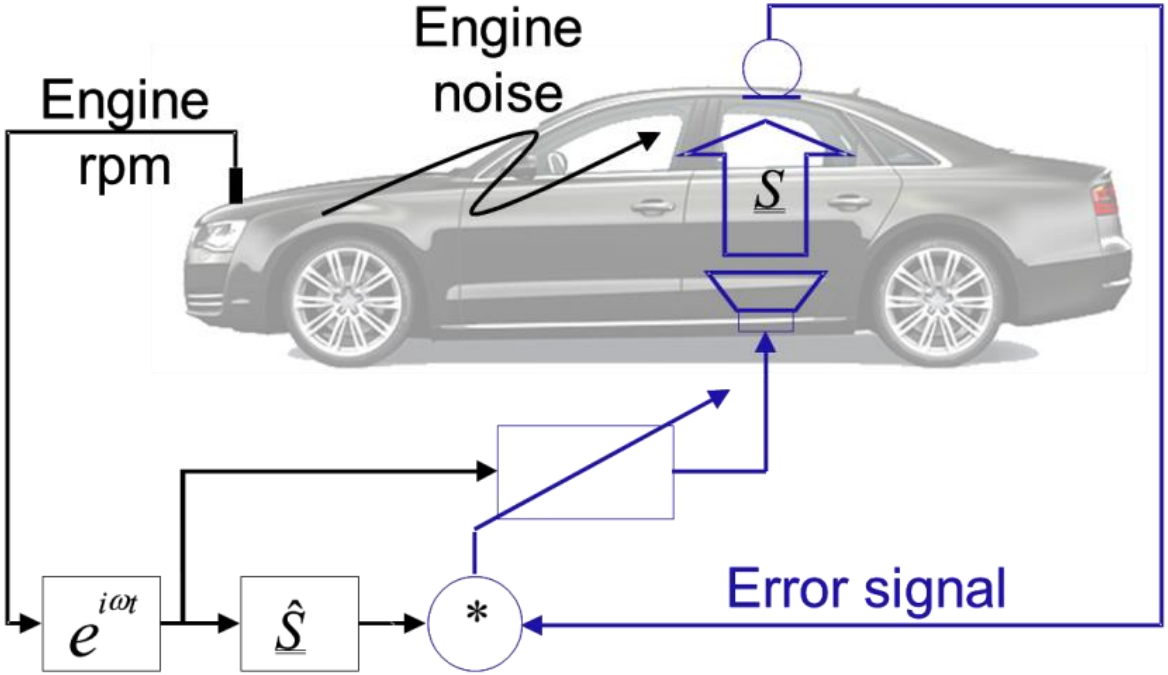
Lueg (1933): Inverse replication

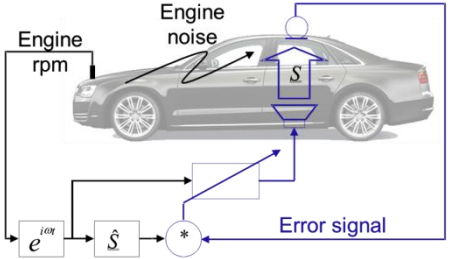


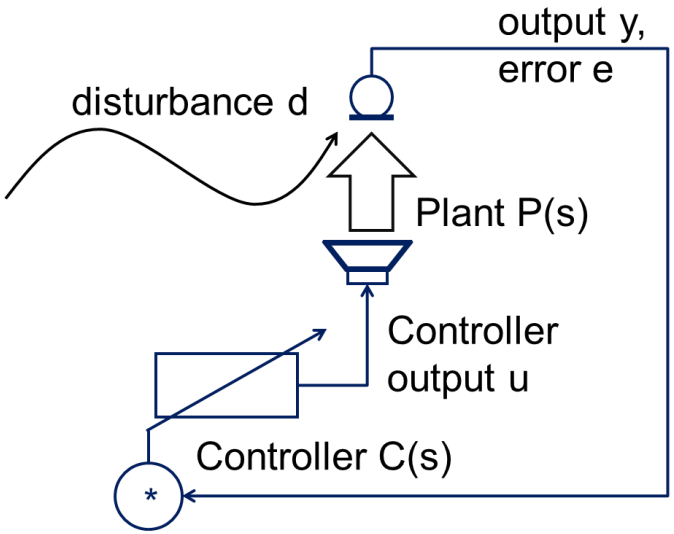


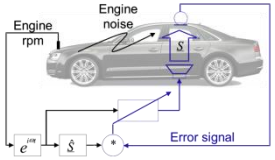


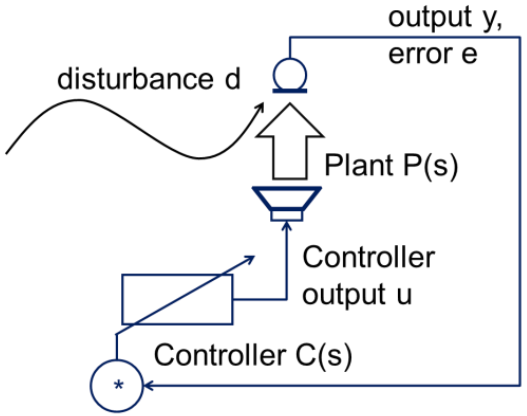
Conover (1956) Periodic signals

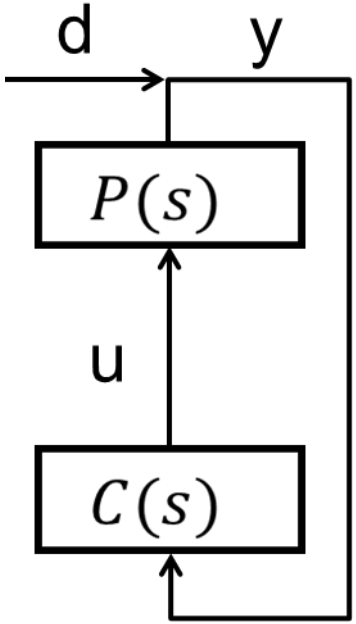

Engine
noise
Engine
rpm
$\underline{S}$
$e^{i\omega t}$
$\hat{\underline{S}}$
*
Error signal

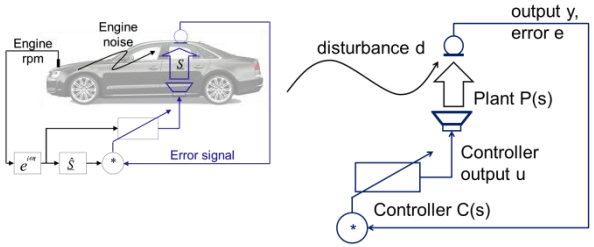

Engine
rpm
Engine
noise
S
Error signal
*


output y,
error e
disturbance d
Plant P(s)
Controller
output u
Controller C(s)
*


Engine noise
Engine rpm
Error signal


output y,
error e
disturbance d
Plant P(s)
Controller
output u
Controller C(s)
*


d
y
P(s)
u
C(s)



Open loop:

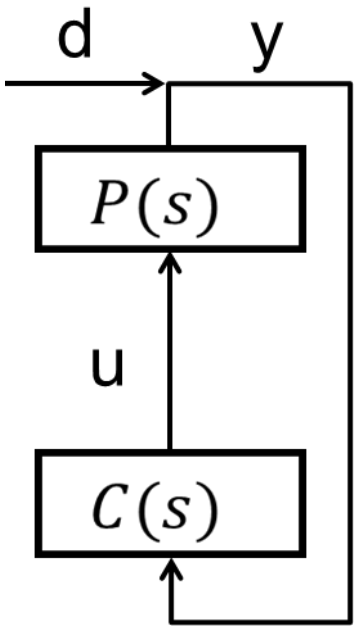
$$y(s) = d(s)$$

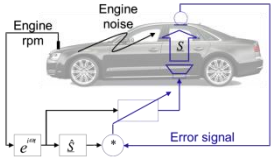
Closed loop:

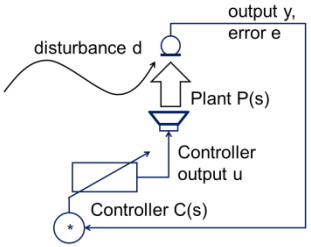
$$y(s) = d(s) * \frac{1}{1 + P(s)C(s)}$$

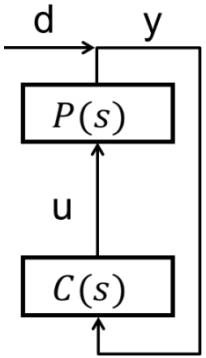
Sensitivity function
(Closed loop transfer function):

$$S(s) = \frac{1}{1 + P(s)C(s)}$$









Open loop:

$$y(s) = d(s)$$

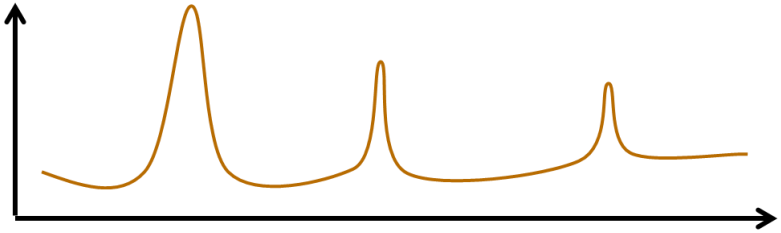
Closed loop:

$$y(s) = d(s) * \frac{1}{1 + P(s)C(s)}$$

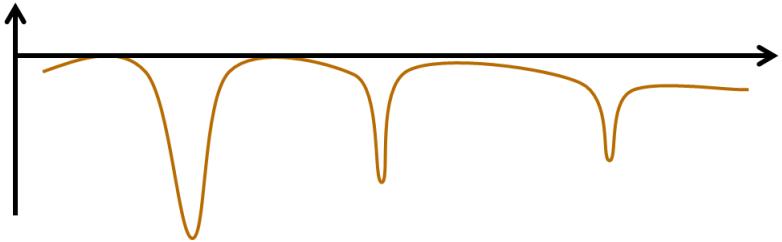
Sensitivity function
(Closed loop transfer function):

$$S(s) = \frac{1}{1 + P(s)C(s)}$$

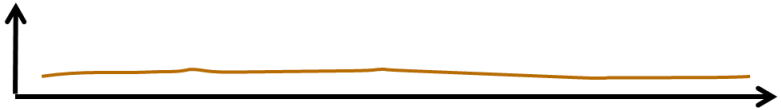
Noise: Engine (Orders)

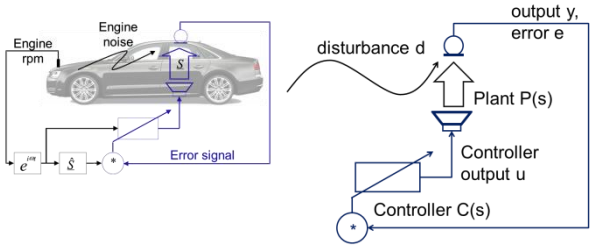


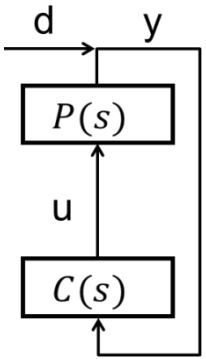
Ideal control sensitivity



Result







Open loop:

$y(s) = d(s)$

Closed loop:

$$y(s) = d(s) * \frac{1}{1 + P(s)C(s)}$$

Sensitivity function
(Closed loop transfer function):

$$S(s) = \frac{1}{1 + P(s)C(s)}$$

Bode sensitivity integral

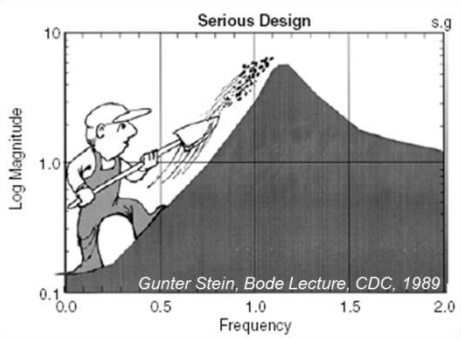
- Let S(s) stable, at least relative degree 2:

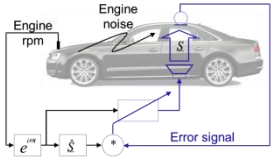
$$\int_0^\infty \log|S(j\omega)|\, d\omega = 0$$

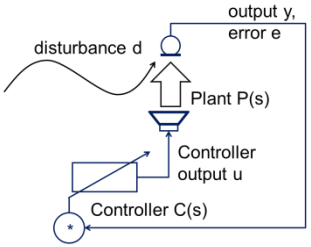
- Conservation principle

- Limit for cancellation

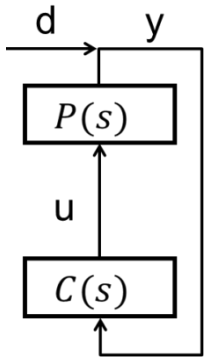
No ANC system can reduce noise without increasing noise at other frequencies!







- No ANC system can reduce noise without increasing noise at other frequencies!



Open loop:

$y(s) = d(s)$

Closed loop:

$$y(s) = d(s) * \frac{1}{1 + P(s)C(s)}$$

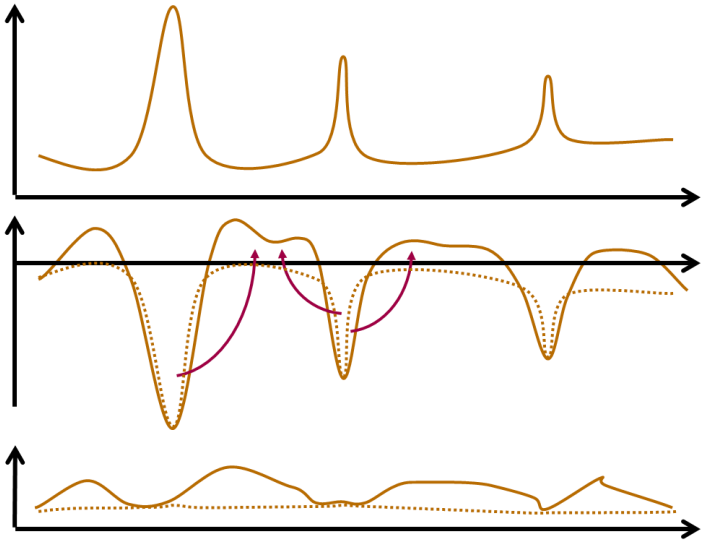
Sensitivity function
(Closed loop transfer function):

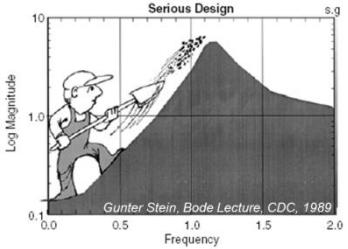
$$S(s) = \frac{1}{1 + P(s)C(s)}$$

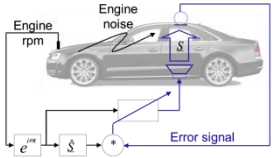
Noise: Engine (Orders)

Real control sensitivity

Result

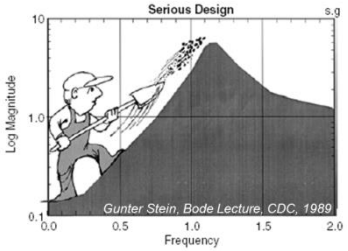






Control theory impact on ANC

- Sensitivity can be calculated
  - Robustness can be guaranteed, effects be predicted
- ANC system component and acoustical transfer functions determine sensitivity
- Control system latency (microphone – speaker) impacts ANC performance:
  Typical requirement ≤ 3 ms (without ANC function)



In-Vehicle Infotainment from a systems perspective

... the computer that, by the way, takes care of audio

In-Vehicle Infotainment

- IVI is a fully digital, highly integrated system
  - Continuous increase of functionality with limited package space and power / heat dissipation restrictions
- IVI is flexible, configurable and has multiple performance levels
  - Different screen sizes, audio levels, rear seat entertainment, radio bands, …
- IVI is a communication hub with „infinite“ interfaces
  - Consumer interfaces with short life-time, often wireless
  - Package and cost enforces bus-based communication with (very) high bandwidth at vehicle interfaces
- IVI is highly integrated with the vehicle
  - Functional sounds, warning sounds, shared displays with digital dashboards
    → Startup time, safety, security?

IVI development trends

- Standard processors / cores with operating system and middleware
  - Complexity in software, not hardware
  - Application of broadly used standards and joint development (HW & SW)
  - Scalability (multi-core processors)
  - Reduced development and update cycles, capabilities for software updates in the field
- Cross-functional SoCs
  - radio tuners & audio: NXP Dirana, SiLab,
  - application core SoC's: ST Accordo, NXP i.MX, Intel Apollo Lake, QualComm SnapDragon
  - Audio DSPs & µC: ADI Griffin, TI Jacinto
- New bus systems, partially domain specific (AVB, A2B)
- Virtualisation to separate "vehicle" and "consumer" domain

IVI system requirements

- Vehicle connectivity
  - Requires vehicle compatibility and qualified components
  - *Separate communication µC, centralised interface, data is distributed within IVI system (Info-CAN, might be tunneled: MOST, AVB)*
- Specific functions require separate HW components
  - Examples: mobile connections, radio, TV, Bluetooth, WiFi
  - Depending on system level and market → # of variants
  - *Implemented through HW modules, specific SoCs*
  - *As far as possible: integration in application processor / SoC (e.g. Software Defined Radio)*

IVI audio outputs (amplifier)

- Loudspeaker are analog components and require power via explicit wires → *many pins*
- Power amplifiers are critical thermal loads (even class D) → *package*
- Channel count and power are differentiating features of system levels → *flexibility in HW required*
- Branding dependencies → *independent sourcing of IVI HW and brand*
- *(Power) amplifier remains separate components, connected via appropriate bus system (MOST, AVB, A2B)*
- *Amplifier (as separate component) enable option for extra features. But: features can be allocated freely, including the application processor*

Audio from an IVI perspective: Signal requirements

- High bandwidth (≥ 48 kHz fs),
- High dynamic range (≥24 Bit)
- High number of channels (≥ 20 source channels, 20 target channels)
- Non-synchronous sources and sinks, different clock domains
  → ASRCs with latency
- Complex signal processing, block processing
  → latency (≥ 128 Sample, corresponds to ≥ 2.7 ms)
- Media playback: No interrupts, synchronization, latency not important
  → buffering with latency
- *Special case Low Latency: Hands-free, ….*

Audio from an IVI perspective: implementation aspects

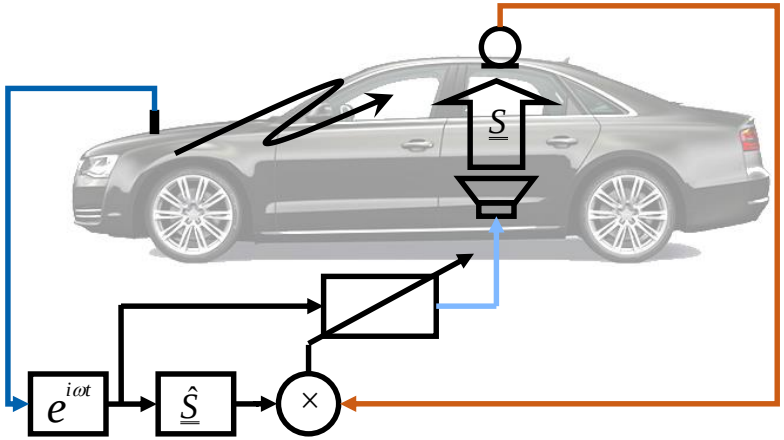
- Functionality can be implemented in a high-level app
  - computational requirements not that special (for today's application cores)
- OS provides standardized driver architectures and hardware abstraction
  - reduced complexity per SW component, software re-use, increased portability
  - HW, driver & OS suppliers gain importance, audio SW suppliers loose …
- Layered implementation allows for virtualisation of audio
  - audio functions need not be in the functional safety domain
- Automotive audio shrinks down to standard app like mobile, home, …

… allowance for latency cures everything …

… including OS, drivers, virtualisation, processor caching, …

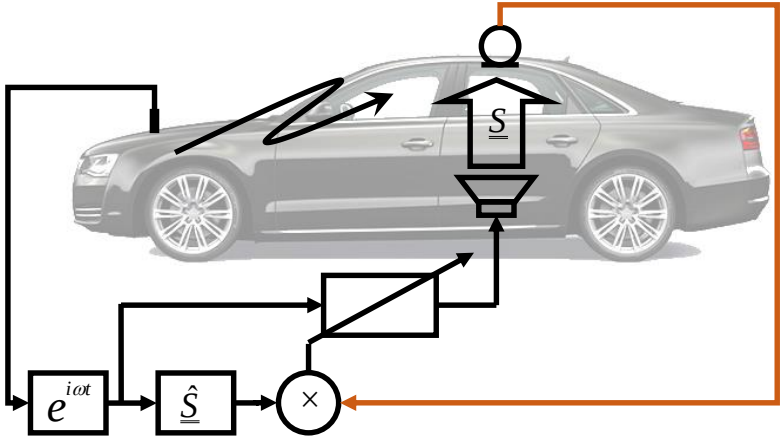
How to integrate ANC with IVI?

Why is ANC different from audio?



- Data interfaces:
  - Vehicle data
  - Microphone data
  - Loudspeaker data
- Functional SW well suited for µC or DSP
- Example implementation: Audio library function, 48 kHz fs, 16 sample block size, MIPS & memory depending on function set
- Key factors:
  - Additional Microphone inputs (cost)
  - Latency vehicle data → Algorithm/Loudspeaker
  - Latency Microphone → Loudspeaker

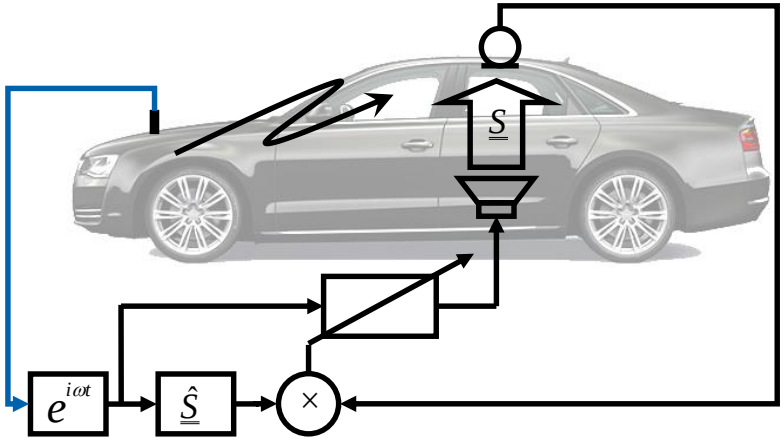
Microphone signals



- Traditional connection:
  - Phantom powered electret mic
  - Twisted pair wiring per mic
  - High pass filter for DC removal
  - Audio-Codec (Sigma-Delta) → I^2S → DSP

- Perspective:
  - MEMS mic: better low freq performace, lower component tolerances
  - MEMS → PDM → A2B → I^2S → DSP: bussed system allow for less wires / pins and does not require high pass filter for DC removal
- Other bus systems: Mind the bus interface latencies (HW & SW)

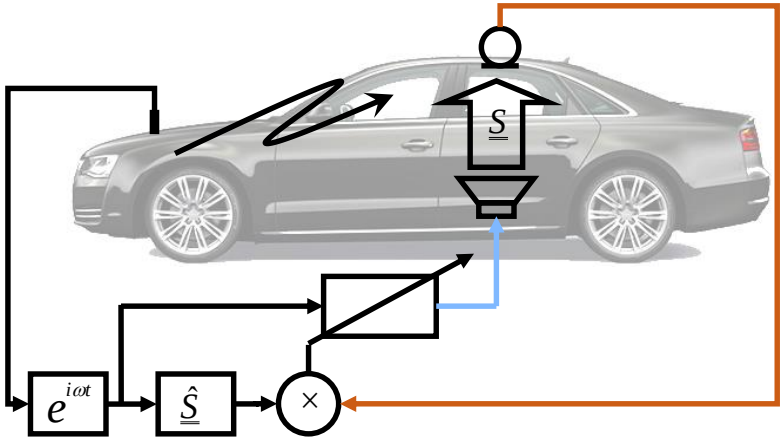
Latency vehicle data (rpm)



- Requirement: ≤ 30 ms from engine to sound
- Critical cases:
  - dynamic driving,
  - dynamic load changes (PHEV charging)
  - Cylinder de-activation

- Can be achieved with direct PT-CAN interface (approx. 20 ms) and reasonable µC / interface designs (budget approx. 5 ms)
- Critical: gateways, low speed CAN bus, tunneling through other protocols (MOST, AVB)

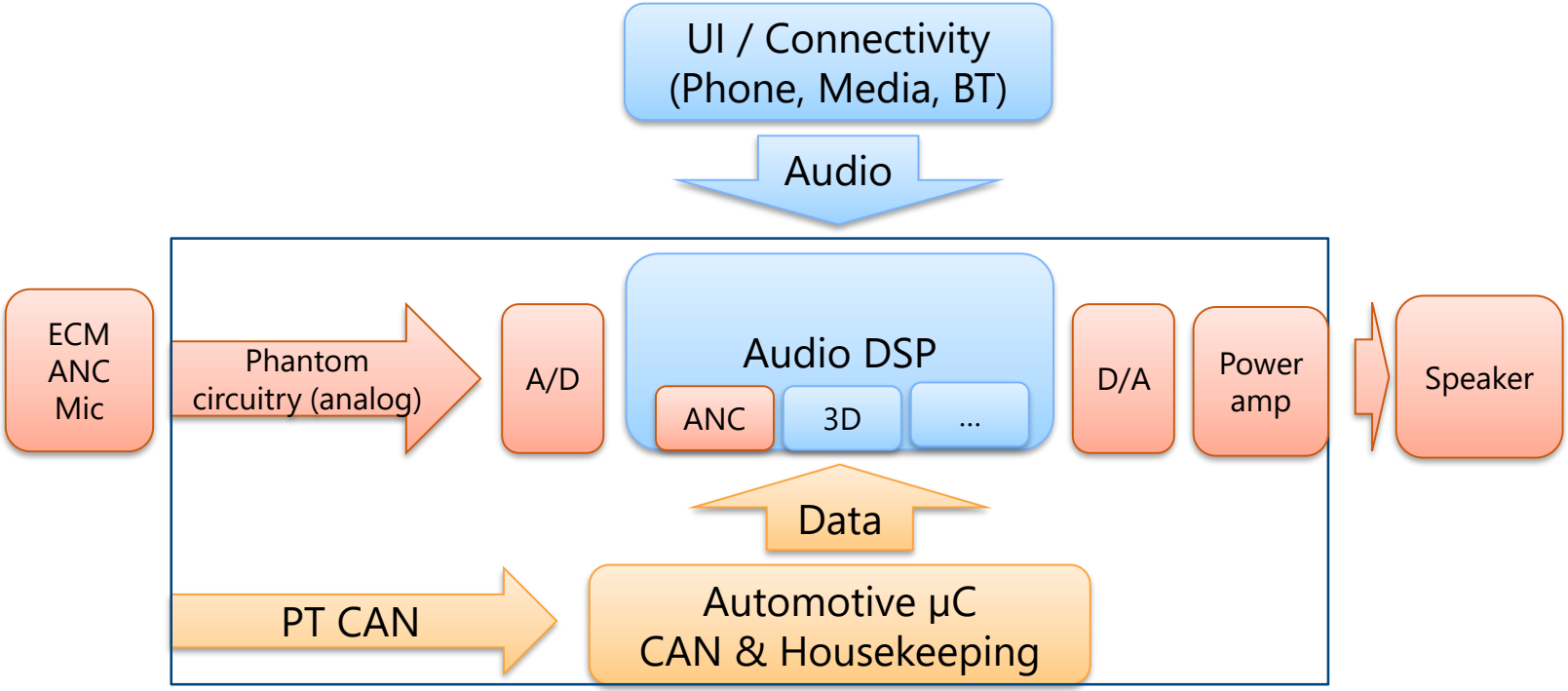
Latency Microphone - Speaker



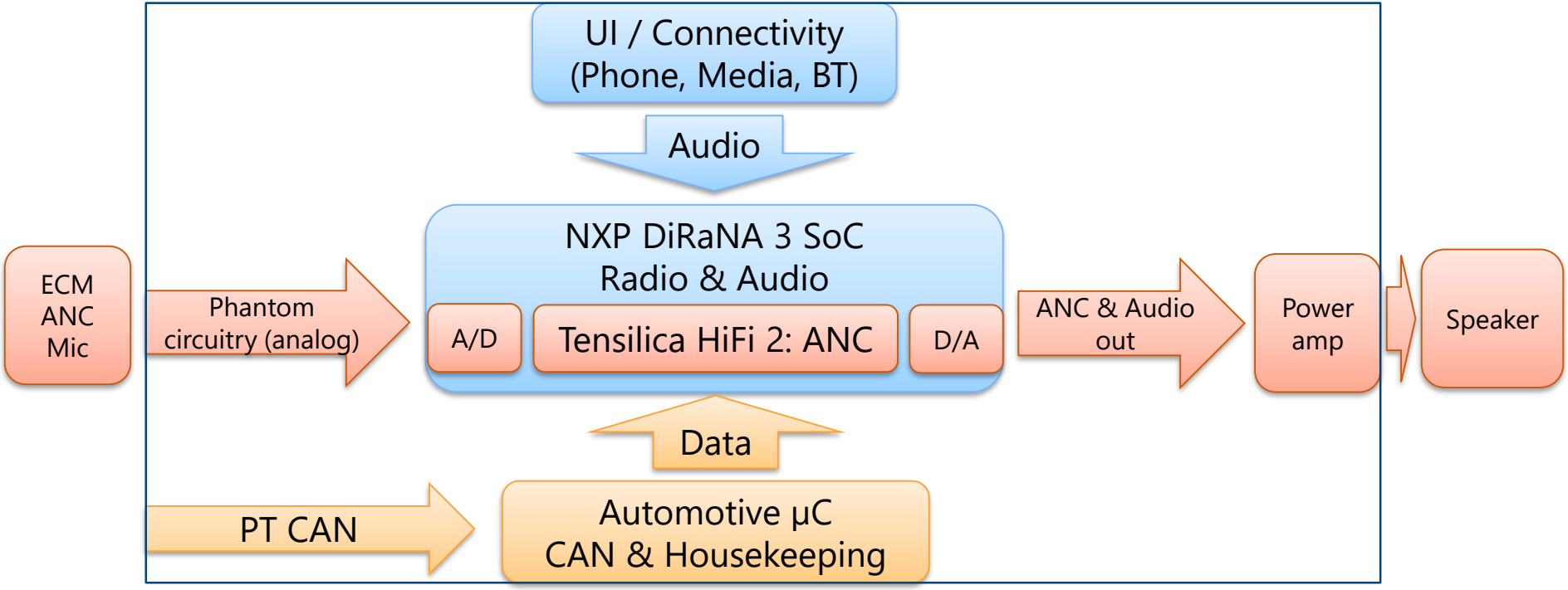
- Requirement: ≤ 3 ms
  (without ANC function)

- Latency budgets:
  - A/D and D/A typ. 0.3 – 0.5 ms each,
  - Mic (Phantom): typ. 1 ms @ 30 Hz, < 0.5 ms @ 50 Hz
  - Block processing 16 Sample, 3 Blocks @ 48 kHz: 1ms
- *Integration in main core framework (≥ 128 sample, i.e. 2.7 ms / Block) not feasible*
- *Bus systems / drivers / virtualisation would additionally increase latency*

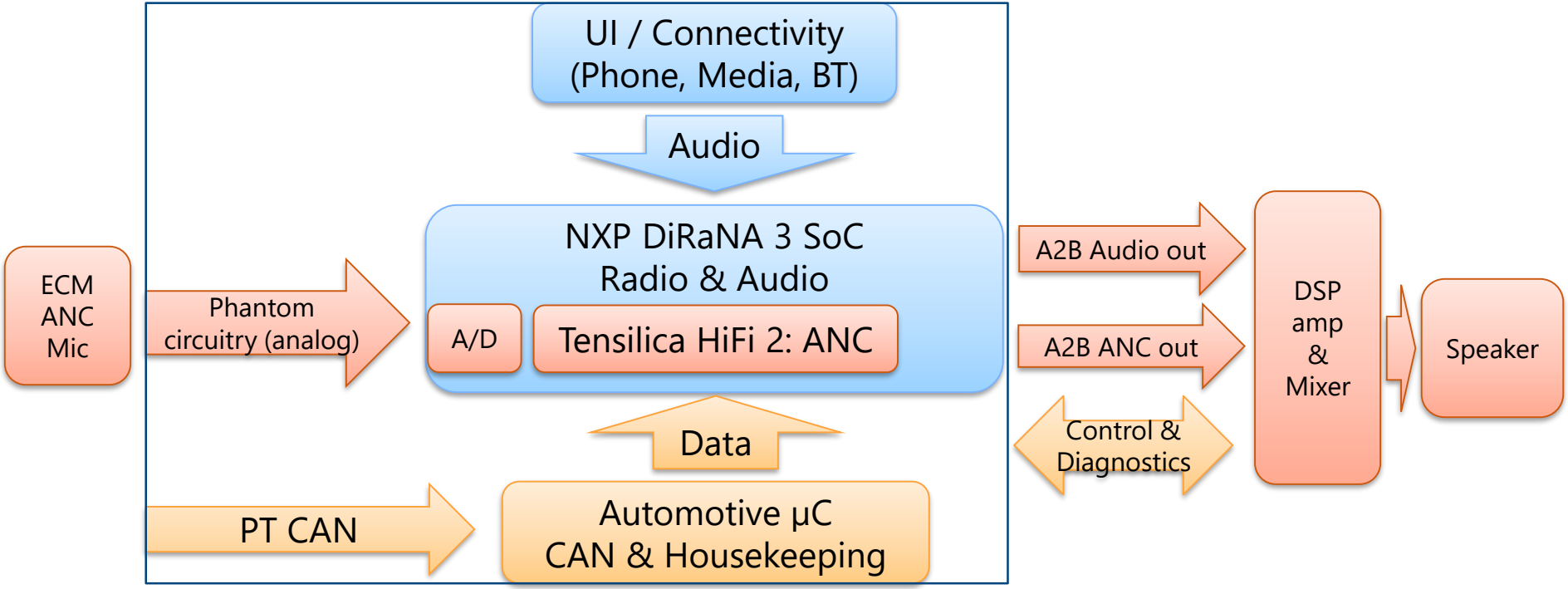
Traditional DSP amp integration example



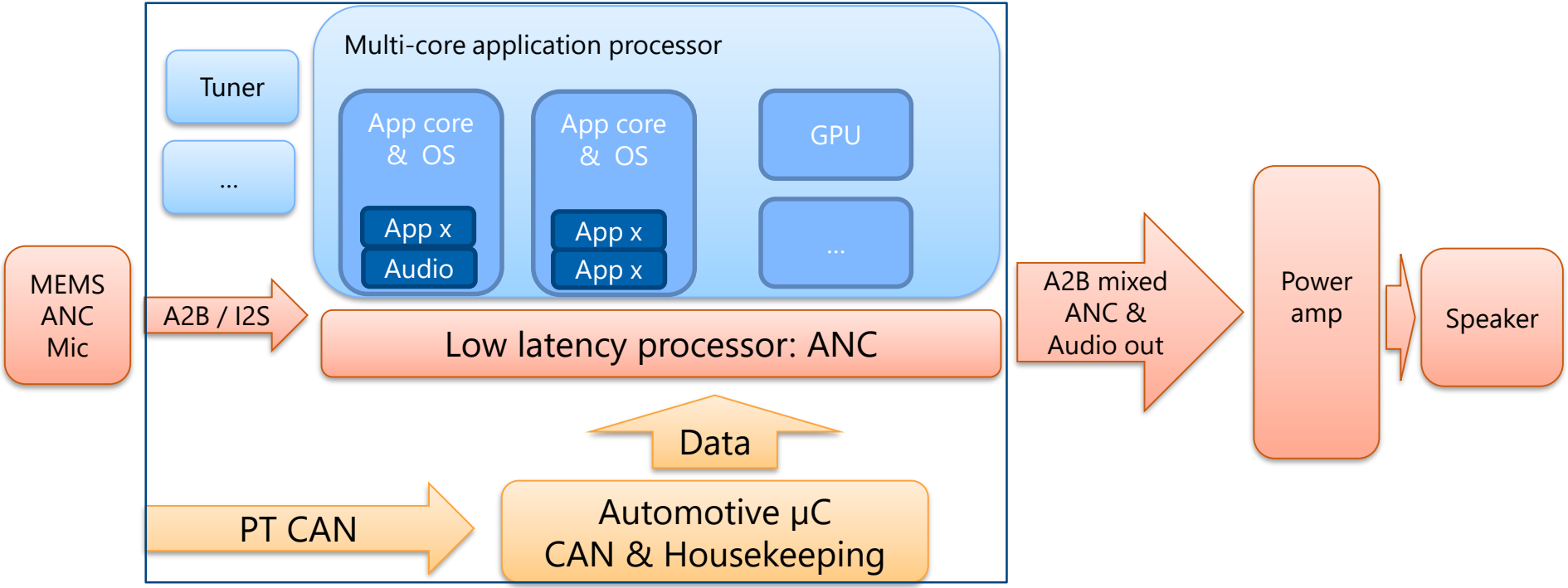
Radio integration example



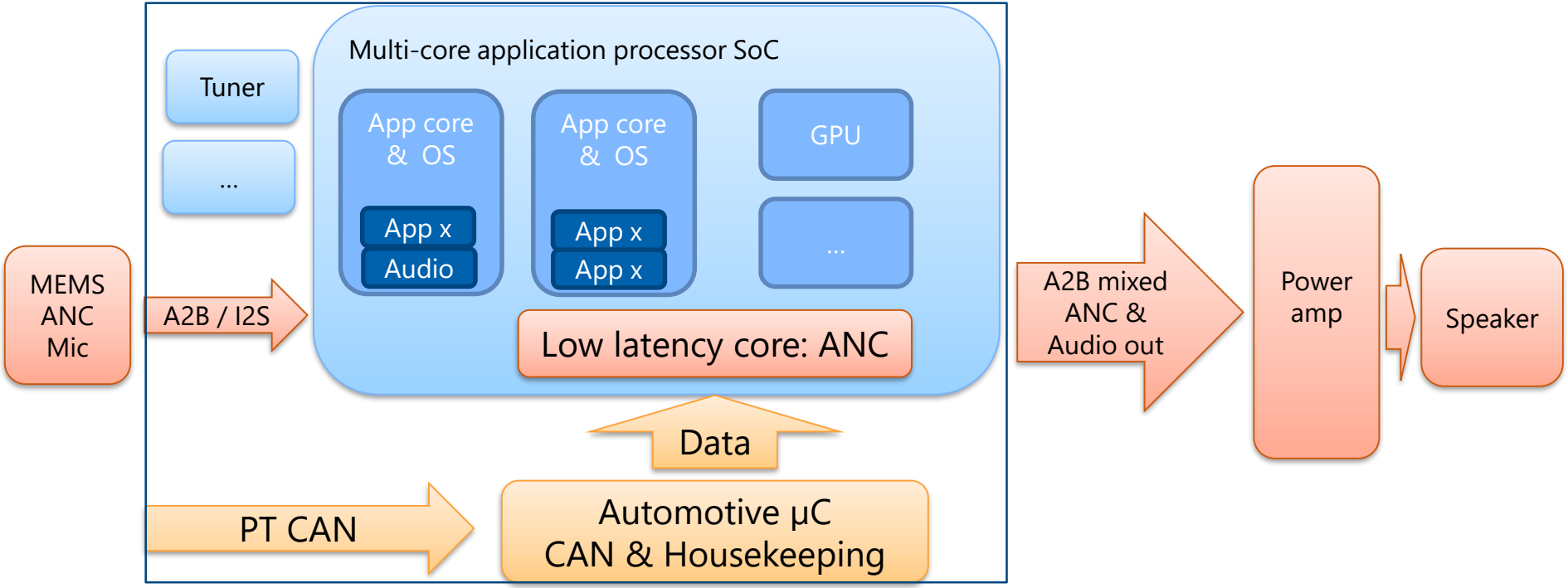
Radio w/ external DSP amp integration example



Generic "IVI application processor" integration example



Generic "IVI application processor" integration example



Summary

- IVI system trends are challenging for traditional automotive audio roles, audio is no longer special but just some more data
- ANC is a standard automotive feature to be integrated in IVI systems
- ANC does not fit in mainstream audio frameworks or OS's due to latency requirements
- IVI system designs need to consider these requirements:
  - Adding a dedicated (DSP) core for low latency processing is a well supported, efficient standard solution
  - For distributed systems, bus latencies (including interface / driver latencies), ANC signal bandwidth and diagnostic signal communication need to be considered as well

Thank you for your attention

Questions and comments welcome!