Design methods and tools for real-time (automotive) embedded systems

Abhijit Davare, Marco Di Natale, Paolo Giusto, Claudio Pinello, Alberto Sangiovanni-Vincentelli, Wei Zheng, Qi Zhu,

GM Research and Development
University of California at Berkeley,
Scuola Superiore S. Anna,
Outline

• Automotive architecture trends and challenges
• Platform-based system-level design and timing evaluation metrics
• Issues with model-based design
• From analysis to synthesis
• Activation models and end-to-end latencies
• Problem definition
  – Example
• MILP Optimization
• Case Study
Active and Passive Safety

- Passive safety (reduced personal injury in event of an accident)
- Active safety (avoiding an accident)

- Side impact protection
- Automatic emergency call
- Air bag
- Underfloor concept
- Side air bag
- Precrash action
- Seat belt
- Smart adaptive controls

- Collision avoidance
- Highway copilot
- Platooning
- EMB & EMS
- Emergency brake
- Environment recognition
- SbW (wb)
- Road recognition (LDW)
- ETC
- BAS
- ACC (Distronic)
- Autopilot
- Driver assistance
- Co-pilot
- Pilot

- ABC: Active body control
- ABS: Antilock brake system
- ACC: Adaptive cruise control
- BAS: Brake assist system
- BbW: Brake by wire
- CA: Collision avoidance
- D/W: Drive by wire
- EBD: Electronic brakeforce distribution
- EMB: Electromechanical brakes
- EMS: Electromechanical steering
- ESP: Electronic stability program
- FTC: Electronic traction control
- SbW: Steer by wire
- (wb) with mechanical backup
AS - ACC (from Continental web site)

- Adaptive Cruise Control (ACC) – Chassis Electronics Combined with Safety Aspects

As with conventional cruise control, the driver specifies the desired velocity - ACC consistently maintains this desired speed.

In addition, the driver can enter the desired distance to a vehicle driving in front. If the vehicle now approaches a car travelling more slowly in the same lane, ACC will recognize the diminishing distance and reduce the speed through intervention in the motor management and by braking with a maximum of 0.2 to 0.3 g until the preselected distance is reached. If the lane is clear again, ACC will accelerate to the previously selected desired tempo.
AS-LDW (from Continental web site)

- Lane Departure Warning System (LDW)

LDW will warn the driver if he or she is on the verge of inadvertently drifting out of the lane. Using a CMOS Camera and an image processing algorithm, this driver assistance system registers the course of the lane in relation to the vehicle. The system "sees", as it were, the course of the road and where the car is going. If the warning algorithm detects an imminent leaving of the current driving lane, the system warns the driver with haptic, kinestatic, or acoustical feedback. Possible warning alerts can be a trembling in the steering wheel, a vibrating seat or a virtual washboard sound. Series production is planned for 2005.
### Evolution of Integrated Functions

<table>
<thead>
<tr>
<th>Post-2014</th>
<th>function17</th>
<th>function16</th>
<th>function15</th>
<th>function14</th>
</tr>
</thead>
<tbody>
<tr>
<td>to 2012/14</td>
<td>function13</td>
<td>function12</td>
<td>function11</td>
<td>function10</td>
</tr>
<tr>
<td>to 2010/12</td>
<td>function9</td>
<td>function8</td>
<td>function7</td>
<td>function6</td>
</tr>
<tr>
<td>Pre-2004</td>
<td>ACC</td>
<td>Stabilitrak 2</td>
<td>Onstar emergency notification</td>
<td>Speed-dependant volume</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Brake</th>
<th>HVAC</th>
<th>Body</th>
<th>Steering</th>
<th>Suspension</th>
<th>Object detection</th>
<th>Environ. sensing</th>
<th>Environ.</th>
<th>Infotainment</th>
<th>Occ. protection</th>
<th>Exterior lighting</th>
<th>Infotainment</th>
<th>Occupant protection</th>
<th>Engine</th>
<th>Transmission</th>
<th>Telematics</th>
</tr>
</thead>
</table>
Automotive architecture trends

- Horizontally-integrated functions are becoming key differentiators and are gaining increasing authority
- An increasing number of functions will be distributed on a decreasing number of ECUs and enabled through an increasing number of smart sensors and actuators
  - today: > 5 buses and > 30 ECUs
- 90% of innovation in cars for the foreseeable future will be enabled through the Electronic Vehicle Architecture
- Transition from single-ECU Black-box based development processes to a system-level engineering process
  - System-level methodologies for quantitative exploration and selection,
  - From Hardware Emulation to Model Based Verification of the System
- Architectures need to be defined years ahead of production time, with incomplete information about (future) features
- Multiple non-functional requirements can be defined
• Automotive architecture trends and challenges
• **Platform-based system-level design and timing evaluation metrics**
  – worst-case analysis
  – stochastic analysis
• Issues with model-based design
• From analysis to synthesis
• Activation models and end-to-end latencies
• Problem definition
  – Example
• MILP Optimization
• Case Study
Deployment Design Process

- Requirements and specification definition
  - IP Functional Library Blocks
  - Functional Design
  - Map function to architecture
    - IP Architecture Library Blocks
    - Architecture Development

- System Quality Analysis

- Solution
Functional model

Input interface

function period activation mode

Jitter constraint deadline

Output interface

signal period is trigger precedence
Architecture model

Functional model

Execution architect. model

ECU

clk speed (Mhz)

register width

OSEK

ECU

CAN

bus speed (b/s)
Deployment model

Functional model

System platform model

Execution architect. model

Task

Period

Priority

WCET

Activ. mode

Message

CANId

Period

Length

Transm. mode

Is_trigger

ECU

OSEK

CAN
Tool integration platform
Design Process and Requirement

- **Functionality**
  - Control algorithm design
  - Plant Model design
  - Fault Model
  - Functional Simulation
  - Task and their WCET
  - Signals
  - Middleware
  - OS

- **Functional Requirements**
  - Allocate Function to Tasks
  - Task and their WCET
  - Signals
  - Middleware
  - OS

- **Functional Model**
  - Allocate Function to Tasks
  - Task and their WCET
  - Signals
  - Middleware
  - OS

- **Fault Model**
  - Functional Simulation
  - Task and their WCET
  - Signals
  - Middleware
  - OS

- **Mapping**
  - Allocating tasks to ECU
  - Allocating signals to BUS

- **Non Functional Requirements**
  - Input Coherency
  - Jitter
  - End to End Latency

- **Cost Architecture Model**
  - Schedulability Theory Based
  - Worst Case Analysis

- **Dependability**
  - Timing
  - Architecture Model
  - Schedulability Theory Based
  - Worst Case Analysis
**Functional Model: An example**

**Function Example**

- Acc. Pedal, brake pedal, steering wheel, Gear level
- Yaw rate, Lat accel, Veh speed, Act gear, Act direction
- Front camera
- Veh speed
- Turn Signal Switches
- Vehicle Path calc
- Forward lane path estimation
- on/off switch
- Vehicle Motion control Supervisor
- Steering torque
- led & switch
- Haptic seat
- Chime
- ggg
- fff

**Requirements and specification definition**

**< 100 ms**
Architecture Model: An example

Architecture Option

Architecture Development

F A M QA
Deployment: An example

End-to-end latencies
ECU and bus utilizations
Periodic Activation Model

- Predictable activation model easy latency computation
- Suffers from high worst case latencies

\[ L_{t,j} = \sum_{k:o_k \in P(i,j)} (T_k + r_k) \quad \text{Where} \quad r_i = C_i + \sum_{j \in hp(i)} \left| \frac{r_i}{T_j} \right| C_j \]

\[ L_{1,3} = T_1 + r_1 + T_2 + r_2 + T_3 + r_3 \]
Data Driven Activation Model

• Shorter end to end latencies
• Large interference intervals with bursty activations

Where Approx.

\[ L_{i,j} = \sum_{k:o_k \in P(i,j)} w_k \]

\[ L_{4,3} = w_1 + w_2 + w_3 \]

\[ w_i = C_i + \sum_{j \in h_p(i)} \left( \frac{w_i + J_j}{T_j} \right) C_j \]
### Case study 1

<table>
<thead>
<tr>
<th>Functions</th>
<th>Reqmt</th>
<th>Alt 1</th>
<th>2</th>
<th>Alt 4</th>
<th>4exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>function5</td>
<td>180</td>
<td>433.92</td>
<td>178.92</td>
<td>312.32</td>
<td>119.82</td>
</tr>
<tr>
<td>function4</td>
<td>100</td>
<td>395.21</td>
<td>155.21</td>
<td>180.93</td>
<td>70.93</td>
</tr>
<tr>
<td>function3</td>
<td>300</td>
<td>520.99</td>
<td>170.99</td>
<td>489.19</td>
<td>139.19</td>
</tr>
<tr>
<td>function2</td>
<td>300</td>
<td>695.38</td>
<td>232.88</td>
<td>728.68</td>
<td>208.68</td>
</tr>
<tr>
<td>function1</td>
<td>300</td>
<td>695.38</td>
<td>232.88</td>
<td>728.68</td>
<td>208.68</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functions</th>
<th>Reqmt</th>
<th>Alt 5</th>
<th>5exp</th>
<th>Alt 6</th>
<th>(event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>function5</td>
<td>180</td>
<td>310.58</td>
<td>118.08</td>
<td>130.1</td>
<td>60.06</td>
</tr>
<tr>
<td>function4</td>
<td>100</td>
<td>300.97</td>
<td>70.97</td>
<td>30.97</td>
<td>58.47</td>
</tr>
<tr>
<td>function3</td>
<td>300</td>
<td>532.74</td>
<td>162.74</td>
<td>333.9</td>
<td>123.9</td>
</tr>
<tr>
<td>function2</td>
<td>300</td>
<td>537.24</td>
<td>167.24</td>
<td>303.8</td>
<td>113.8</td>
</tr>
<tr>
<td>function1</td>
<td>300</td>
<td>537.24</td>
<td>167.24</td>
<td>318.9</td>
<td>128.9</td>
</tr>
</tbody>
</table>

- By transmitting messages “on event”, the worst case latency can be reduced in most cases.
- By properly allocating functions to ECUs the end-to-end latency can be improved.
Stochastic and simulation-based analysis

- **Simulation**
  - Built C++ simulator for can message analysis (at bit level – only arbitration)
  - Currently being expanded to end-to-end computations, periodic sampling model for latency analysis

- **Stochastic analysis**
  - Approximate analysis of pmf of message latencies in CAN bus (complete - target ?)
  - Future work
    - End-to-end analysis of sampling model
    - Regression-based analysis to define pmf from general information (such as load or loads at harmonic rates)
Stochastic and simulation-based analysis

Figure 5. Latency cdfs of two high priority representative messages in the test set

Figure 6. Latency cdfs of two low priority representative messages in the test set

62 msg set (subset of chassis bus). Low priority msg – Distributions of latencies
Outline

• Automotive architecture trends and challenges
• Platform-based system-level design and timing evaluation metrics
  – worst-case analysis
  – stochastic analysis
• Issues with model-based design
• From analysis to synthesis
• Activation models and end-to-end latencies
• Problem definition
  – Example
• MILP Optimization
• Case Study
Issues with model-based development

- Model-based design methodologies
  - improve the quality and the reusability of software.
  - The possibility of defining components (subsystems) at higher levels of abstraction and with well defined interfaces allows separation of concerns and improves modularity and reusability.
  - The availability of verification tools (often by simulation) gives the possibility of a design-time verification of the system properties.

- However, most modern tools for model-based design have a number of shortcomings
Issues with model-based development

- Lack of separation between the functional model and the architecture model
- Lack of support for the definition of the task and resource model
- Insufficient support for the specification of timing constraints and attributes
- Lack of modeling support for the analysis and the back-annotation of scheduling-related delays
- Issue of semantics preservation
Outline

- Automotive architecture trends and challenges
- Platform-based system-level design and timing evaluation metrics
  - worst-case analysis
  - stochastic analysis
- Issues with model-based design
- Time predictability and timing isolation
- **From analysis to synthesis**
- Activation models and end-to-end latencies
- Problem definition
  - Example
- MILP Optimization
- Case Study
Opportunities for synthesis

Periods Activation modes

System

Functionality

System Architecture

Mapping

Task and message priorities

Function to ECU allocation

Number and type of ECUs and buses

System topology

Simul. annealing

DATE 07 (MILP)

RTAS 07 (B&B)

Flow To Implementation

Performance Analysis

Current (formulation found for 1 bus case – MILP)

DAC 07 (GP)
Periodic Activation Model

High latency, but allows decoupling the scheduling problem

\[ l(i,j) = \sum_{k:o_k \in P(i,j)} (T_k + r_k) \quad \text{where (approx.)} \]

\[ r_i = C_i + \sum_{j \in hP(i)} \left[ \frac{r_i}{T_j} \right] C_j \]
Event-based Activation Model

Lower latency for high priority paths, jitter increases along the path

\[ l(i,j) = \sum_{k: o_k \in P(i,j)} w_k \]

where (approx.)

\[ w_i = C_i + \sum_{j \in h_p(i)} \left[ \frac{w_i + J_j}{T_j} \right] C_j \]
Activation modes: latency tradeoffs

End to end latency requirements
\[ d_{\text{o}_{14}, \text{o}_{15}} \Rightarrow 70 \]
\[ d_{\text{o}_{16}, \text{o}_{17}} \Rightarrow 100 \]
\[ d_{\text{o}_{18}, \text{o}_{19}} \Rightarrow 120 \]

Mixed activation mode
\[ L_{i,j} = \sum_{k:k=j \land l_{k,q} \in \mathcal{E}_p} (J_k + w_k) + \sum_{q:l_{k,q} \in \mathcal{E}_p} T_q \]

<table>
<thead>
<tr>
<th>Object</th>
<th>( \pi_i )</th>
<th>( T_i )</th>
<th>( C_i )</th>
<th>( r_i )</th>
<th>( l_i )</th>
<th>( J_i )</th>
<th>( w_i )</th>
<th>( r_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_1 )</td>
<td>13</td>
<td>15</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>12</td>
<td>15</td>
<td>2</td>
<td>4</td>
<td>27</td>
<td>8</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>( \tau_3 )</td>
<td>11</td>
<td>15</td>
<td>8</td>
<td>8</td>
<td>50</td>
<td>12</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>( m_4 )</td>
<td>10</td>
<td>15</td>
<td>2</td>
<td>6</td>
<td>71</td>
<td>20</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>( \tau_5 )</td>
<td>9</td>
<td>15</td>
<td>6</td>
<td>6</td>
<td>92</td>
<td>26</td>
<td>6</td>
<td>32</td>
</tr>
<tr>
<td>( \tau_6 )</td>
<td>8</td>
<td>40</td>
<td>6</td>
<td>14</td>
<td>14</td>
<td>0</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>( m_7 )</td>
<td>7</td>
<td>40</td>
<td>2</td>
<td>8</td>
<td>62</td>
<td>30</td>
<td>12</td>
<td>42</td>
</tr>
<tr>
<td>( \tau_8 )</td>
<td>6</td>
<td>40</td>
<td>14</td>
<td>30</td>
<td>122</td>
<td>42</td>
<td>30</td>
<td>72</td>
</tr>
<tr>
<td>( \tau_9 )</td>
<td>5</td>
<td>30</td>
<td>2</td>
<td>42</td>
<td>42</td>
<td>0</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>( m_{10} )</td>
<td>4</td>
<td>30</td>
<td>2</td>
<td>10</td>
<td>82</td>
<td>190</td>
<td>18</td>
<td>208</td>
</tr>
<tr>
<td>( \tau_{11} )</td>
<td>3</td>
<td>30</td>
<td>6</td>
<td>28</td>
<td>140</td>
<td>208</td>
<td>58</td>
<td>266</td>
</tr>
<tr>
<td>( m_{12} )</td>
<td>2</td>
<td>30</td>
<td>2</td>
<td>10</td>
<td>180</td>
<td>266</td>
<td>36</td>
<td>302</td>
</tr>
<tr>
<td>( \tau_{13} )</td>
<td>1</td>
<td>30</td>
<td>8</td>
<td>14</td>
<td>224</td>
<td>302</td>
<td>32</td>
<td>384</td>
</tr>
</tbody>
</table>

Periodic

Event-based
Model Definition

- Selection of the activation event and link groups

An object can be activated by:
- Periodic Timer
- Signal from a single predecessor
- AND composition of signals from a link group
Latencies of OSEK Tasks and CAN Messages

<table>
<thead>
<tr>
<th>Linear Combination</th>
<th>First Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong></td>
<td></td>
</tr>
<tr>
<td>Upper</td>
<td></td>
</tr>
<tr>
<td>$w_i^\uparrow = C_i + \sum_{j \in hp(i)} \left( \frac{w_i^\uparrow + J_j}{C_j} \right) \frac{1}{(q+1)C_i}$</td>
<td></td>
</tr>
<tr>
<td>$w_i(q) = (q+1)C_i \left( \sum_{j \in hp(i)} \frac{C_j + J_j u_j}{(q+1)C_i} \right)$</td>
<td></td>
</tr>
<tr>
<td>$w_i(\alpha) = \frac{1}{C_i} \sum_{j \in hp(i)} \left[ w_i(q) - \frac{w_i(q)}{(q+1)C_i} \right] C_j$</td>
<td></td>
</tr>
<tr>
<td>$w_i = \max q { w_i(q) - q T_j } J_j$</td>
<td></td>
</tr>
<tr>
<td>$r_i J_i = C_i w_i \left( \sum_{j \in hp(i)} \frac{C_j + J_j u_j}{(q+1)C_i} \right)$</td>
<td></td>
</tr>
<tr>
<td>$q = 0...q^<em>, r_i(q^</em>) \leq T_i$</td>
<td></td>
</tr>
<tr>
<td><strong>Lower</strong></td>
<td></td>
</tr>
<tr>
<td>$w_i = C_i + \sum_{j \in hp(i)} \left[ \frac{w_i + J_j}{T_j} \right] C_j$</td>
<td></td>
</tr>
<tr>
<td>$r_i = J_i + w_i$</td>
<td></td>
</tr>
</tbody>
</table>

| **Bus**            |               |
| Upper              |               |
| $l_i(q)q_i^\uparrow B_i B q E_i = \sum_{j \in hp(i)} \left[ \frac{w_i + J_j}{T_j} \right] C_j$ |
| $l_i(q)q_i = \max q \{ C_i + w_i(q) + J_j u_j \}$ |
| $w_i = \max \left\{ C_i + w_i(q) + J_j u_j \right\}$ |
| $q = 0...q^*, r_i(q^*) \leq T_i$ |
| $w_{q_i} = B_i + \sum_{j \in hp(i)} \left[ \frac{w_{q_i} + J_j}{T_j} \right] C_j$ |
| $w_i = w_{q_i} + C_i$ |
| $r_i = J_i + w_i$ |
A linear combination of linear upper and lower bounds can be sufficiently accurate to be used as an estimator of actual e2e latency.
## MILP Solution

### Sets

- $\mathbf{V}$: Set of objects implementing the computation and communication functions
- $\mathbf{E}$: Set of links connecting schedulable objects
- $\mathbf{R}$: Set of resources (CAN, ECUs)

### Parameters

- $\pi_i$: Priority of object $o_i$
- $T_i$: Period of object $o_i$
- $C_i$: Worst-case execution/transmission time of object $o_i$

### Variables

- $r_i$: Worst case response time of object $o_i$
- $J_i$: Release Jitter of object $o_i$
- $w_i$: Worst-case runnable queueing time of object $o_i$
- $L_{s,t}$: End to end latency between object $o_s$ and $o_i$
- $y_{h,k} = \begin{cases} 1, & \text{if activation of } o_k \text{ is event-driven by } o_h \\ 0, & \text{otherwise} \end{cases}$
Feasibility Constraints 1

**Jitter Inheritance Rule**

\[ y_{r,k} = y_{s,k} \]

\[ \sum_{Lg_h \in G(o_k)} y_{r,k} \leq 1 \]

\[ J_k \leq \sum_{Lg_h \in G(o_k)} y_{r,k} \times M \]

\[ 0 \leq J_k \]

\[ J_k \leq r_r + (1 - y_{r,k}) \times M \]

\[ r_r - (1 - y_{r,k}) \times M \leq J_k \]

\[ r_h + (y_{h,k} - 1) \times M \leq J_k \]

\[ J_k \leq r_h \]

\[ J_k \leq y_{h,k} \times M \]

- All links in one group assume the same activation model
- Only one of the incoming link group can provide its activation signal
- If none of incoming groups carry activation signal, then release jitter of object k is 0
- Release jitter inherited from object r which has largest wcrt from the activating group
- Simplified version of link groups
Feasibility Constraints 2

**WCRT Rule**

\[
W_h = r_h C_h + \sum_{k \in hP(h)} \left( \frac{W_h + J_k}{T_k} + \alpha \right) C_k
\]

\[
\sum_{P_r \in P} \left( \alpha \times L^\uparrow_{P_r} + (1 - \alpha) \times L^\downarrow_{P_r} - L_{P_r} \right)^2
\]

*Calculation of worst case response time*

• A linear combination of linear upper and lower bounds is used as an estimation of runnable queuing time

• alpha is chosen to minimize the mean square fit function

**Latency Rule**

\[
w_j \leq z_{i,j}
\]

\[
z_{i,j} \leq w_j + (1 - y_{i,j}) \times M
\]

\[
z_{i,j} \leq w_j + J_j + T_j
\]

\[
w_j + J_j + T_j - y_{i,j} \times M \leq z_{i,j}
\]

\[
L_{s,t} = \sum_{l_{u,v} \in P_{s,t}} z_{u,v}
\]

\[
L_{s,t} \leq d_{s,t}
\]

*Path end to end latency can not exceed deadline*
**Possible Objective Function**

\[
\text{Maximize } \sum_{L \in G} y_{j,k} \\
\text{Minimize } \sum_{P \in P} L_{p_r} \\
\text{Minimize } \sum_{P \in P} \gamma_{p_r} \times \text{Max} (L_{p_r} - d_{p_r}, 0)
\]

- Minimization of the number of event buffers in the system
- Minimization of sum of end to end latencies
- Minimization of sum of weighted deadline violation
Outline

• Automotive architecture trends and challenges
• System-level design methodology and timing evaluation
• Activation models and end-to-end latencies
• Problem definition
  – Example
• MILP Optimization
• Case Study
Experimental vehicle case study

- Functional Architecture
  - Mapping
    - 100 Tasks
    - 322 Messages

- Physical Architecture
  - Using Schedulability theory to set schedulable object activation model

- 6 BUSes
- 38 ECUs

ECU1, ECU2, ..., ECU61, ECU62
**Case study results**

**Before Optimization (all periodic)**
- Worst case = 577ms was found for paths with deadline 300ms
- Worst case = 255.5ms found for paths with deadline 200ms
- Worst case = 145.4ms found for paths with deadline 100ms

**Problem characterization**
- 38 ECUs, 6 Buses
- Bus speed between 25 and 500 kb/s
- Bus utilization between 30% to 50%
- CPU utilization between 5% to 60%
- 100 tasks, 322 messages
- Number of links in the functional dataflow is 507
- 184 Paths analyzed between 10 pairs of functional nodes

**Optimization results**
- A feasible solution is found if using the largest lateness path metric after changing 24 groups
  - 294.8 for paths with d=300
  - 158.1 for paths with d=200
  - 95.46 for paths with d=100 (61.57 average slack)
- The solution was improved with 5 extra branches (76.79 average slack)
- α practically constant = 0.465 with weighted sum of path latencies (evaluating all nodes) no solution found

**Time to solve is**
- 2.6 s for the exact analysis
- 7 s for the linear approx (on a 1.4GHz PC)
Approach

- Mathematical programming
  - Modifying an object period affects multiple paths
  - Additional constraints due to legacy tasks and messages

- Geometric Programming: Poly-time optimization
  - Standard Form:
    \[
    \begin{align*}
    \text{minimize} & \quad f_0(x) \\
    \text{subject to} & \quad f_i(x) \leq 1 \quad i = 1, \ldots, m \\
    & \quad g_i(x) = 1 \quad i = 1, \ldots, p
    \end{align*}
    \]
  - \( x = (x_1, x_2, \ldots, x_n) \) are positive real-valued variables
  - \( g \) is a set of monomial functions
    \[ m(x) = cx_1^{a_1}x_2^{a_2} \ldots x_n^{a_n} \quad c > 0, a_i \in \mathbb{R} \]
  - \( f \) is a set of posynomial functions
    - Sum of monomials

41 41
Geometric programming formulation

- Approximate the response time \( r_i \) with \( s_i \)
  - \( 0 \leq \alpha_i \leq 1 \)
  - If all \( \alpha_i = 1 \), \( s_i \geq r_i \)

\[
s_i = c_i + \sum_{j \in h_p(i)} \left( \frac{s_i}{t_j} + \alpha_i \right) c_j \quad \forall o_i \in T
\]

- Minimize the sum of approx. response times
- Meet end-to-end latency deadlines
- Transformed equations for approx. response times
- Ensure schedulability
- Meet utilization bounds
- Lower and upper bounds for periods
Iterative Procedure to Reduce Error

- Iteratively change $a_i$ based on error
- Parameters
  - maxIt – max. # of iterations
  - errLim – max. permissible error

Start

all $a_i = 1$; ItCount = 0;

ItCount++; $(s, t) = GP(a)$; Calculate $r$;
$e_i = (s_i - r_i)/r_i$;

Max($|e_i|$) < errLim OR ItCount > maxIt

Yes

End

No $a_i = h(a_i, e_i)$
Case Study: Advanced Safety Vehicle

- From GM Research
- E.g. enhanced cruise control, lane departure warning, parallel parking assist

- Architecture
  - 38 ECUs
  - 4 buses

- Functionality
  - 92 tasks
  - 196 messages

- End-to-end latency constraints
  - Over 12 source-sink task pairs
  - 222 total paths
  - Deadlines range from 100ms to 300ms
Experiments

- GP optimization meets all deadlines in 1^{st} iteration
- Solution time: 24s

- Maximum error reduced from 58% to 0.56% in 15 iterations
- Average error (not shown) reduced from 6.98% to 0.009%
Concluding remarks

- Quantitative analysis offers opportunities for architecture exploration and selection
- Domains of cost, dependability and time have been identified as prime candidates
  - not considering, for example, power
- Analysis techniques are at different levels of maturity
- Uncertainty challenge
  - Some required information is typically not available in the early development stages
  - Requirements extraction process is not mature
- Synthesis to be extended to other domains
  - leveraging MILP or GP formulations of the placement, priority assignment and period definition problems
Concluding remarks

- Worst case timing analysis can be applied to design optimization problems
- With respect to end-to-end latencies in distributed architectures there are multiple dimensions that can be explored
  - task allocation
  - period assignment
  - priority assignment
  - ...
- Also, most active safety functions are not truly hard real-time and worst case analysis may be pessimistic
  - end-to-end stochastic analysis
  - design optimizations based on stochastic analysis
Q&A

Thank you!
E-mail: marco@sssup.it