Control-Quality Optimization for Distributed Embedded Systems with Adaptive Fault Tolerance

Soheil Samii\textsuperscript{1,2}, Unmesh D. Bordoloi\textsuperscript{2}, Petru Eles\textsuperscript{2}, Zebo Peng\textsuperscript{2}, Anton Cervin\textsuperscript{3}

\textsuperscript{1}Embedded Intelligent Solutions, Semcon AB
\textsuperscript{2}Dept. of Computer and Information Science, Linköping University
\textsuperscript{3}Dept. of Automatic Control, Lund University

Sweden
Motivation

- Control quality
- Periods, control laws
- Mapping, schedule
Motivation
Motivation

- Plant
- Plant
- Plant
Motivation
Motivation

- Node faults lead to new configurations
- Unpractical to synthesize solutions for all configurations
Overview of our approach

- Classify feasible configurations
  - Sufficient computation capacity
  - Availability of external interfaces to sensors and actuators

- Synthesis of a certain set of base configurations is sufficient to satisfy fault-tolerance requirements

- Design optimization for additional configurations to optimize control quality
Outline

- System model
- Example: Distributed control systems with faults
- Base configurations
- Control-quality optimization
- Experiments
System model

\[
dx(t)/dt = Ax(t) + Bu(t) + v(t) \\
y(t) = Cx(t) + e(t)
\]
Control quality

- Quadratic cost: \( J = E\{ x^T Q_1 x + u^T Q_2 u \} \)

- Depends on
  - the sampling period,
  - the control law, and
  - the mapping and schedule (delays between sampling and actuation)

- ”Jitterbug” (Lund University)
Co-Design Tool for Distributed Control

Mapping, scheduling, synthesis, and optimization tool

Optimize task mapping

Genetic algorithms
Simulation
CLP/ILP
Matlab
Jitterbug

DATE 2009
RTCSA 2011
Example

- Sensors: Node A and C
- Actuators: Node C and D
Configurations

- Sensors: Node A and C
- Actuators: Node C and D
Configurations

- Sensors: Node A and C
- Actuators: Node C and D

Feasible configuration \{A, B, C\}

Task migration
Store tasks
Configurations

- Sensors: Node A and C
- Actuators: Node C and D

Infeasible configuration \{A,B\}

Actuation cannot be done!


- Synthesize mapping, schedule, and control laws for each base configuration

---

Configurations

- \{A, B, C, D\}
- \{A, B, C\}
- \{A, B, D\}
- \{B, C, D\}
- \{A, C, D\}
- \{A, B\}
- \{A, C\}
- \{A, D\}
- \{B, C\}
- \{B, D\}
- \{C, D\}
- \{A\}
- \{B\}
- \{C\}
- \{D\}
Configurations

- \{A, B, C, D\}
- \{A, B, C\}
- \{A, B, D\}
- \{B, C, D\}
- \{A, C, D\}
- \{A, B\}
- \{A, C\}
- \{A, D\}
- \{B, C\}
- \{B, D\}
- \{C, D\}
- \{A\}
- \{B\}
- \{C\}
- \{D\}
Configurations

\{A, B, C, D\} → \{A, B, C\} → \{A, B, D\} → \{B, C, D\} → \{A, C, D\} → \{A, B\} → \{A, C\} → \{A, D\} → \{B, C\} → \{B, D\} → \{C, D\} → \{A\} → \{B\} → \{C\} → \{D\} → \{4.3\} → \{5.4\}
Configurations

{A, B, C, D}

{A, B, C}
{A, B, D}
{B, C, D}
{A, C, D}

{A, B}
{A, C}
{A, D}
{B, C}
{B, D}
{C, D}

{A}
{B}
{C}
{D}
Configurations

\{A, B, C, D\}

\{A, B, C\} \quad 5.4

\{A, B, D\} \quad 4.3

\{B, C, D\} \quad 5.4

\{A, C, D\} \quad 4.3

\{A, B\} \quad 5.4

\{A, C\} \quad 4.3

\{A, D\} \quad 5.4

\{B, C\} \quad 4.3

\{B, D\} \quad 5.4

\{C, D\} \quad 5.4

\{A\} \quad 5.4

\{B\} \quad 5.4

\{C\} \quad 5.4

\{D\} \quad 5.4
Optimization

- Construct solutions for additional configurations (heuristic considers node failure probabilities)
  - Trade-offs: control quality, design time

- Mapping realization (ILP formulation)
  - Task migration (time constraint, overhead)
  - Store tasks on nodes (memory constraint)

- Cost function to minimize: \[ \sum_{C} p_{C} \cdot J_{C} \]
  - \( p_{C} \): Probability of reaching configuration \( C \)
Experiments

![Graph showing relative cost improvement over CPU time for different node counts. The x-axis is labeled 'CPU time [minutes]' and the y-axis is labeled 'Relative cost improvement [%].']

- 5 nodes
- 7 nodes
- 9 nodes
Conclusions

- Faults lead to different configurations
  - Not practical to design a customized solution to each configuration

- Synthesize solutions to a subset of all configurations in order to achieve a level of fault tolerance given by the available sensor/actuator interfaces and capacity of the platform

- Optimization method for control-quality improvements in the configurations