CoBRA - A Generic Architecture for Robust Treatment of Uncertain Information

Werner Brockmann, Andreas Buschermöhle, Jan Hendrik Schoenke

Institute of Computer Science
Smart Embedded Systems Group

Koblenz, 9/16/2013
Motivation

- Unknown, time-variant environment
- Dynamically varying uncertainties
- System behavior not completely predictable
Motivation

• Unknown, time-variant environment
• Dynamically varying uncertainties
• System behavior not completely predictable
Motivation

• Unknown, time-variant environment
• Dynamically varying uncertainties
• System behavior not completely predictable

Observation -> Processed value

True value -> Uncertainty

© BigDog Boston Dynamics
Motivation

- Unknown, time-variant environment
- Dynamically varying uncertainties
- System behavior not completely predictable

Uncertainties propagate throughout architecture
Aim: Best performance in presence of uncertainties
Motivation

- Unknown, time-variant environment
- Dynamically varying uncertainties
- System behavior not completely predictable

Uncertainties propagate throughout architecture

Aim: Best performance in presence of uncertainties

Representation of uncertainty

Architecture for treatment of uncertain information
Overview

State of the Art

CoBRA-Architecture

Example

Conclusion
Uncertainty Representations

Four abstractions of representing uncertainty
Uncertainty Representations

Four abstractions of representing uncertainty

Increasing expressiveness
Uncertainty Representations

Four abstractions of representing uncertainty

- **Scalar Attribute**
- **Set of Possibilities**
- **Weighted Possibilities**
- **Second Order Weighting**

---

Increasing expressiveness

Increasing computational and engineering complexity
Uncertainty Representations

Four abstractions of representing uncertainty

Scalar Attribute

Set of Possibilities

Weighted Possibilities

Second Order Weighting

Increasing expressiveness

Increasing computational and engineering complexity
Uncertainty Treatment Architectures

- **Coarse-grained** → Uncertainty steers module interaction
  - Subsumption architecture \([\text{Brooks}]\)
    - Different behaviors deal implicitly with uncertainties
  - Activation-based behavior control architecture \([\text{Berns, Albiez}]\)
    - Extension to steer the behaviors activities
  - Organic robot control architecture \([\text{Maehle, Brockmann}]\)
    - Directly influence module interaction by health signals
Uncertainty Treatment Architectures

**Coarse-grained** → Uncertainty steers module interaction
- Subsumption architecture [Brooks]
  - Different behaviors deal implicitly with uncertainties
- Activation-based behavior control architecture [Berns, Albiez]
  - Extension to steer the behaviors activities
- Organic robot control architecture [Maehle, Brockmann]
  - Directly influence module interaction by health signals

**Fine-grained** → Uncertainty processing in modules
- Probabilistic robotics [Thrun]
  - Explicit modeling of uncertainties with probabilities
- Pain level [Ferrell]
  - Sensor-based fine-grained status assessment
- Trust management [Brockmann]
  - Explicit uncertainty attribute for (possibly) each information
CoBRA Approach

(Confidence-Backed Real-time Architecture)

- Keep macro-architecture
CoBRA Approach
(Confidence-Backed Real-time Architecture)

- Keep macro-architecture
- *Explicit representation* of uncertainties by scalar *Trust Level* attributes
CoBRA Approach
(Confidence-Backed Real-time Architecture)

- Keep macro-architecture
- *Explicit representation* of uncertainties by scalar *Trust Level* attributes

**Trust Management**
Uncertainty representation with a unified metric of scalar *Trust Signals* $\vartheta \in [0,1]$

A *Trust Signal* expresses the trustworthiness of an information by a *Trust Level*

$\vartheta = 1 \rightarrow$ full information
  no ambiguity
  normal operation
  full trust

$\vartheta = 0 \rightarrow$ no information
  full ambiguity
  failure
  no trust

the value should have full impact on the calculation

the value must not influence the calculation
CoBRA Approach
(Confidence-Backed Real-time Architecture)

• Keep macro-architecture
• Explicit representation of uncertainties by scalar Trust Level attributes
• Three operating modes are distinguished
  • Normal → correct information
  • Critical → reduced performance
  • Catastrophic → endanger safety
• Gradual transition from normal to critical mode
• Sharp distinction to catastrophic mode
CoBRA Approach
(Confidence-Backed Real-time Architecture)

- Keep macro-architecture
- *Explicit representation* of uncertainties by scalar *Trust Level* attributes
- Three operating modes are distinguished
  - Normal → correct information
  - Critical → reduced performance
  - Catastrophic → endanger safety
- *Gradual* transition from normal to critical mode
- *Sharp* distinction to catastrophic mode
  - Incorporate uncertain module *inputs*
  - Reflect *output* uncertainty
CoBRA Micro-architecture

- **Basic Control Unit** → *Fine-grained* extension of modules where necessary
- **Fall-back Control Unit** → *Coarse-grained* in case of catastrophic uncertainty
- **Meta Control Unit** → Blending / selection of output
CoBRA Micro-architecture

- **Basic Control Unit** → *Fine-grained* extension of modules where necessary
- **Fall-back Control Unit** → *Coarse-grained* in case of catastrophic uncertainty
- **Meta Control Unit** → Blending / selection of output
Input Uncertainty at BCU

Problem:

- Garbage in → garbage out
Problem:

- Garbage in $\rightarrow$ garbage out
Input Uncertainty at BCU

Problem:

• Garbage in \(\rightarrow\) garbage out

• Influence of inputs varies
  - per input and
  - depending on other inputs
Input Uncertainty at BCU

Problem:

• Garbage in $\rightarrow$ garbage out
• Influence of inputs varies
  • $\textcircled{$1$}$ per input and
  • $\textcircled{$1$}$ depending on other inputs

Aim:

• Extension for *input uncertainty*

$\Rightarrow$ garbage in $\rightarrow$ maximum certainty out
Problem:
- Garbage in → garbage out
- Influence of inputs varies
  - per input and
  - depending on other inputs

Aim:
- Extension for *input uncertainty*
  - garbage in → maximum certainty out
- *Degree of uncertainty* has to be reflected dynamically at the output
Fine-grained Uncertainty Treatment Approach

Limit the influence of uncertain inputs linearly:

\[
\frac{\partial f^\theta}{\partial x_k} = \theta_{x_k} \frac{\partial f}{\partial x_k} \quad \forall \ k
\]

\[
(x) = \sum_{i=1}^{N} \alpha_i \cdot \phi_i(x)
\]

(E.g.: PID-controller, polynomials, filter, fuzzy systems)
Fine-grained Uncertainty Treatment Approach

\[ i=1 \sum_{i=1}^{N} \alpha_i \cdot \phi_i(x) \]

Limit the influence of uncertain inputs linearly:

\[ \frac{\partial f^\theta}{\partial x_k} = \theta_{x_k} \frac{\partial f}{\partial x_k} \quad \forall \ k \]

(E.g.: PID-controller, polynomials, filter, fuzzy systems)

(E.g.: PID-controller, polynomials, filter, fuzzy systems)
Fine-grained Uncertainty Treatment Approach

\[ i=1 \ N \ \alpha \ i \cdot g \ i \ (x) \ \forall i=1 \ N \ \alpha \ i \cdot g \ i \ (x) \ N \ N \ i=1 \ N \ \alpha \ i \cdot g \ i \ (x) \ \alpha \ i \ \alpha \ \alpha \ \alpha \]

Limit the influence of uncertain inputs linearly:

\[ \frac{\partial f^g}{\partial x_k} = g_{x_k} \frac{\partial f}{\partial x_k} \ \forall \ k \]

(E.g.: PID-controller, polynomials, filter, fuzzy systems)

(E.g.: PID-controller, polynomials, filter, fuzzy systems)

\[ f^g(x) = \sum_{i=1}^{N} \alpha_i \cdot \prod_{k=1}^{M} (g_{x_k} \phi_{i,k}(x_k) + (1 - g_{x_k})c_{i,k}) \]
Fine-grained Uncertainty Treatment Approach

\[ i=1 \ N \ \alpha \ i \cdot \phi \ i \ (x) \ ii=1 i=1 N \ \alpha \ i \cdot \phi \ i \ (x) \ ii=1 i=1 N \ \alpha \ i \cdot \phi \ i \ (x) \ \alpha \ i \ \alpha \ \alpha \ \alpha \ i \] i ii \ \alpha \ i \cdot \phi \ i \ \phi \ \phi \ \phi \ i \ (x) \ \alpha \ i \cdot \phi \ i \ (x) \]

Limit the influence of uncertain inputs linearly:

\[ \frac{\partial f^\theta}{\partial x_k} = \vartheta_{x_k} \frac{\partial f}{\partial x_k} \ \forall \ k \]

(E.g.: PID-controller, polynomials, filter, fuzzy systems)

\[ \sigma^2 (x, \theta) = \sum_{i,j=1}^{\alpha} \alpha \ \phi_{i,j} \left( \prod_{k=1}^{n} \left( \phi_{x_k} \phi_{u_k} \phi_{j,k}(u) \phi_{j,k}(u) \phi_{j,k}(u) \right) + (1 - \vartheta_{x_k}) \frac{\int_{x_k}^{\bar{x}_k} \phi_{i,k}(u) \phi_{j,k}(u) \ du}{\bar{x}_k - x_k} \right) \]

\[ - \prod_{k=1}^{M} \left( \vartheta_{x_k} \phi_{i,k}(x_k) \phi_{j,k}(x_k) + (1 - \vartheta_{x_k}) \right) \frac{\int_{x_k}^{\bar{x}_k} \phi_{i,k}(u) \ du}{\bar{x}_k - x_k} \]

\[ \sigma^2 (x, \theta) \rightarrow \vartheta_{out} \]
Exemplary Application

• Contact strength of a robot leg depending on its load and acceleration
Exemplary Application

- Contact strength of a robot leg depending on its load and acceleration

\[ \vartheta_{Acc} : 1.0 \]
Exemplary Application

- Contact strength of a robot leg depending on its load and acceleration
Exemplary Application

- Contact strength of a robot leg depending on its load and acceleration

\[ \vartheta_{Acc} : \{1.0, 0.66, 0.33\} \]
Exemplary Application

- Contact strength of a robot leg depending on its load and acceleration

\[ v_{Acc} : 1.0 \quad 0.66 \quad 0.33 \quad 0.0 \]
Exemplary Application

- Contact strength of a robot leg depending on its load and acceleration

\[ \theta_{Acc} : \begin{array}{cccc}
1.0 & 0.66 & 0.33 & 0.0 \\
\end{array} \]
Exemplary Application

- Contact strength of a robot leg depending on its load and acceleration
Conclusion

CoBRA: Confidence-Backed Real-time Architecture

- Architecture for explicit uncertainty treatment
  - Coarse- and fine-grained uncertainty treatment
  - Distinction of normal, critical and catastrophic operation
  - Easy to engineer and migrate from existing architectures
Conclusion

**CoBRA: Confidence-Backed Real-time Architecture**

- Architecture for explicit uncertainty treatment
  - Coarse- and fine-grained uncertainty treatment
  - Distinction of *normal*, *critical* and *catastrophic* operation
  - Easy to engineer and migrate from existing architectures

- Algorithms for fine-grained uncertainty processing
  - Increased certainty of output
  - Dynamically depending on inputs and input uncertainties
  - Degree of uncertainty reflected at output
  - Low computational overhead
Conclusion

**CoBRA: Confidence-Backed Real-time Architecture**

- Architecture for explicit uncertainty treatment
  - Coarse- and fine-grained uncertainty treatment
  - Distinction of *normal*, *critical* and *catastrophic* operation
  - Easy to engineer and migrate from existing architectures

- Algorithms for fine-grained uncertainty processing
  - Increased certainty of output
  - Dynamically depending on inputs and input uncertainties
  - Degree of uncertainty reflected at output
  - Low computational overhead

→ Simple architecture to deal with uncertainties in embedded systems