TECHNISCHE
UNIVERSITÄT
WIEN
Vienna | Austria

Institut für
Computertechnik
Institute of
Computer Technology

Self-Awareness in Systems on Chip and Cyber-Physical Systems

Axel Jantsch

TU Wien, Vienna, Austria

September 11, 2019

TU Wien

- Faculty of Electrical Engineering and Information Technology
  - Institute of Computer Technology
    - Axel Jantsch

TU Wien

- Faculty of Electrical Engineering and Information Technology
  - Institute of Computer Technology
    - Axel Jantsch

Acknowledgment

Axel Jantsch, David Juhasz, Hedyeh Kholerdi, Amir Rahmani, Nima Taherinejad, Edwin Willegger

TU Wien, Vienna, Austria

Nikil Dutt, Kasra Moazzemi, Amir Rahmani

UC Irvine, California

Anil Kanduri, Arman Anzanpour, Elham Shamsa, Iman Azimi, Maximilian Götzinger, Pasi Liljeberg

University of Turku, Finland

Outline

Reading List I

Until Thursday, 5 Sep

- Nikil Dutt, Axel Jantsch, and Santanu Sarma. “Self-Aware Cyber-Physical Systems-on-Chip”. In: *Proceedings of the International Conference for Computer Aided Design*. invited. Austin, Texas, USA, Nov. 2015
- Nima TaheriNejad, Axel Jantsch, and David Pollreisz. “Comprehensive Observation and its Role in Self-Awareness - An Emotion Recognition System Example”. In: *Proceedings of the Federated Conference on Computer Science and Information Systems*. Gdansk, Poland, Sept. 2016

Reading List II

Until Tuesday, 10 Sep

- Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Leonardi. “Experiencing SAX: a Novel Symbolic Representation of Time Series”. In: *Data Mining and Knowledge Discovery* 15.2 (Oct. 2007), pp. 107–144
- Maximilian Götzinger et al. “Model-free Condition Monitoring with Confidence”. In: *International Journal of Computer Integrated Manufacturing* 32.4-5 (2019)

Reading List III

Until Thursday, 12 Sep

- Nima TaheriNejad and Axel Jantsch. “Improved Machine Learning using Confidence”. In: *IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. Edmonton, Canada, May 2019
- Elham Shamsa, Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Axel Jantsch, and Nikil Dutt. “Goal-Driven Autonomy for Efficient On-chip Resource Management: Transforming Objectives to Goals”. In: *Proceedings of the Design and Test Europe Conference (DATE)*. Florence, Italy, Mar. 2019

General Questions for Work on Self-Awareness

- What does SA accomplish?
- What are the alternatives?
- What are the benefits?
- How does the paper quantify them?

Reading List - Papers 1 & 2

Until Thursday, 5 Sep

- Nikil Dutt, Axel Jantsch, and Santanu Sarma. “Self-Aware Cyber-Physical Systems-on-Chip”. In: *Proceedings of the International Conference for Computer Aided Design*. invited. Austin, Texas, USA, Nov. 2015
- Nima TaheriNejad, Axel Jantsch, and David Pollreisz. “Comprehensive Observation and its Role in Self-Awareness - An Emotion Recognition System Example”. In: *Proceedings of the Federated Conference on Computer Science and Information Systems*. Gdansk, Poland, Sept. 2016

Reading List - Paper 1

Self-Aware Cyber-Physical Systems-on-Chip, N. Dutt et al., ICCAD 2015.

- To which degree are individual SA features realized?
- What are their benefits?
- How does the paper quantify them?

Reading List - Paper 2

Comprehensive Observation and its Role in Self-Awareness, N. TaheriNejad et al., FedCSIS 2016.

- What is the difference between *data reliability* and *confidence*?
- Do you agree with the list of elements as part of *observation*? Anything missing?
- Do you agree with the list of challenges and offered solutions?
- What are the alternatives and what are the benefits?
- Are the benefits quantified?

Reading List - Papers 3 & 4

Until Thursday, 10 Sep

- Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Leonardi. "Experiencing SAX: a Novel Symbolic Representation of Time Series". In: *Data Mining and Knowledge Discovery* 15.2 (Oct. 2007), pp. 107–144
- Maximilian Götzinger et al. "Model-free Condition Monitoring with Confidence". In: *International Journal of Computer Integrated Manufacturing* 32.4-5 (2019)

Reading List - Paper 3

Experiencing SAX, J. Lin et al., DMKD 2007.

- What are the advantages of SAX over alternative representations of time series?
- What is normalization of a time series and why is it required?
- Why should the distance between two symbolic representations be a lower bound of the distance of the original time series?
- How would you generalize SAX to multiple time series?

Reading List - Paper 4

Model-free Condition Monitoring with Confidence, M. Götzinger et al., IJCIM 2019.

- What are the pros and cons of using model-free vs. model based monitoring?
- What is the effect of fuzzy logic in CCAM?
- How would you improve CCAM?

Reading List - Papers 5 & 6

Until Thursday, 12 Sep

- Nima TaheriNejad and Axel Jantsch. “Improved Machine Learning using Confidence”. In: *IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. Edmonton, Canada, May 2019
- Elham Shamsa, Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Axel Jantsch, and Nikil Dutt. “Goal-Driven Autonomy for Efficient On-chip Resource Management: Transforming Objectives to Goals”. In: *Proceedings of the Design and Test Europe Conference (DATE)*. Florence, Italy, Mar. 2019

Reading List - Paper 5

Improved Machine Learning using Confidence, N. TaheriNejad et al., CCECE 2019.

- What are characteristic properties for a probability metric?
- What are characteristic properties for a distance metric?
- Which metric is preferable for confidence: distance or probability?

Reading List - Paper 6

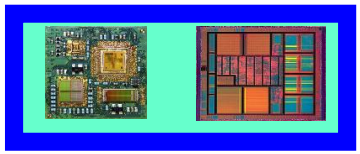
Goal-Driven Autonomy, E. Shamsa et al., DATE 2019.

- What is the difference between reward and urgency?
- Can you think of goals that cannot be expressed in the GDA framework?
- Propose a mechanism to combine conflicting goals, e.g. $P \rightarrow \min$ and $P \rightarrow \max$.

Motivation

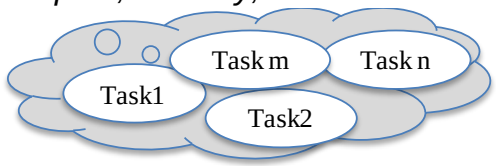
The Problem

- Large number of resources
- Many tight constraints
- Varying application demands, both within and between applications;
- Functional Aberrations:
  - Design errors or omissions;
  - Malicious attacks;
  - Aging;
  - Soft errors;
- Non-functional Aberrations:
  - Performance;
  - Power consumption;

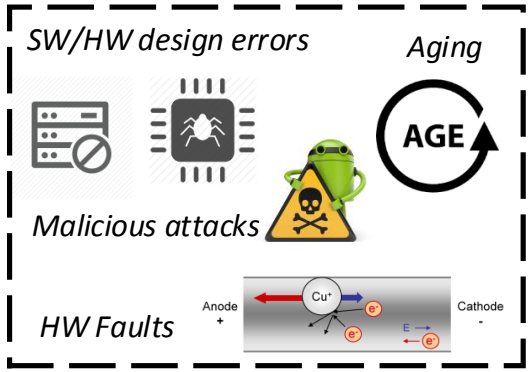


Varying Application and User Demands

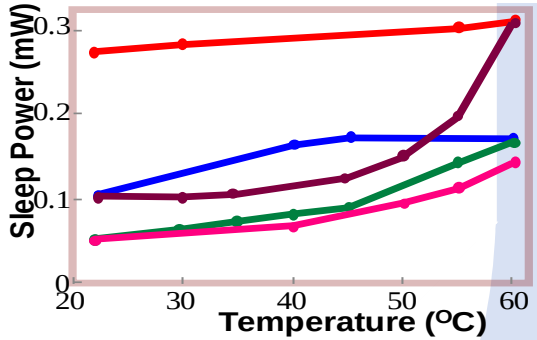
workload phasic behavior
user inputs
varying compute, memory, and communication



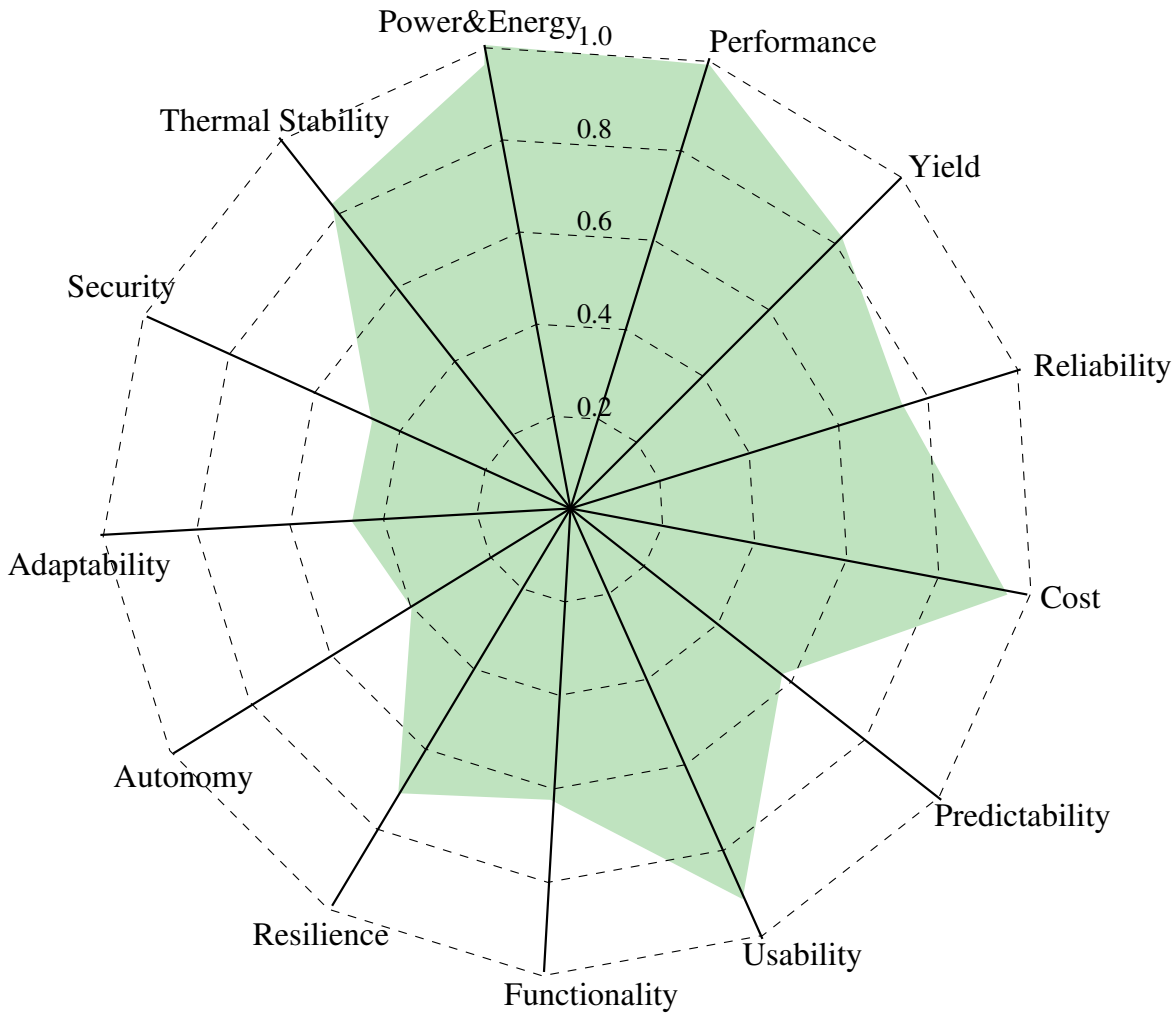
Functional Aberrations



Non-functional Aberrations

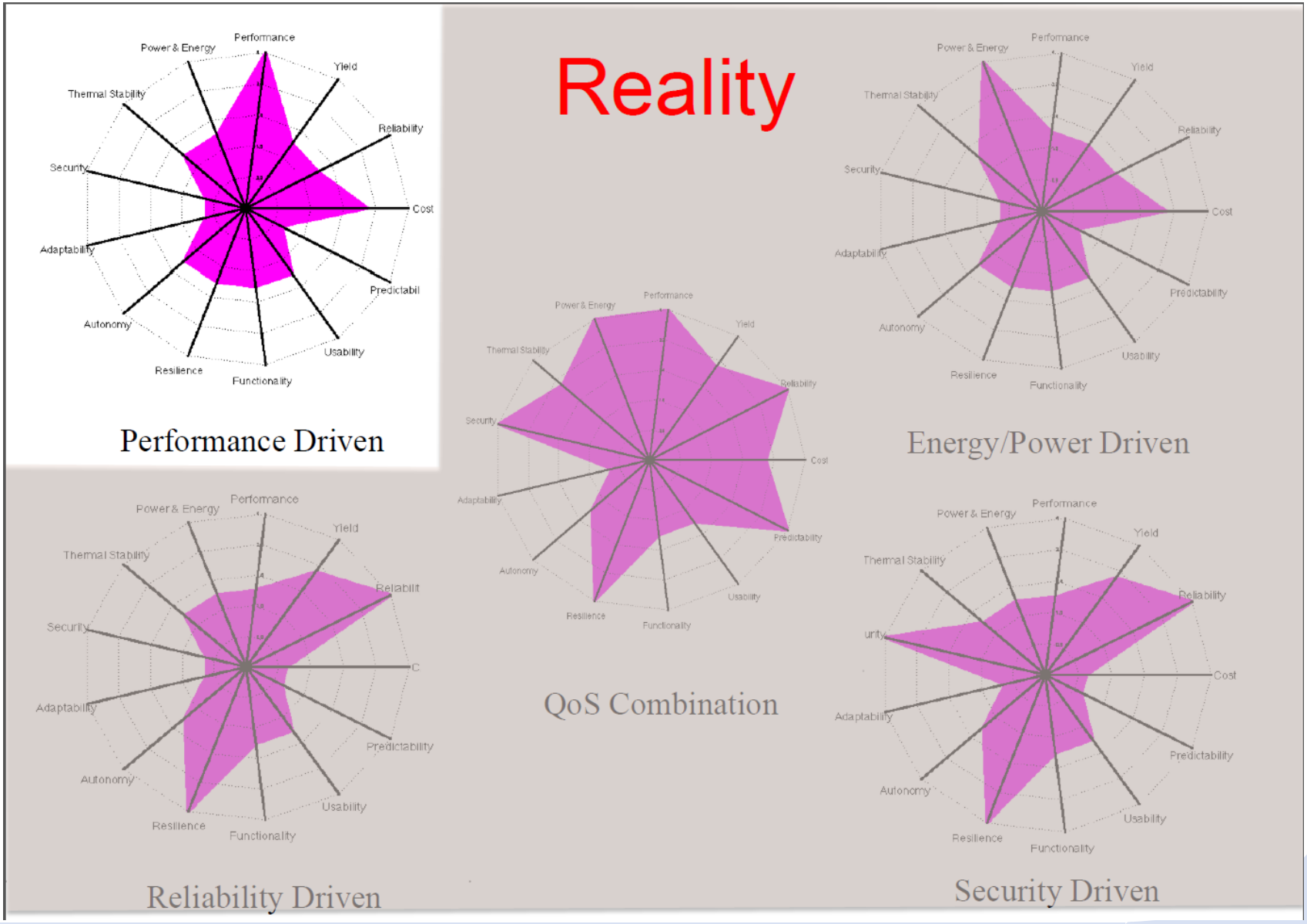


The SoC Radar

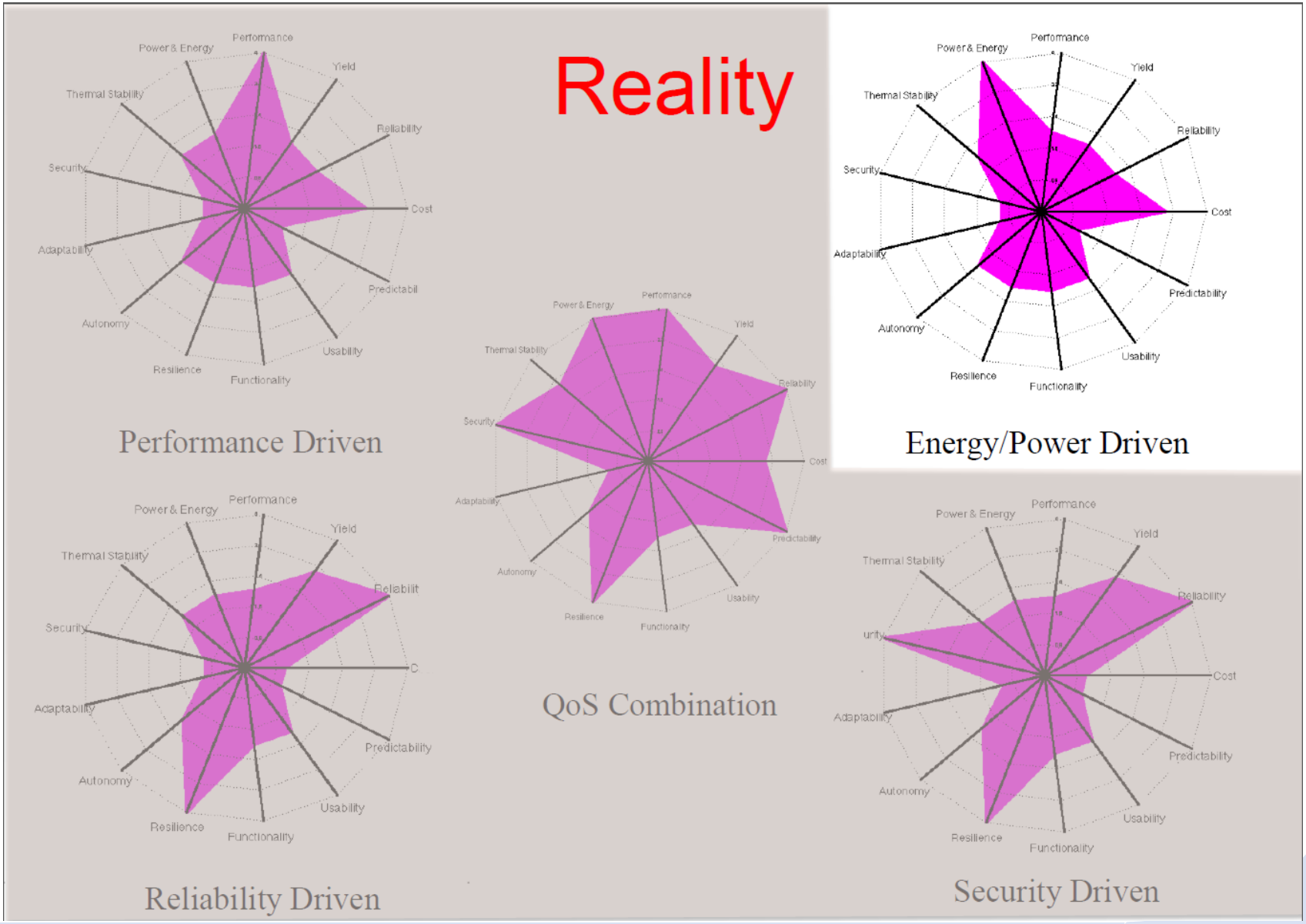


Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. "On-Chip Self-Awareness Using Cyberphysical-Systems-On-Chip (CPSoC)". In: *Proceedings of the 12th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. New Delhi, India, Oct. 2014

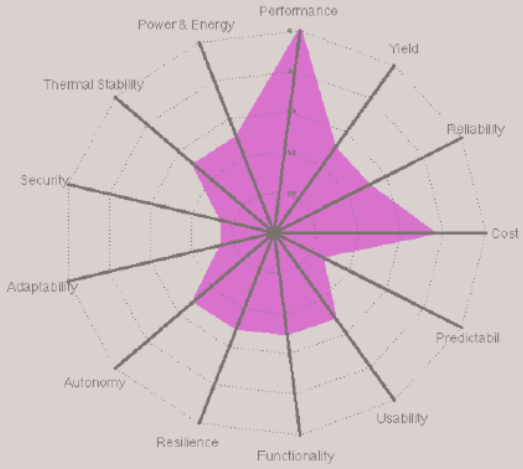
The SoC Radar



The SoC Radar

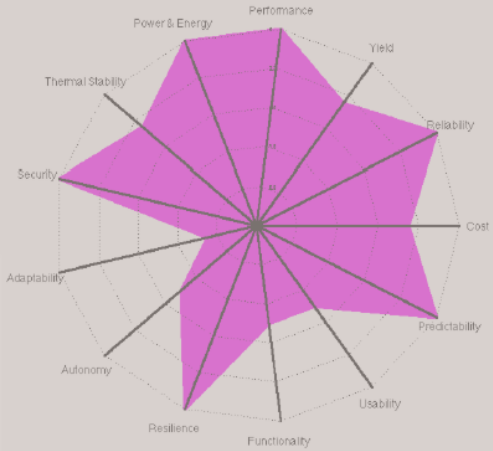


The SoC Radar

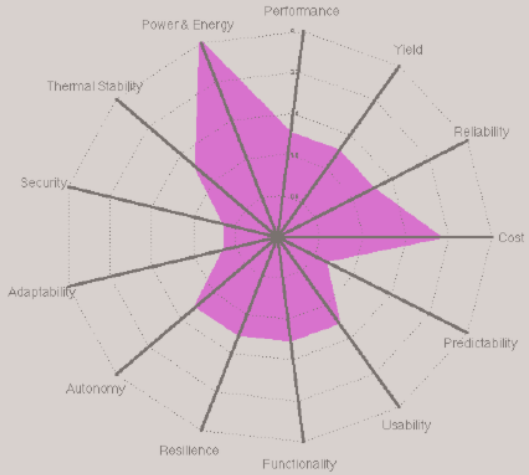


Performance Driven

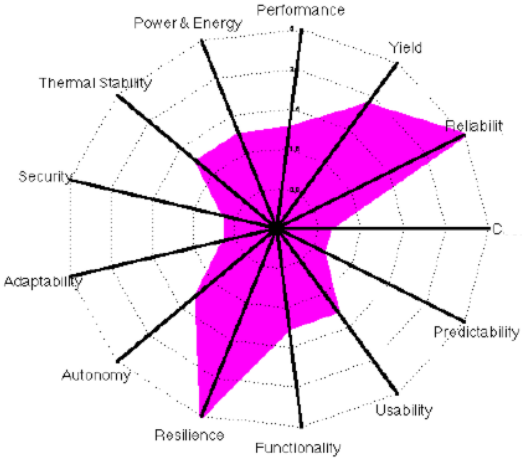
Reality



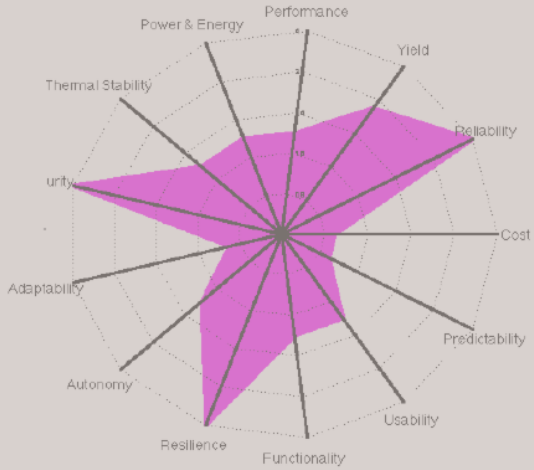
QoS Combination



Energy/Power Driven

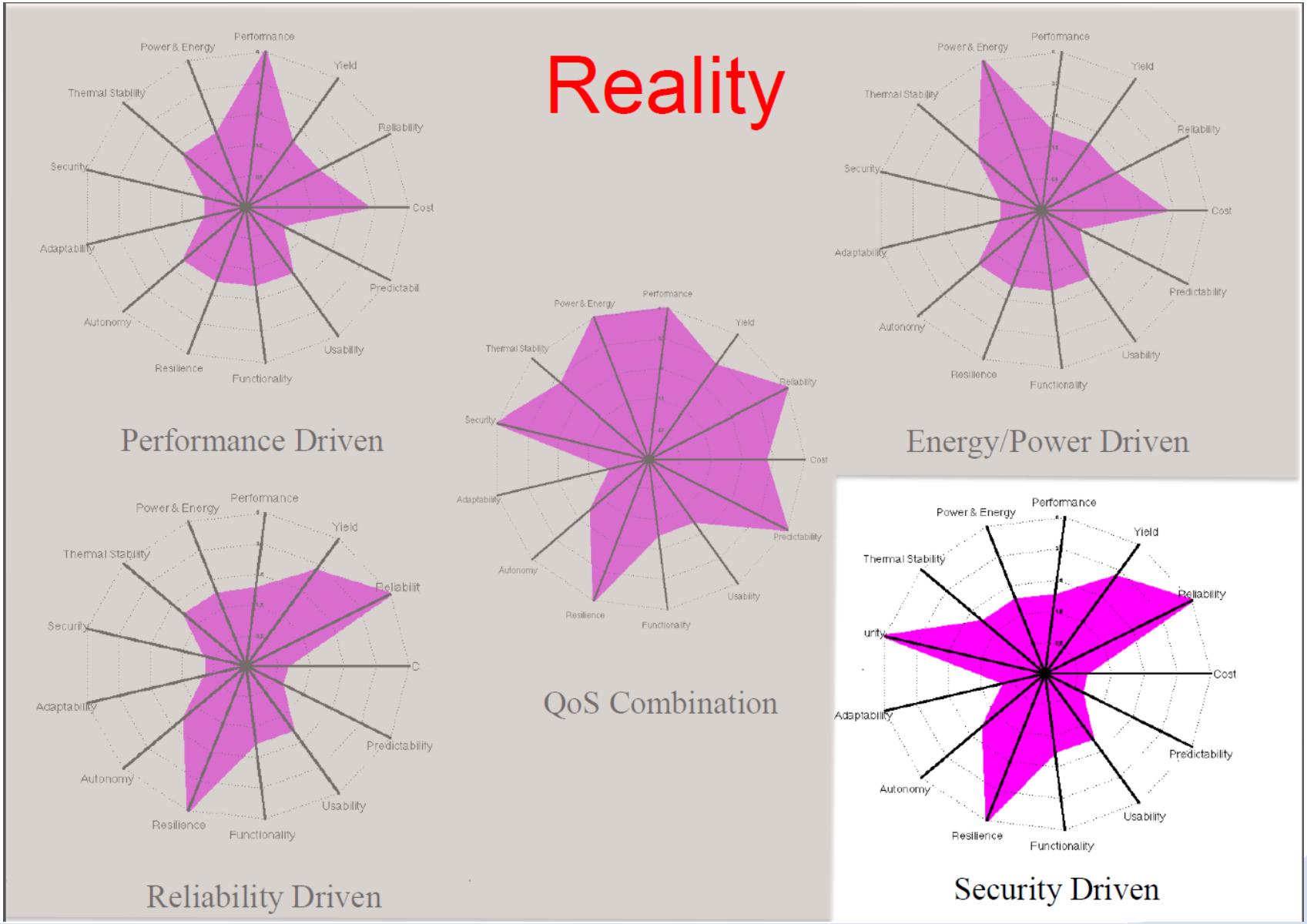


Reliability Driven

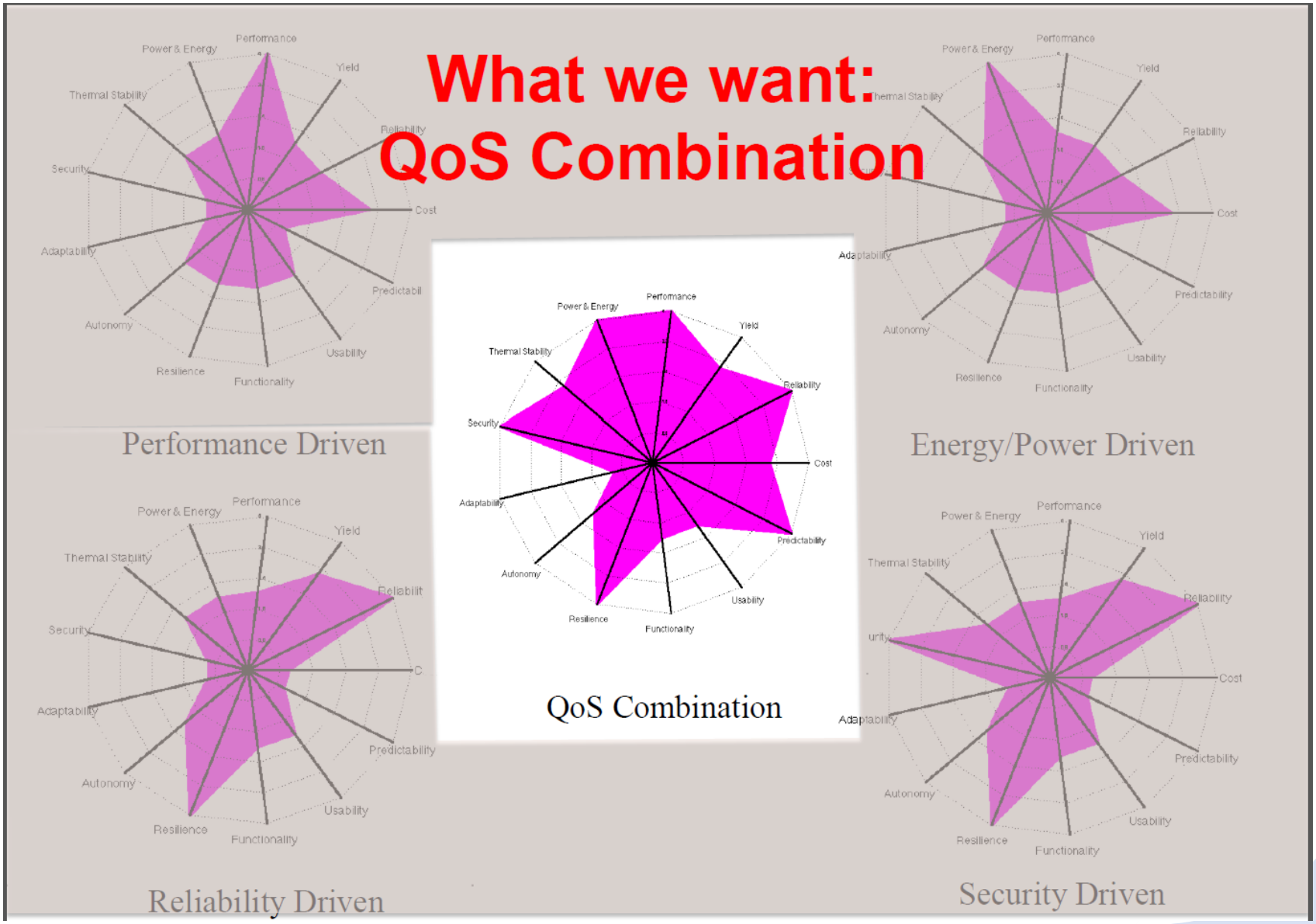


Security Driven

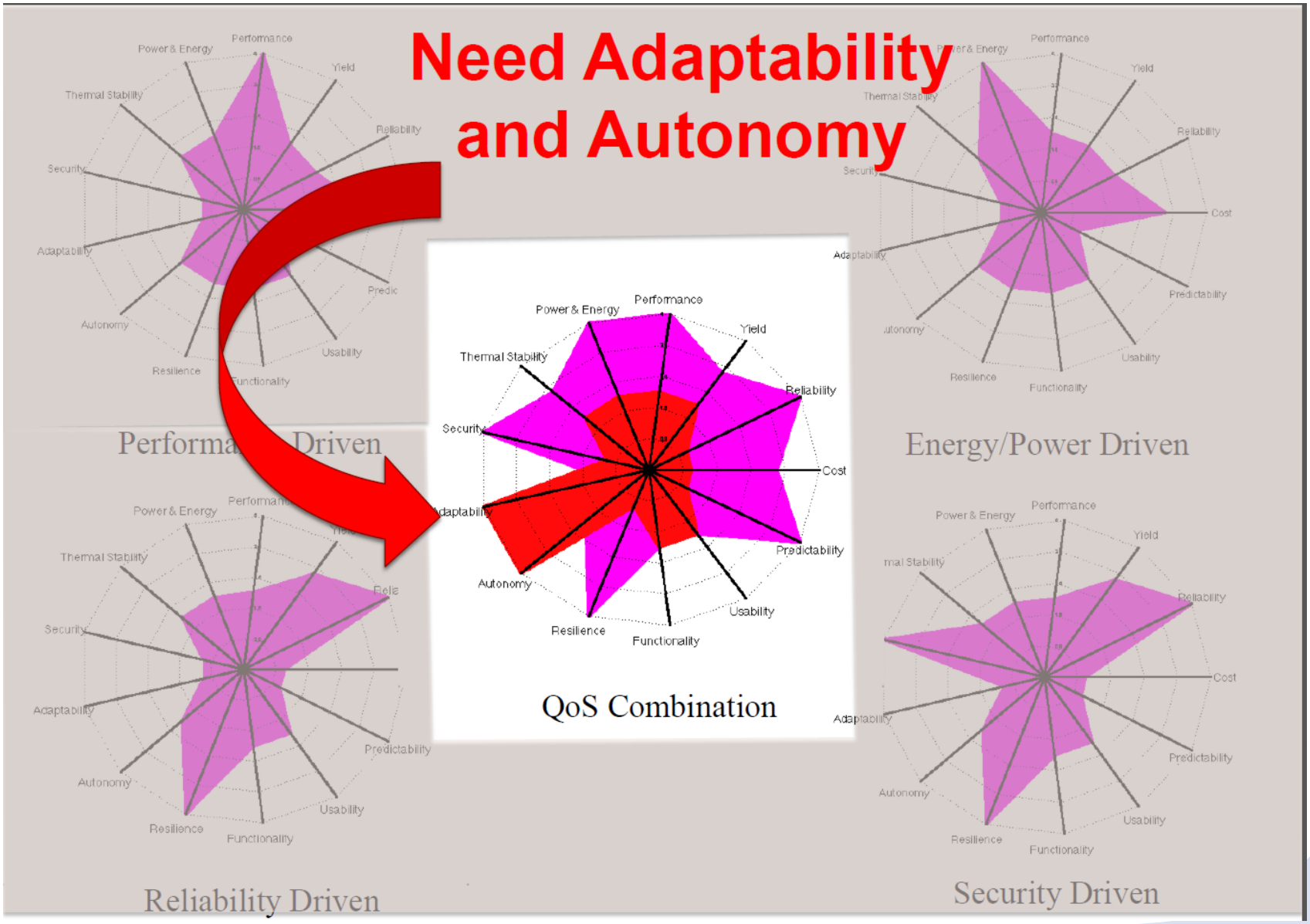
The SoC Radar



The SoC Radar



The SoC Radar



Autonomy and Adaptivity

Adaptivity is the ability to effect run-time changes and handle unexpected events.

Autonomy is the ability to operate independently, without external control.

Necessary

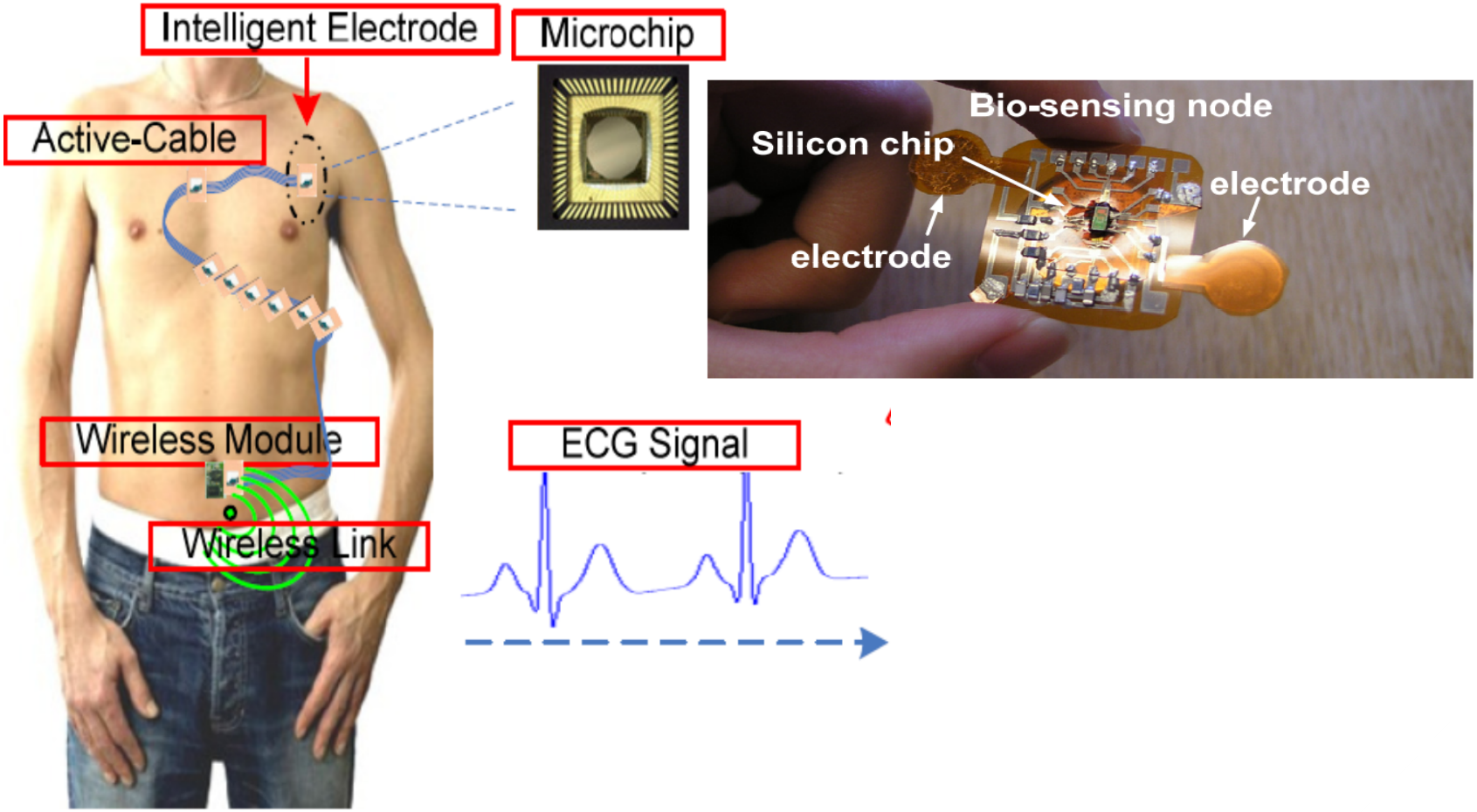
- When requirements, objectives, environments change dynamically during operation; and
- When these changes are unknown at design time.

Concepts of Self-Awareness

Which Ingredients Lead to Awareness ?

- Data abstraction
- Disambiguation
- Desirability scale
- History
- Goals
- Attention
- Learning
- Introspection

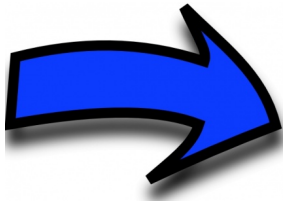
Awareness for Resource Constrained, Insect-like Gadgets

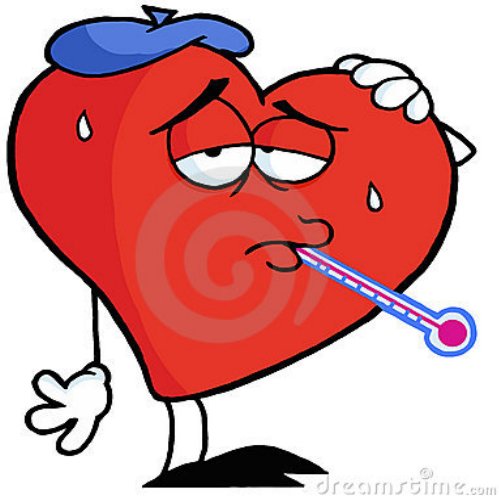


Courtesy of KTH

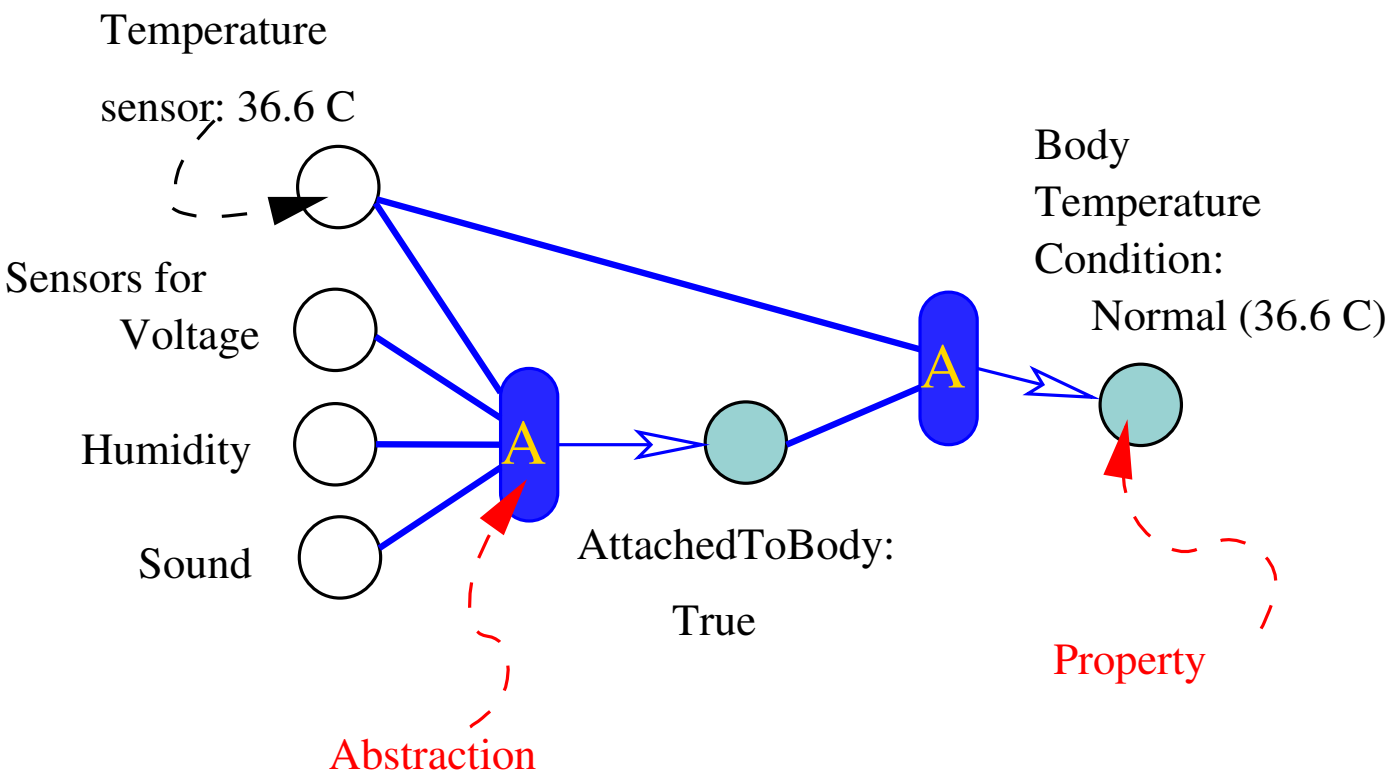
Abstractions and Models

Abstraction: Mapping of Measurements $\Rightarrow$ Properties



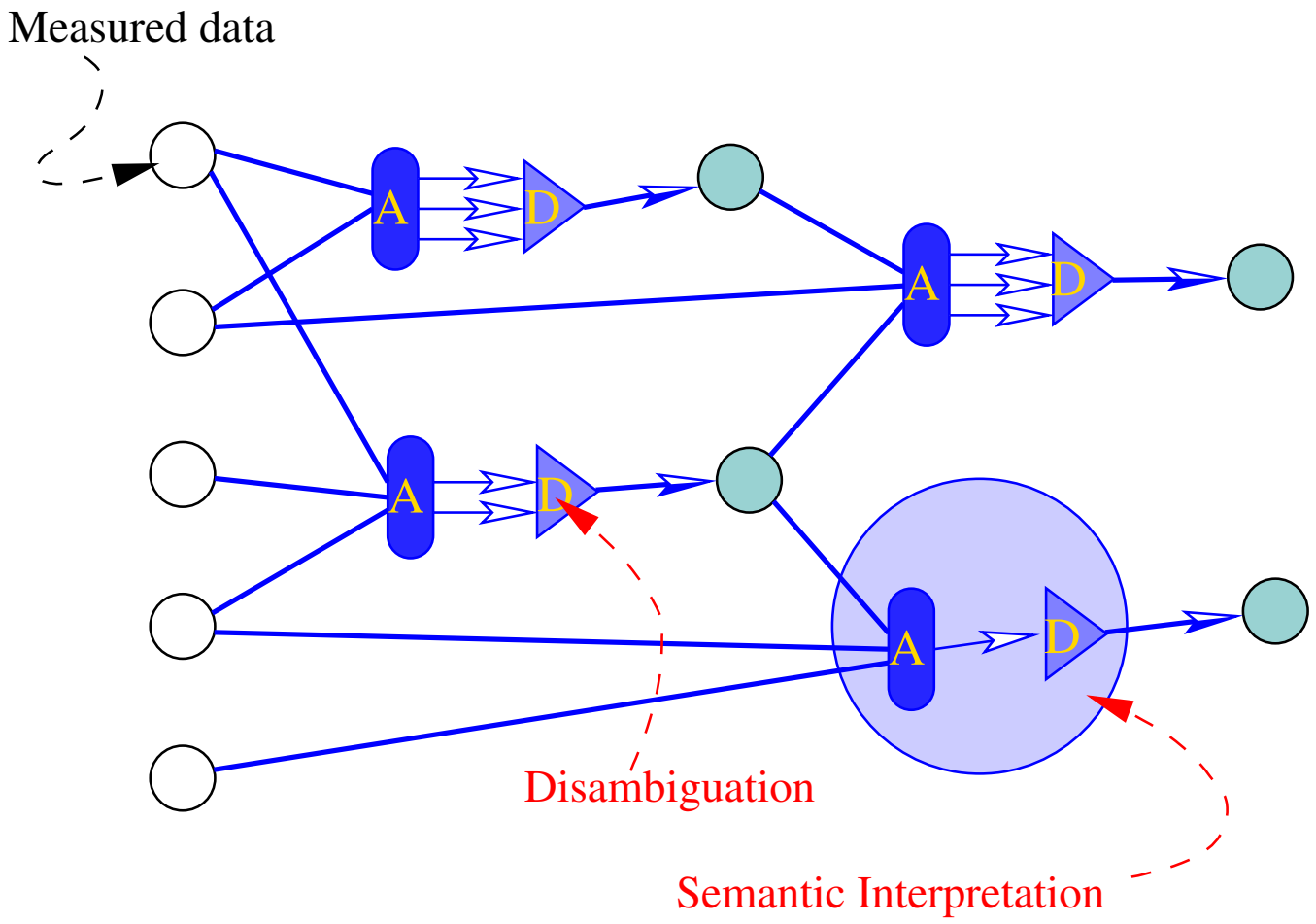


Abstractions and Models



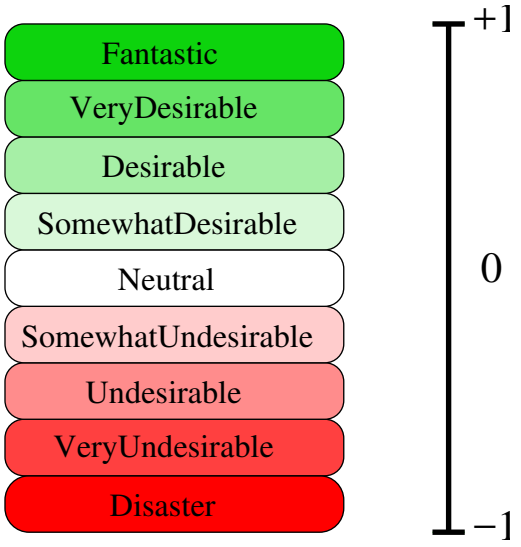
Disambiguation

Selection among several interpretations



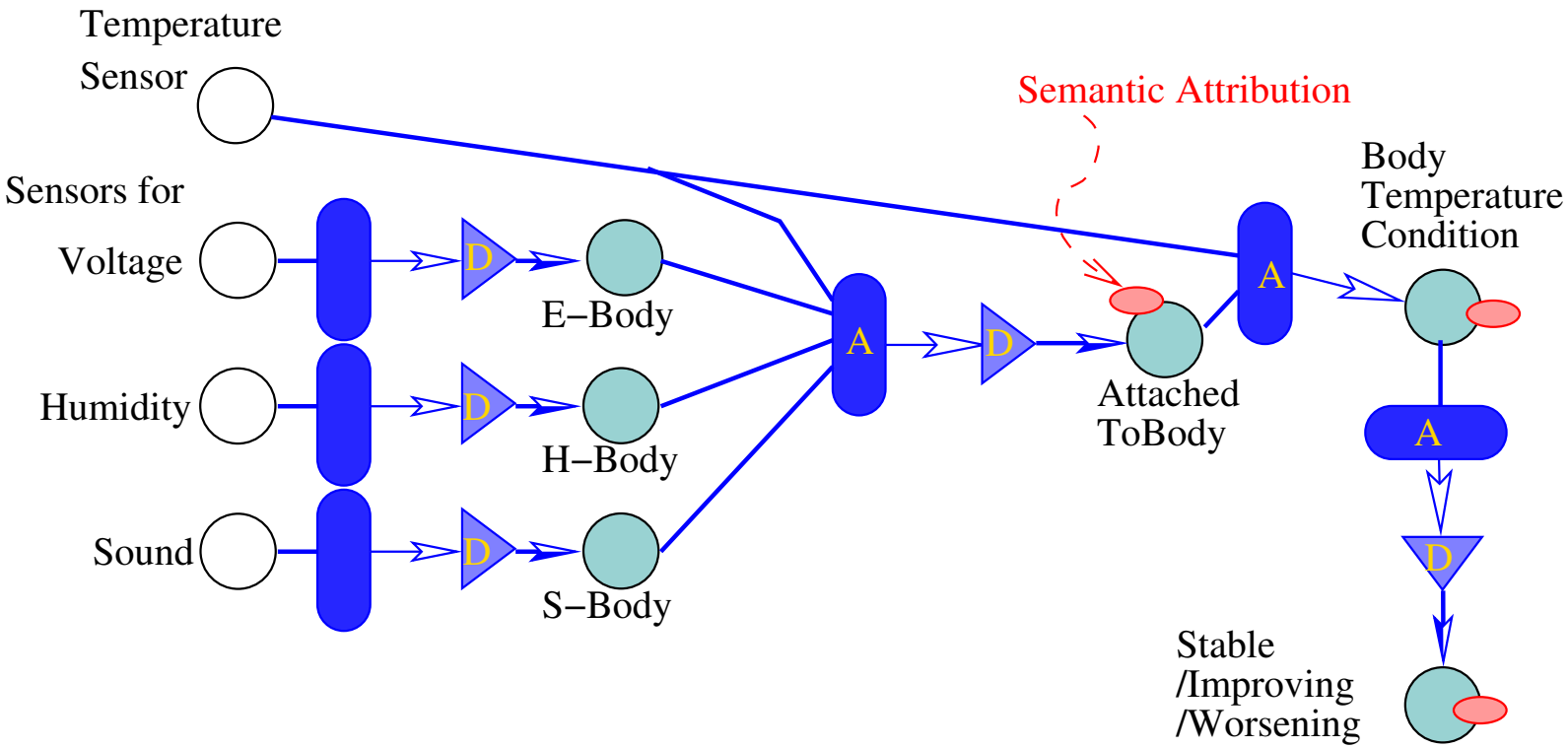
Desirability Scale

Desirability is the common, internal currency.

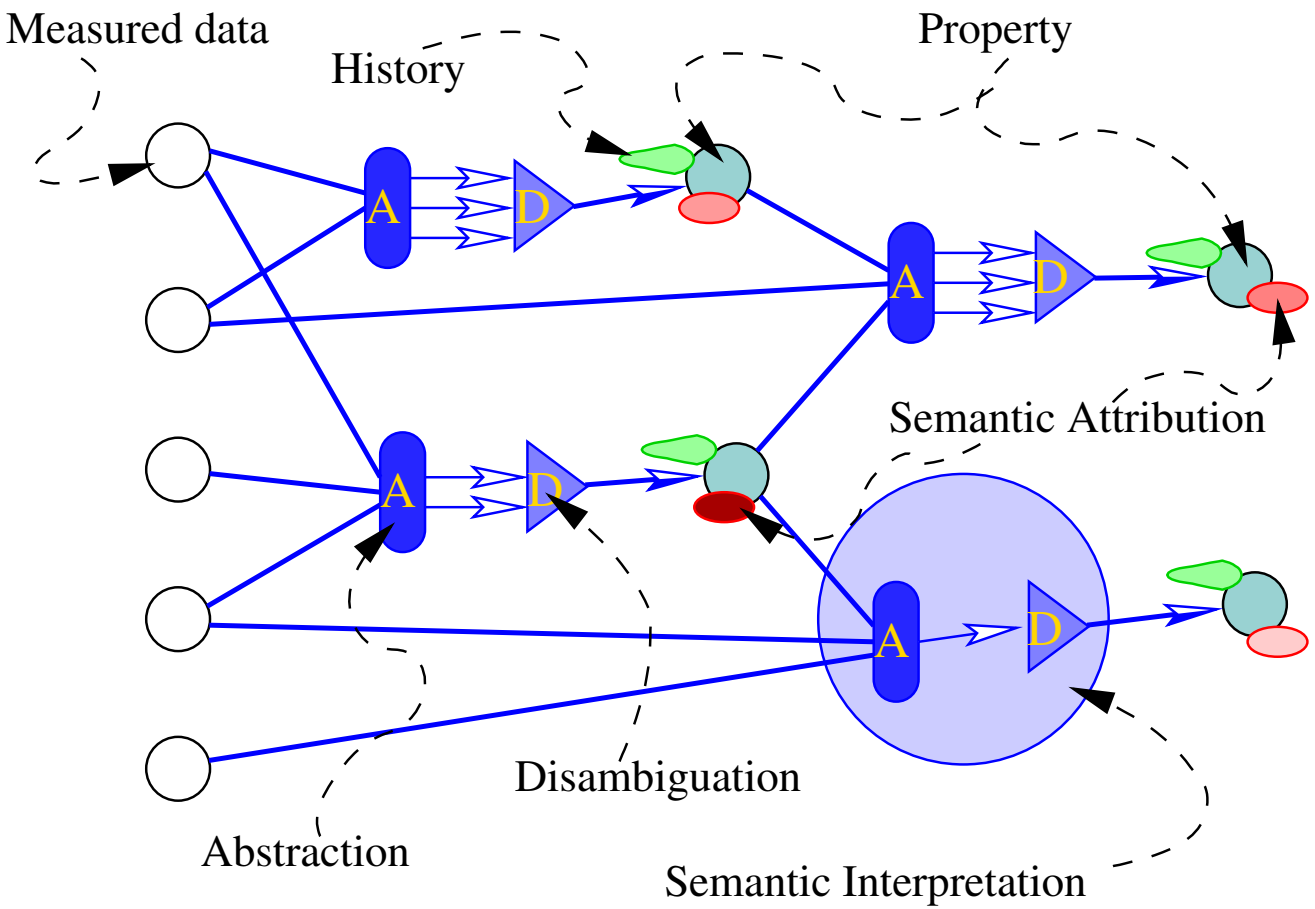


Semantic Attribution maps the values of a property to a point in the desirability scale.

BioPatch with Semantic Attribution



Semantic Interpretation



History

History of a Property The evolution of the values of a property.

Abstracted History The history stores abstracted values.

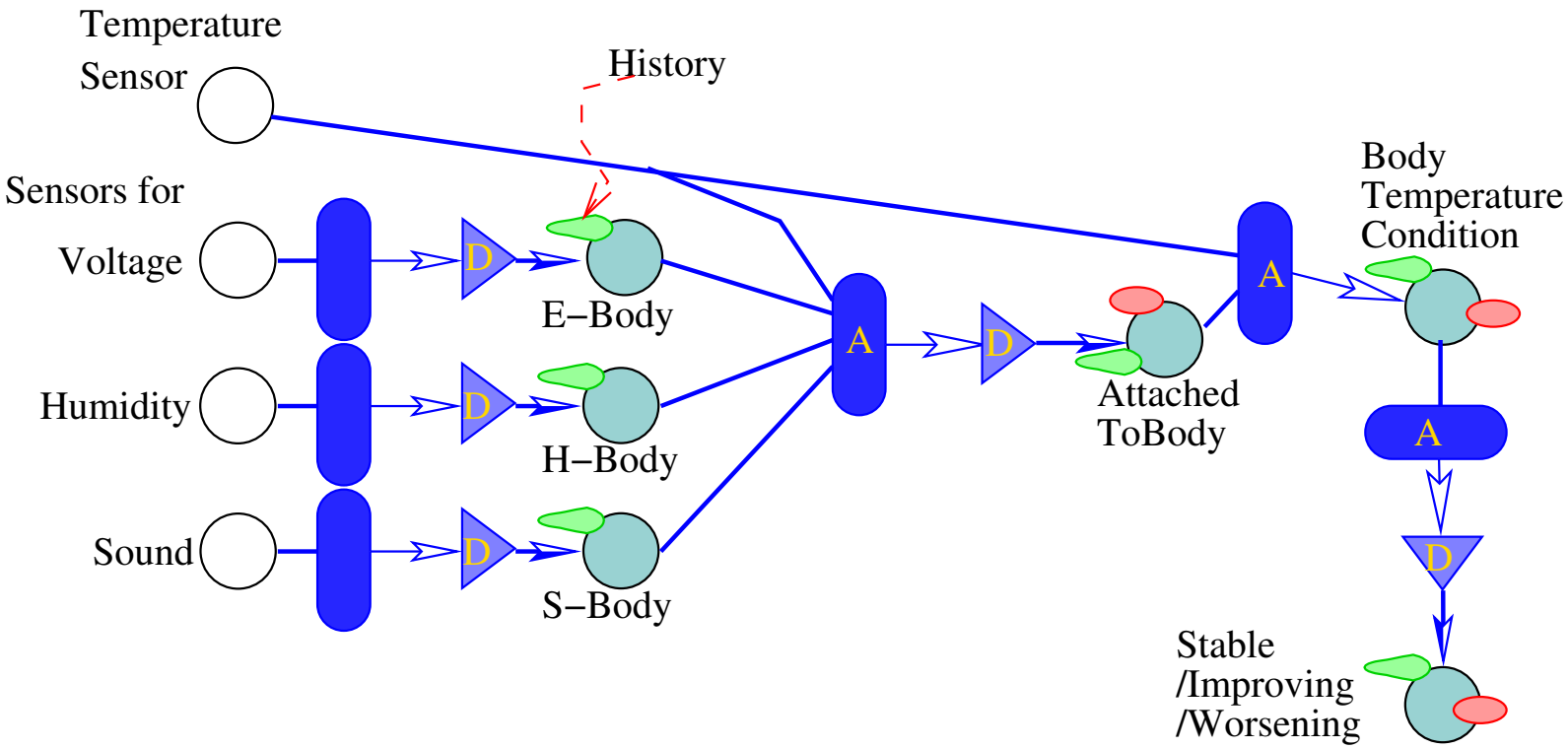
Attributed History The history is annotated with attributions.

Fading History If the property values are more abstracted the longer ago they have occurred.

Consolidating History Relevant memories are enforced, irrelevant memories are cleaned.

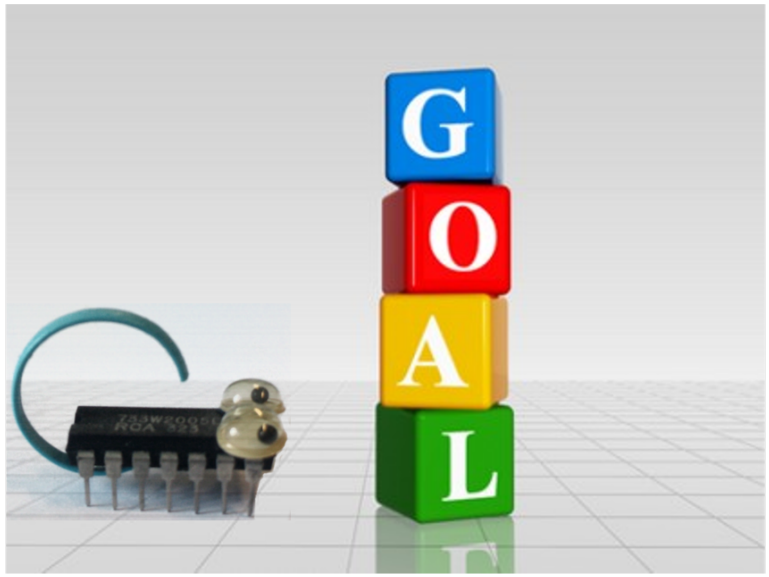
Evolving History Memories are adjusted to fit later observations.

BioPatch with History

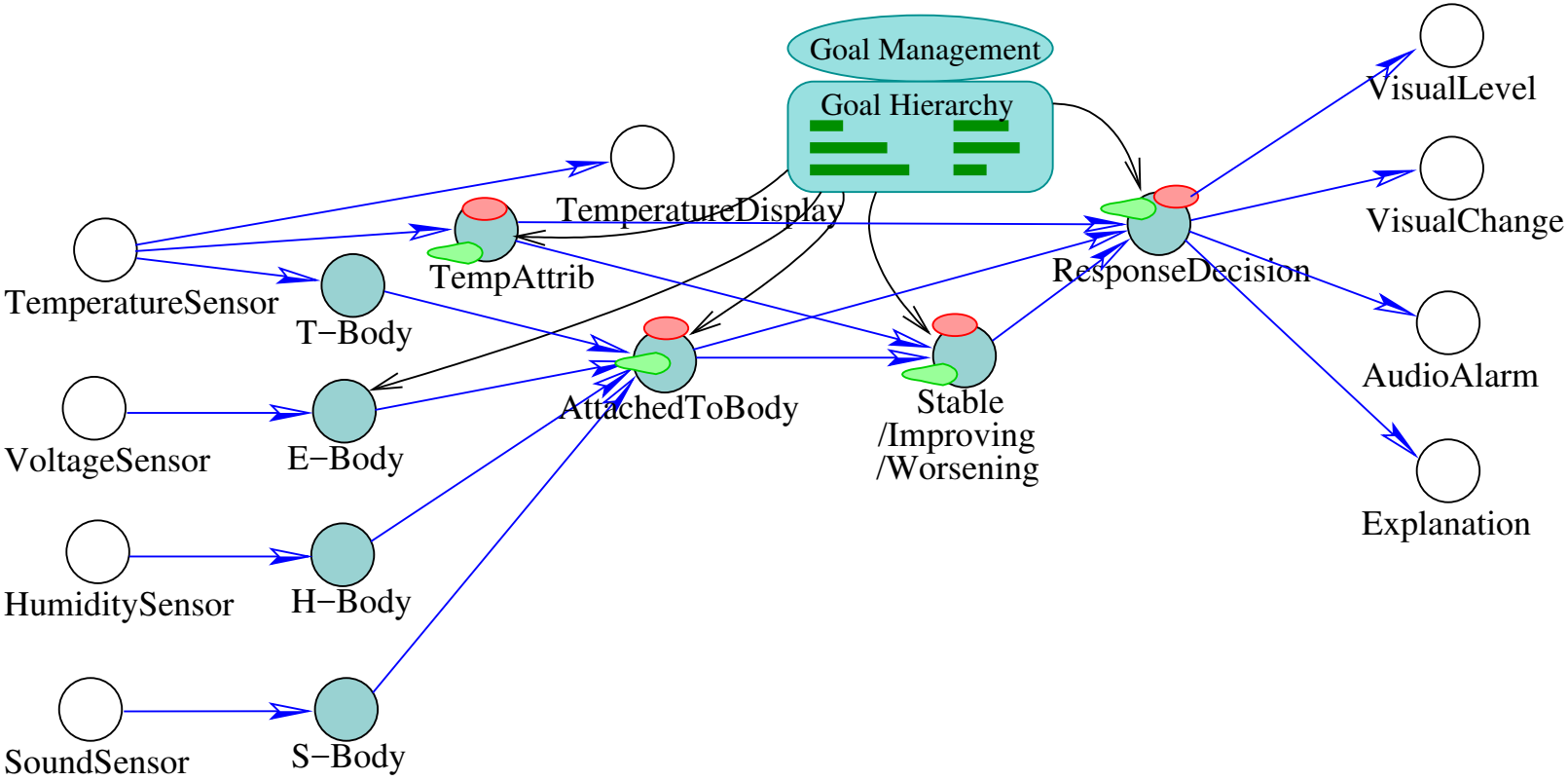


Expectations and Goals

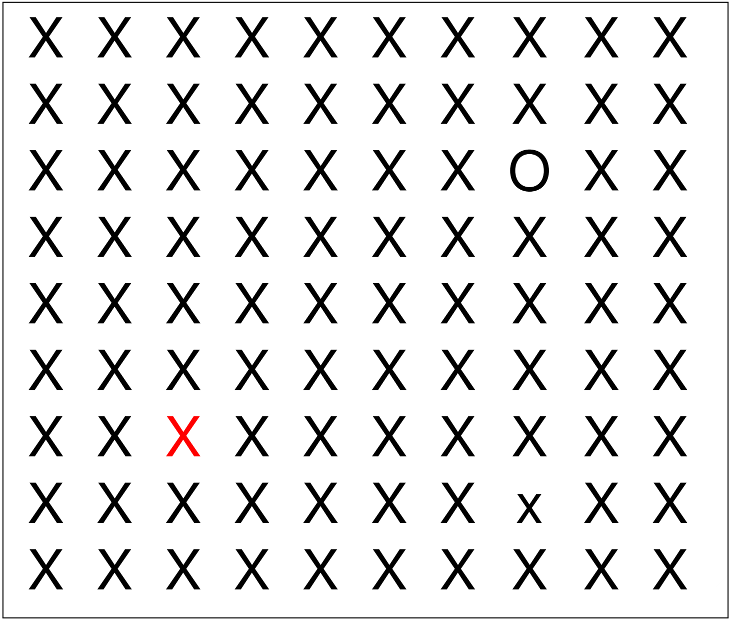
- Expectations on Environment
- Expectations on Subject
- Sub-Goals
- Goals
- Purpose



Acting BioPatch



Attention



Attention


H. Rousseau

Attention

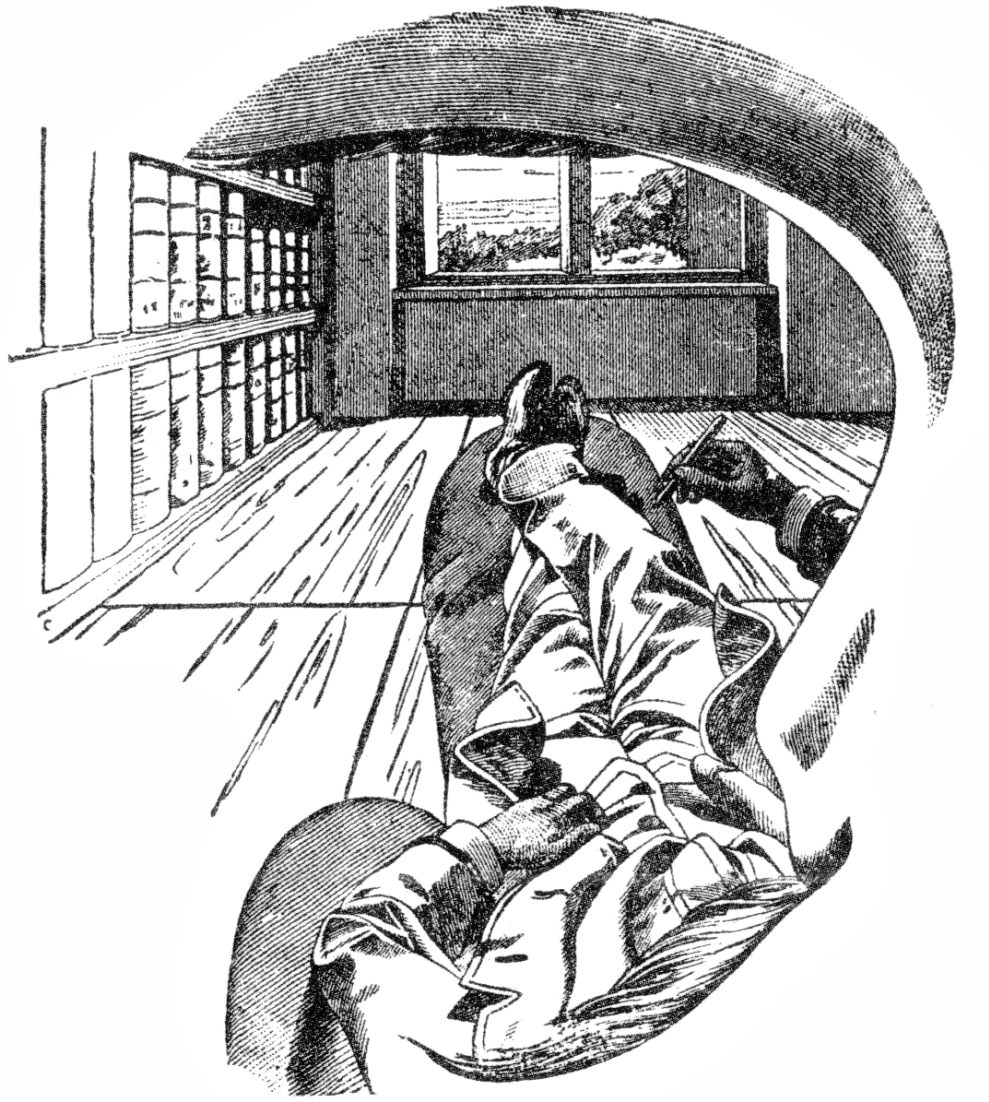
- Bottum up with selected hard coded features;
- Top down and goal directed;
- Top down - bottom up: steered by prediction quality

Introspection and Simulation

Self Inspection Engine

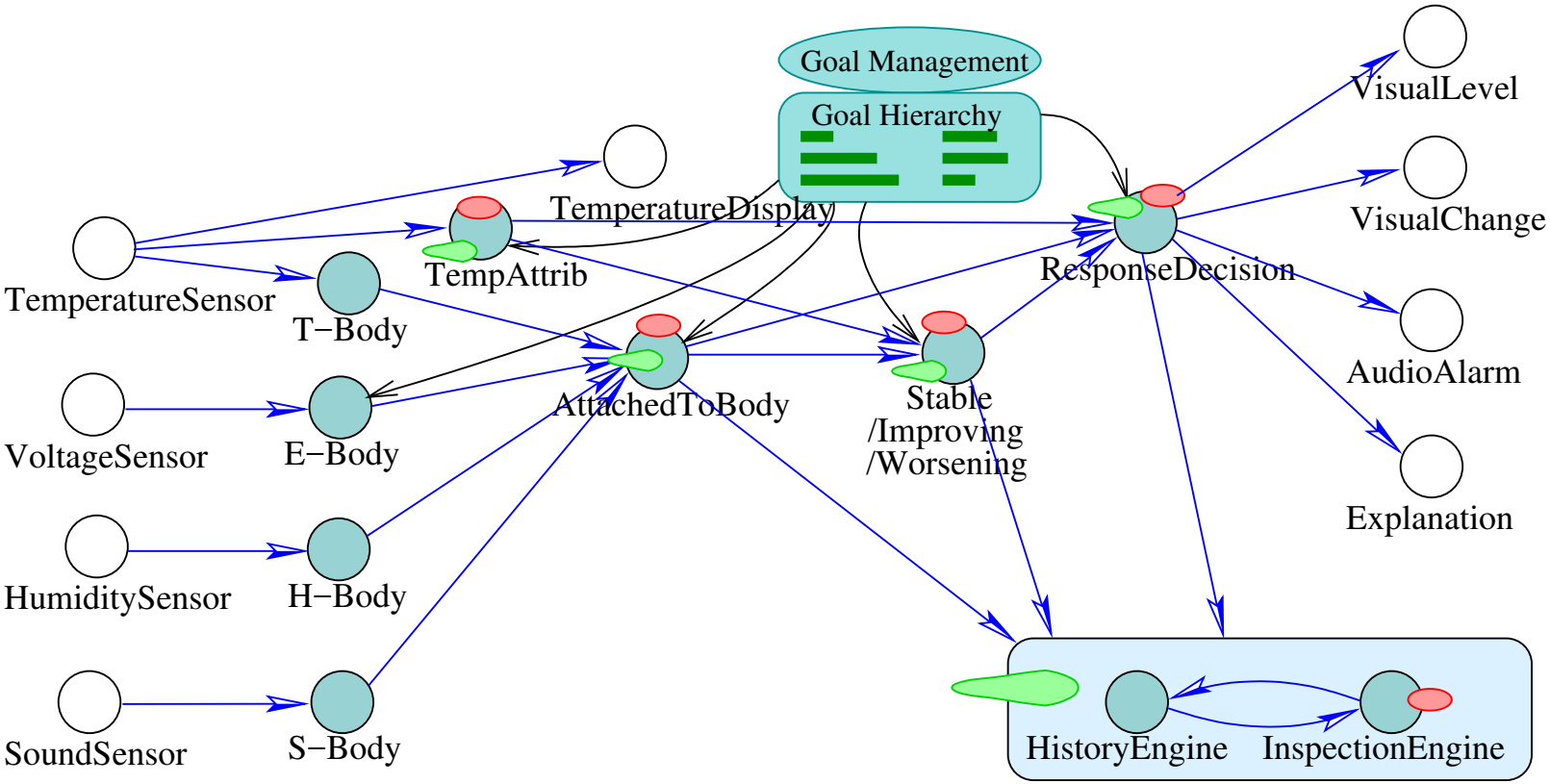
Model Transformation

Simulation

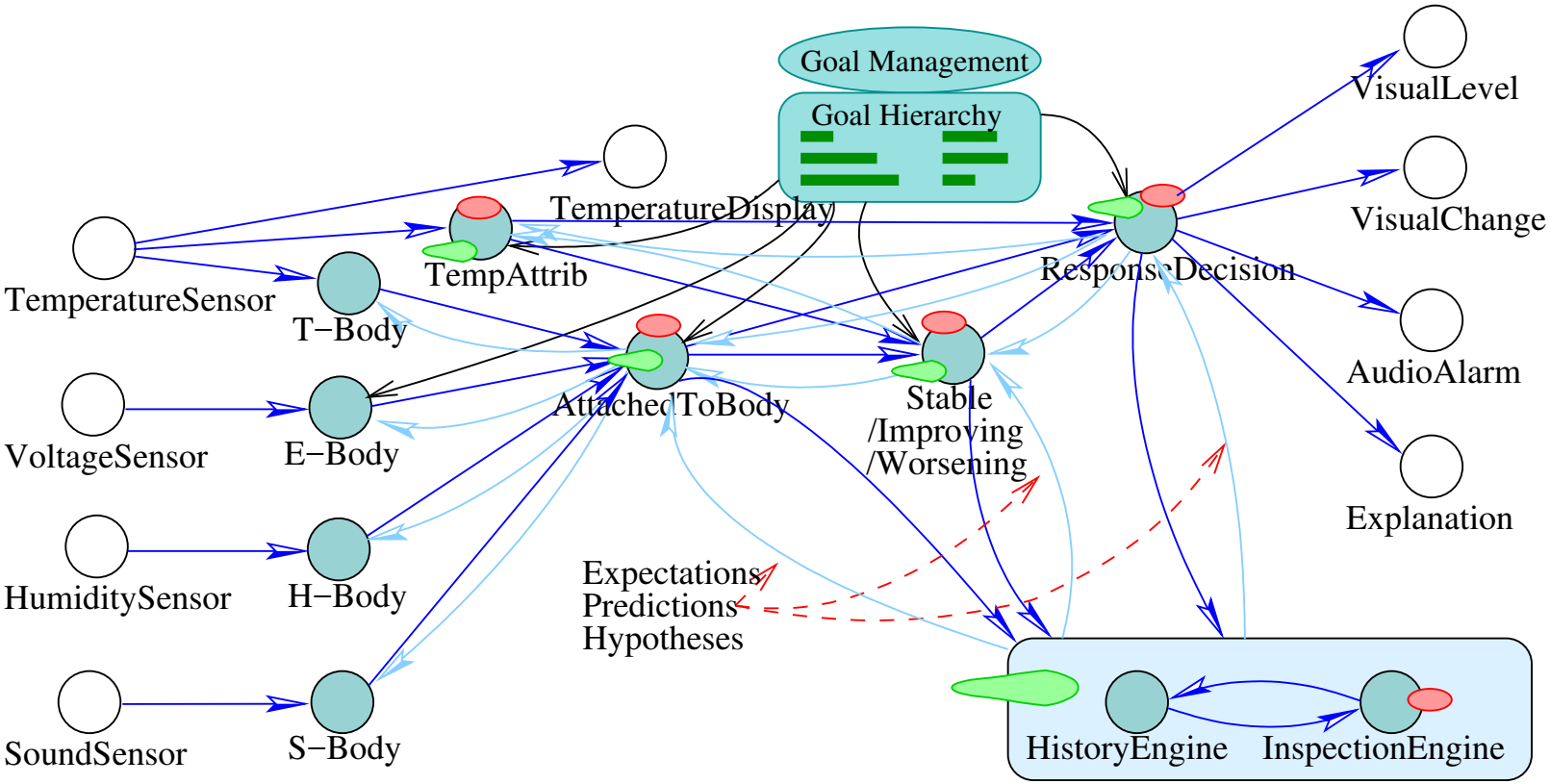


Ernst Mach “Innenperspektive”, 1886

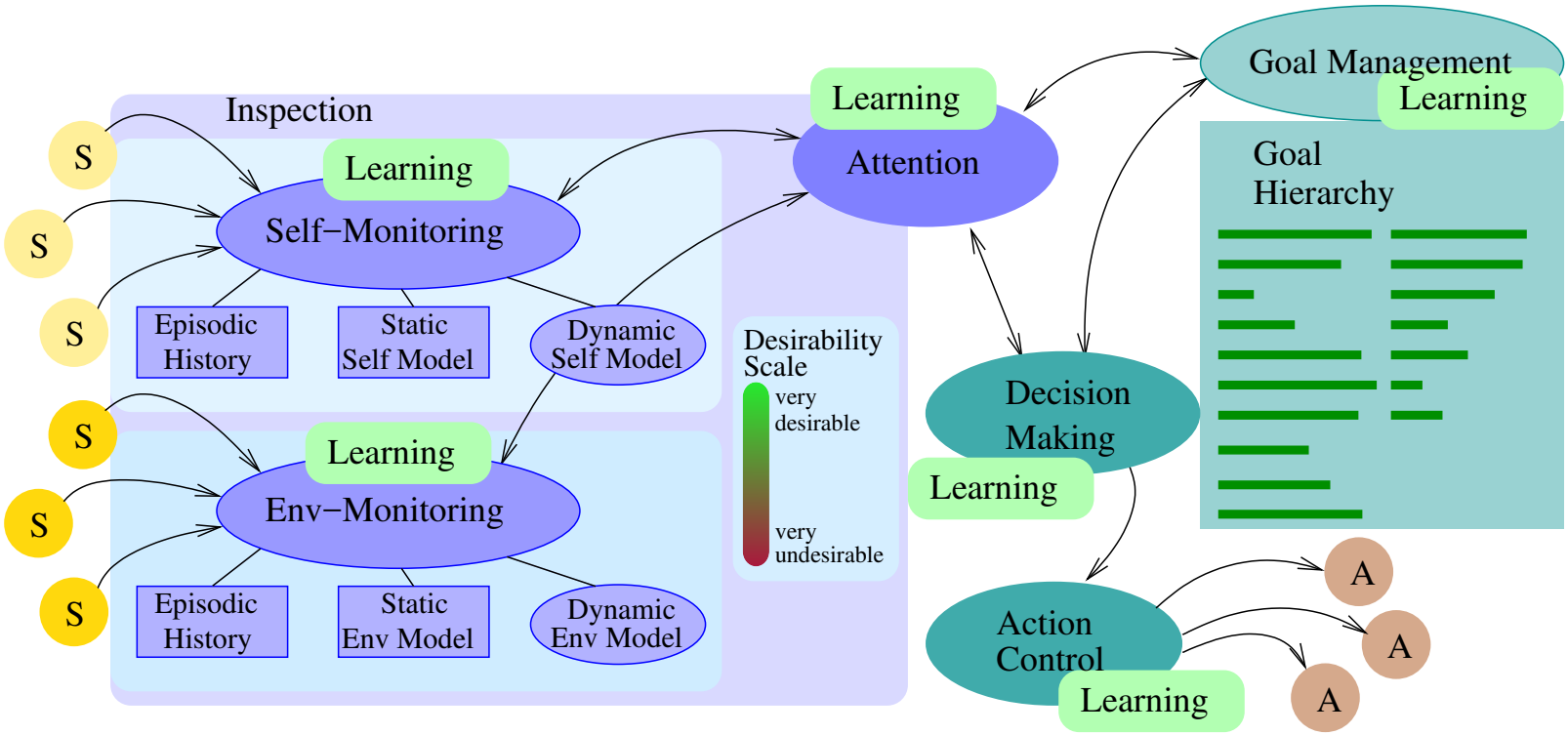
Self-inspecting BioPatch



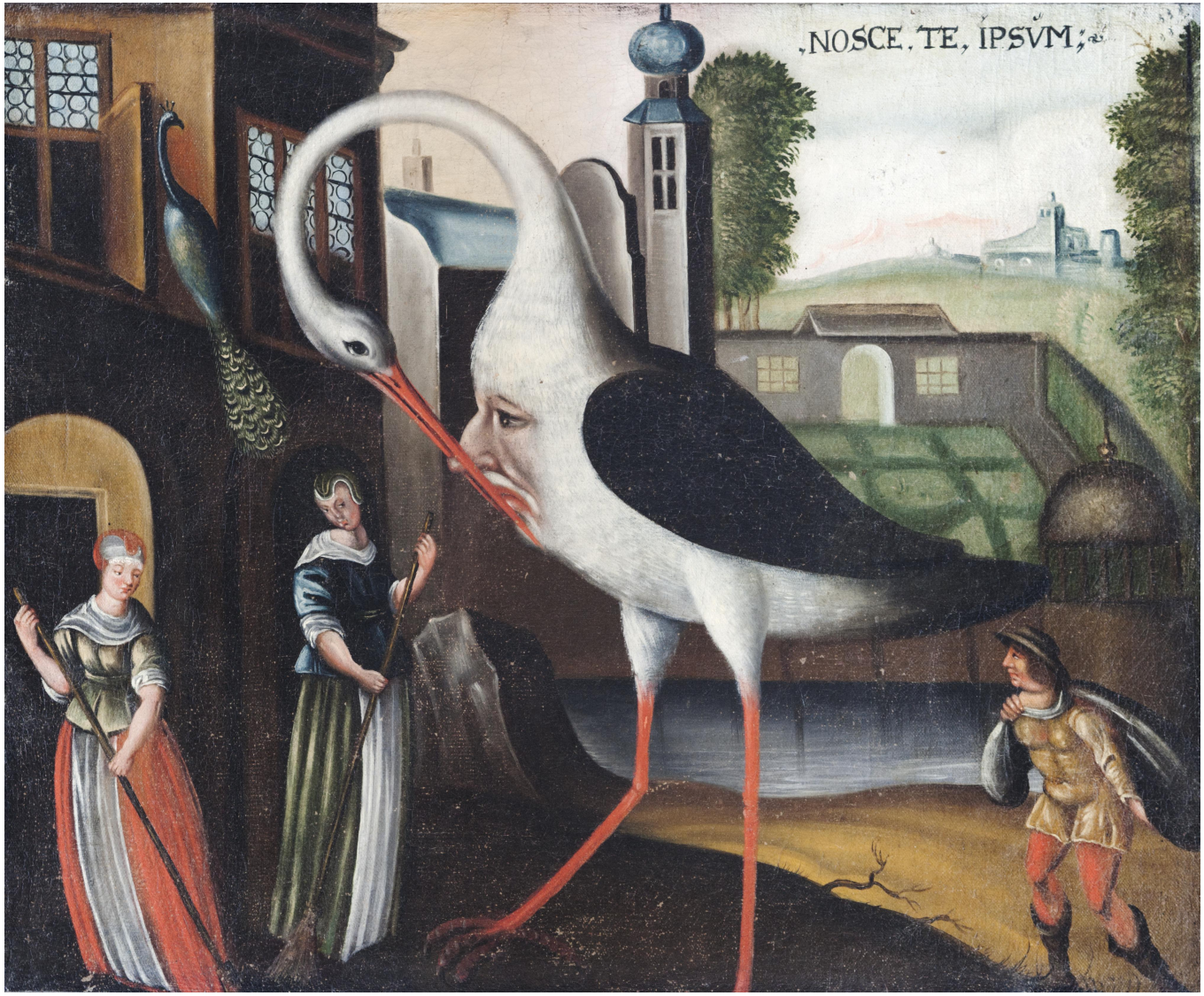
BioPatch with Top-down Prediction



Self-Awareness Architecture



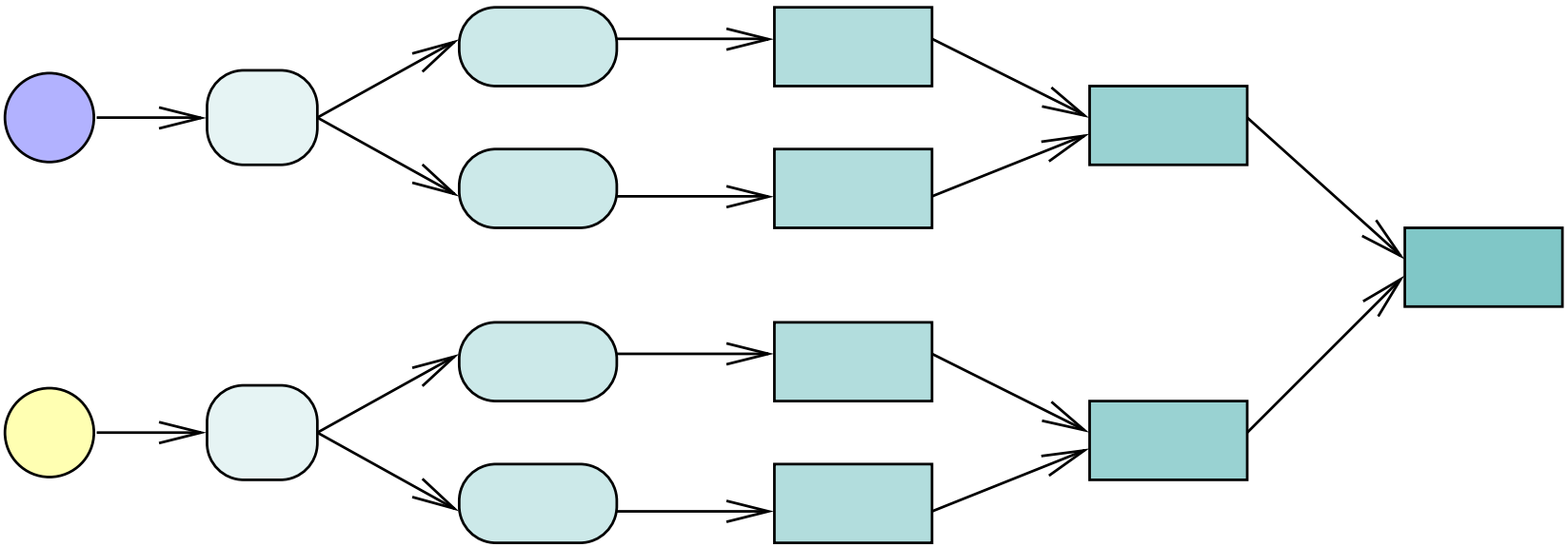
System, Know Thyself



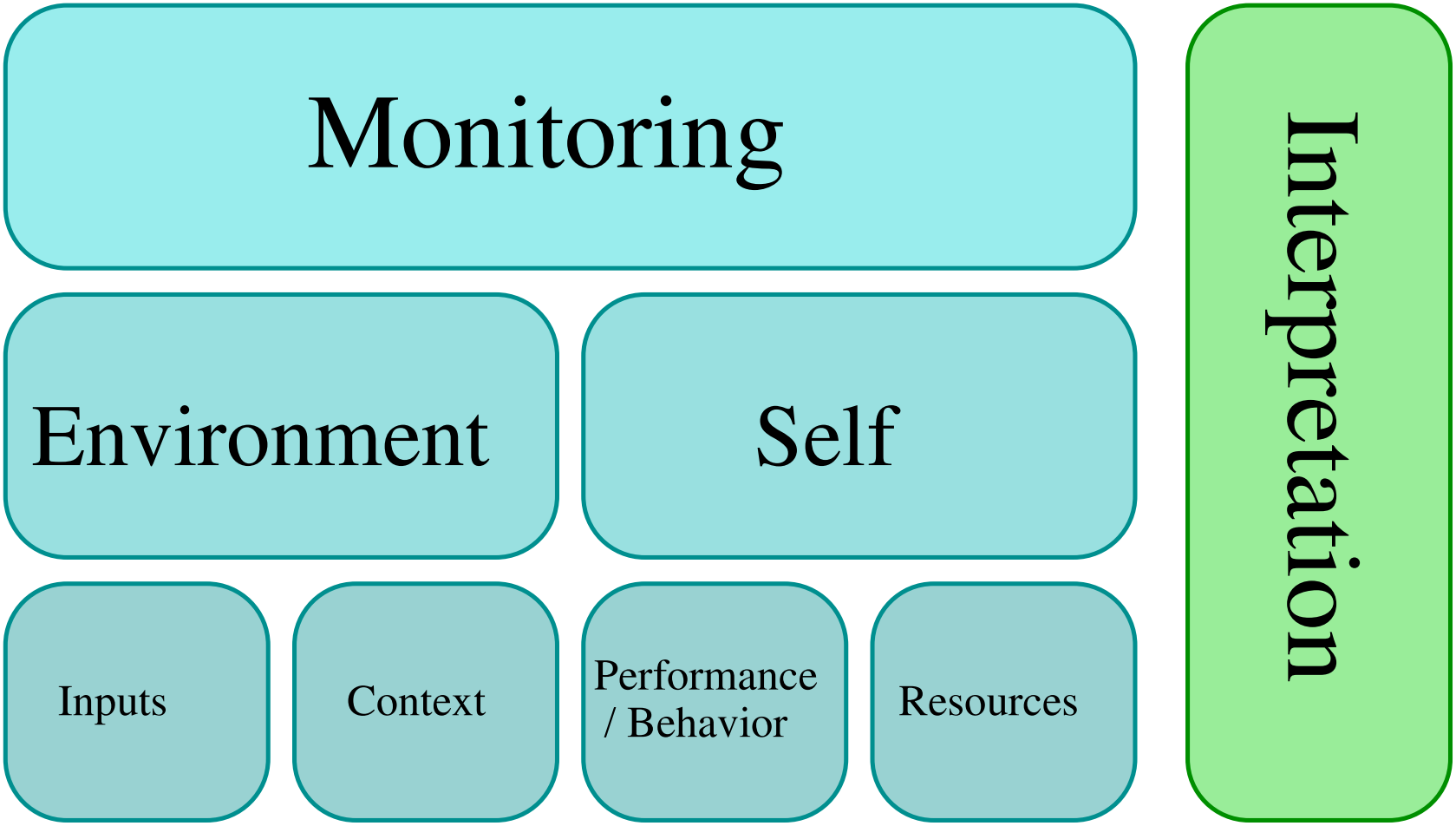
By Unknown - LSH 88977 (sm_dig3542), Public Domain,
https://commons.wikimedia.org/w/index.php?curid=28911161

Observation and Abstraction

Observation Pipeline

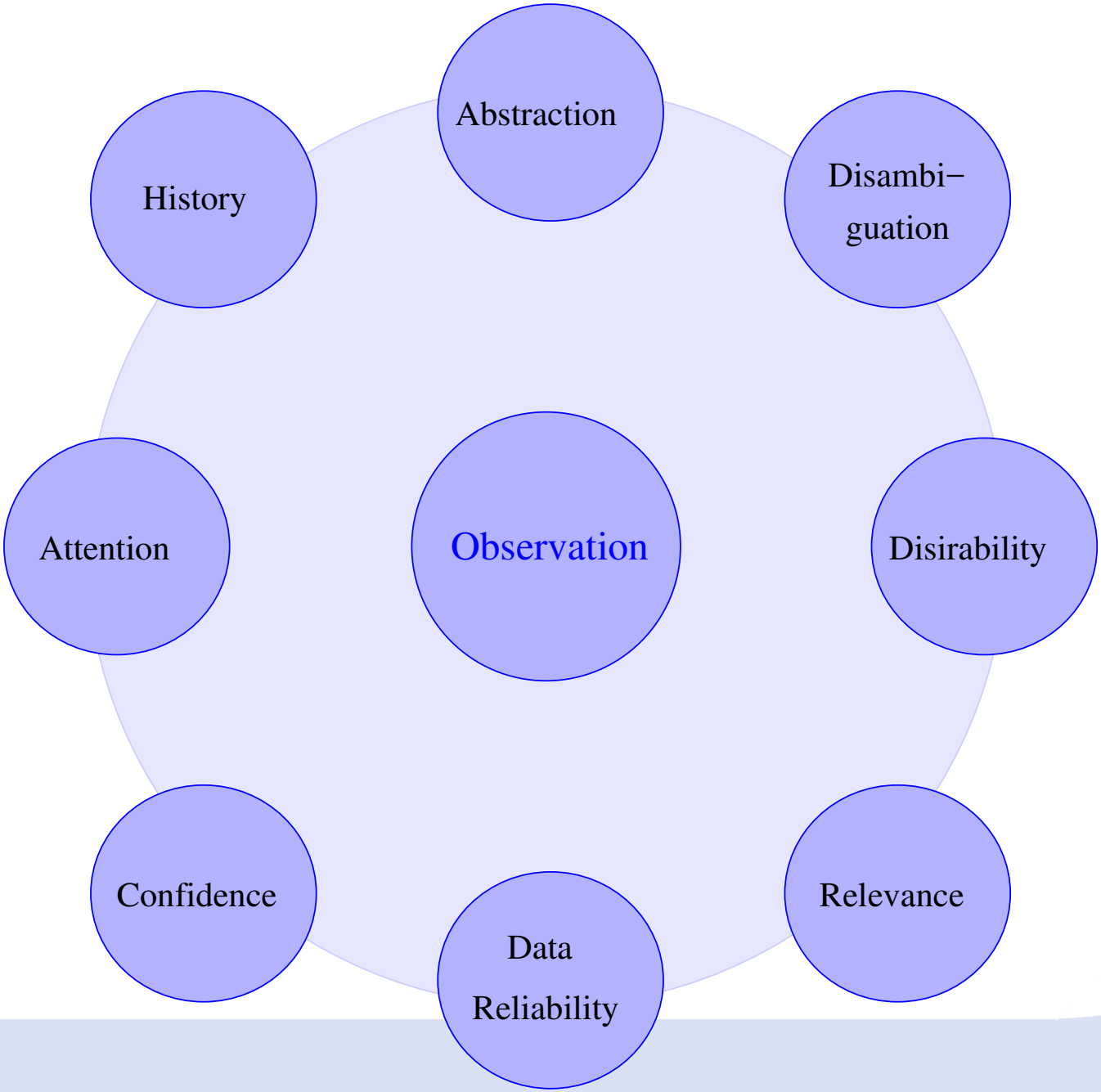


Comprehensive Observation

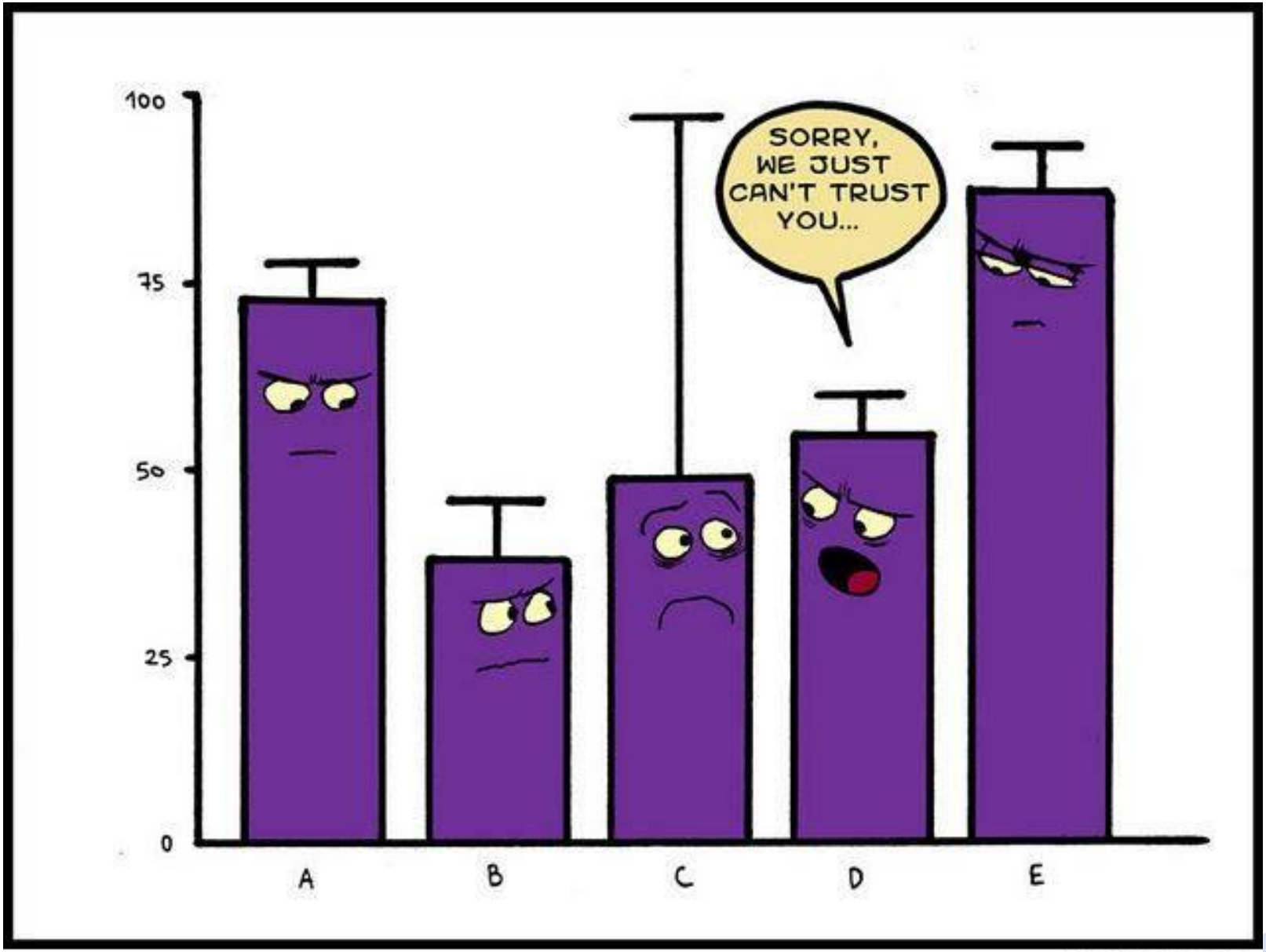


Nima TaheriNejad, Axel Jantsch, and David Pollreisz. "Comprehensive Observation and its Role in Self-Awareness - An Emotion Recognition System Example". In: *Proceedings of the Federated Conference on Computer Science and Information Systems*. Gdansk, Poland, Sept. 2016

Observation Circle



Data and Meta-Data I



Data and Meta-Data II

Accuracy Systematic errors, a measure of statistical bias.

Precision Random errors, a measure of statistical variability.

Data and Meta-Data III

Given: $X = \langle x_0, \ldots, x_n \rangle$ ground truth

$X' = \langle x'_0, \ldots, x'_n \rangle$ measured data

$$\mu(X) = \frac{1}{n} \sum_i x_i$$ mean of ground truth

$$\mu(X') = \frac{1}{n} \sum_i x'_i$$ mean of measured data

Data and Meta-Data IV

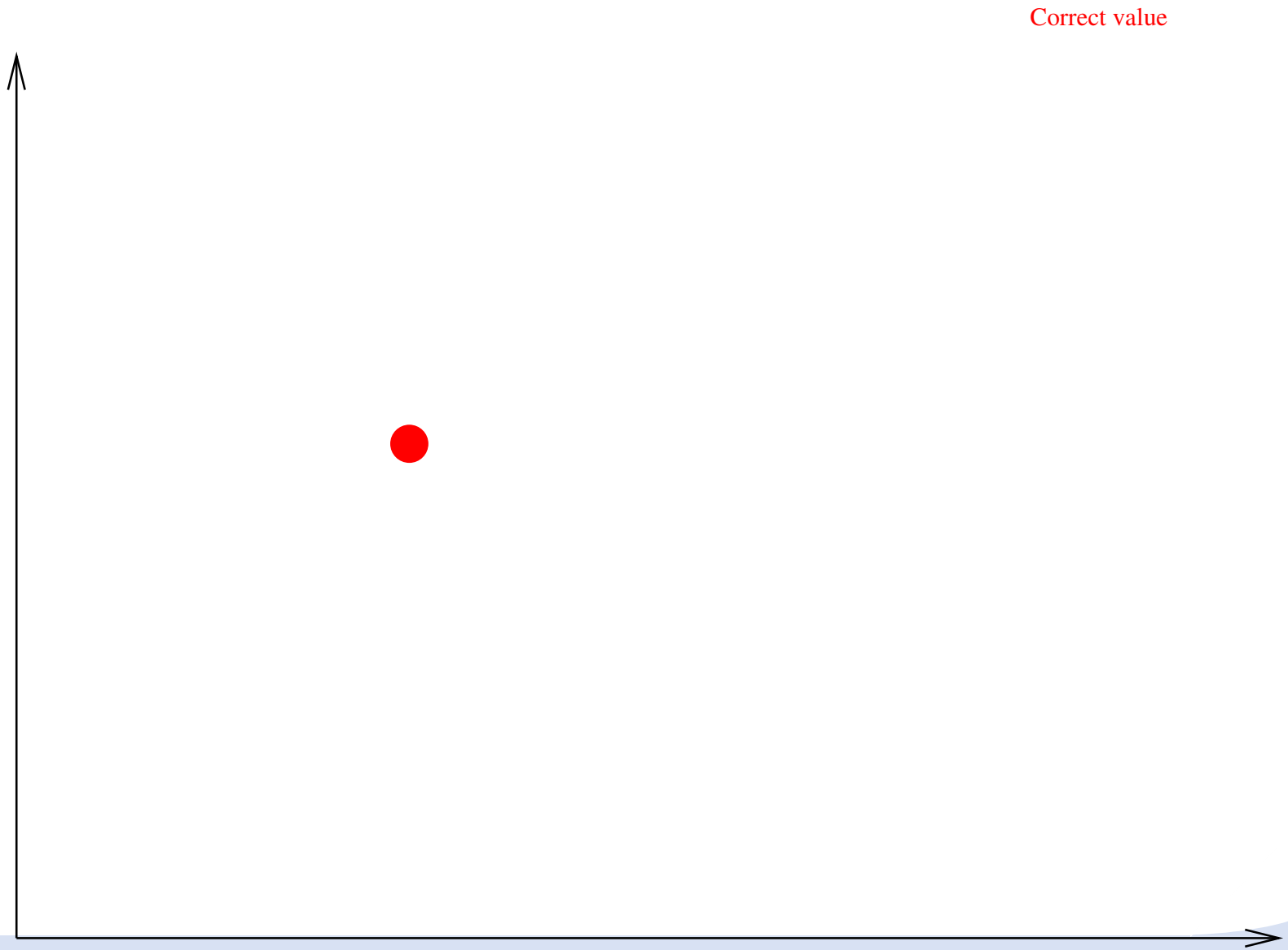
Accuracy:

$$A(X') = \mu(X) - \mu(X')$$

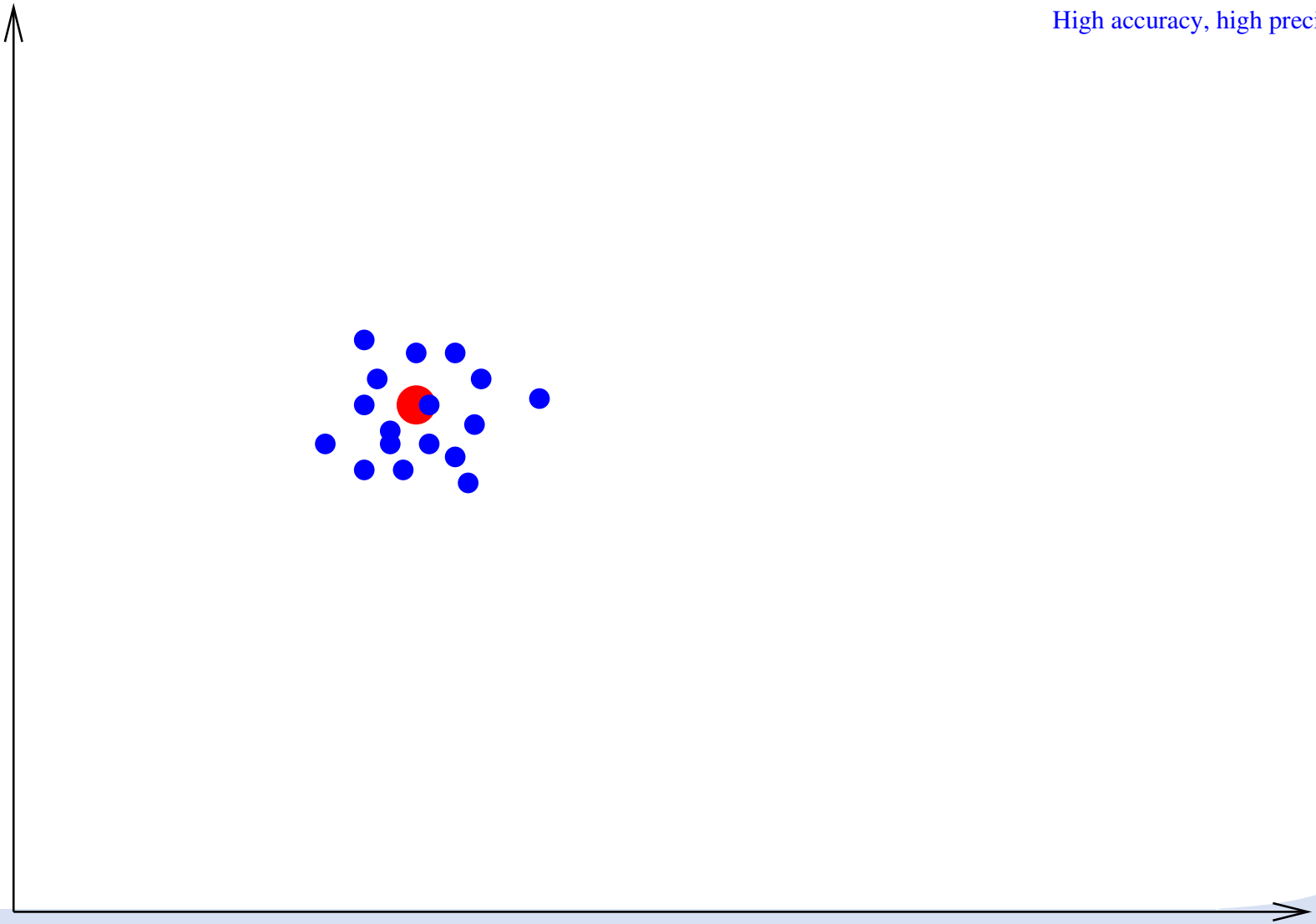
Precision:

$$P(X') = \sigma(X') = \sqrt{\frac{1}{n}\sum_i (x'_i - \mu(X'))^2}$$

Data and Meta-Data V



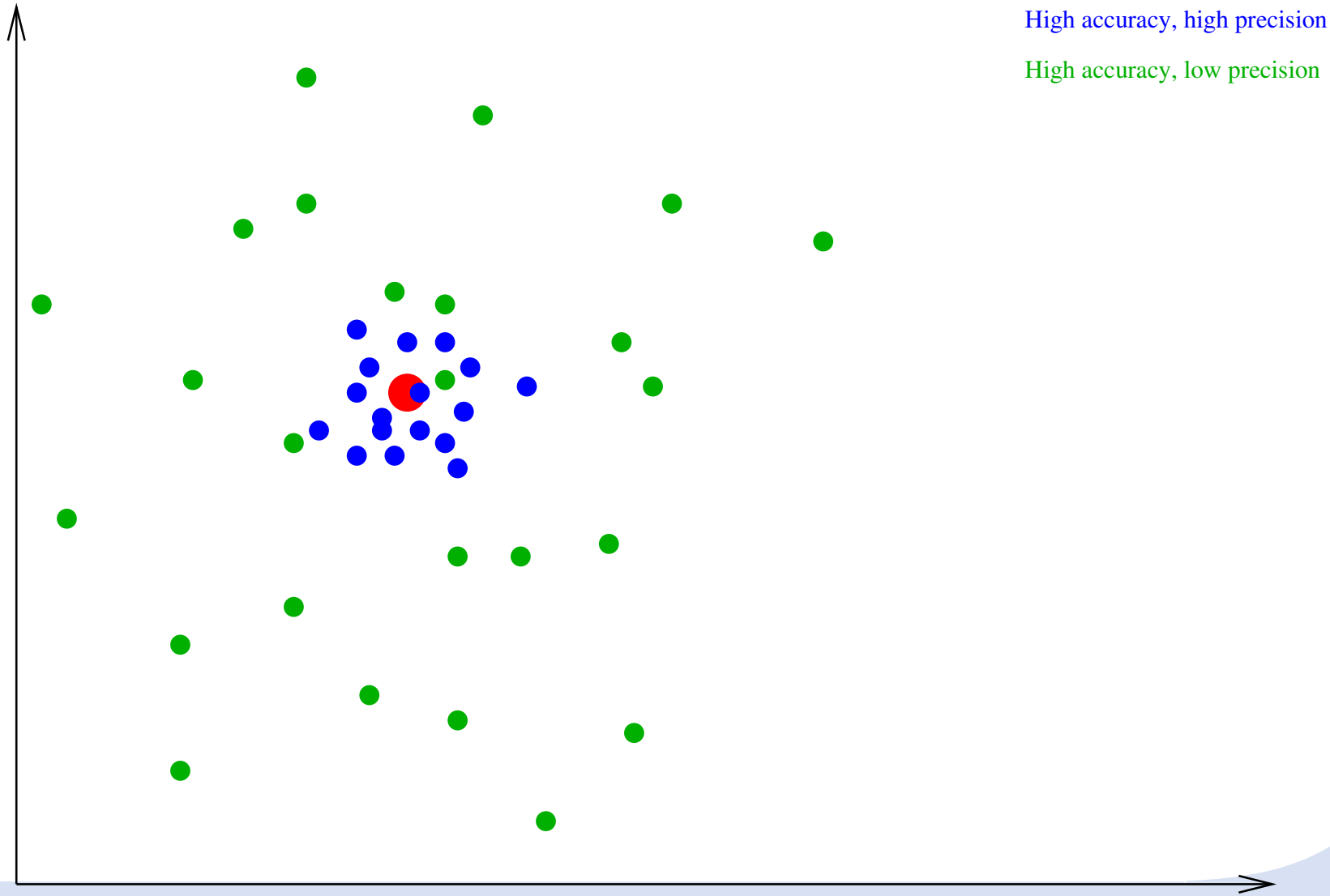
Data and Meta-Data VI



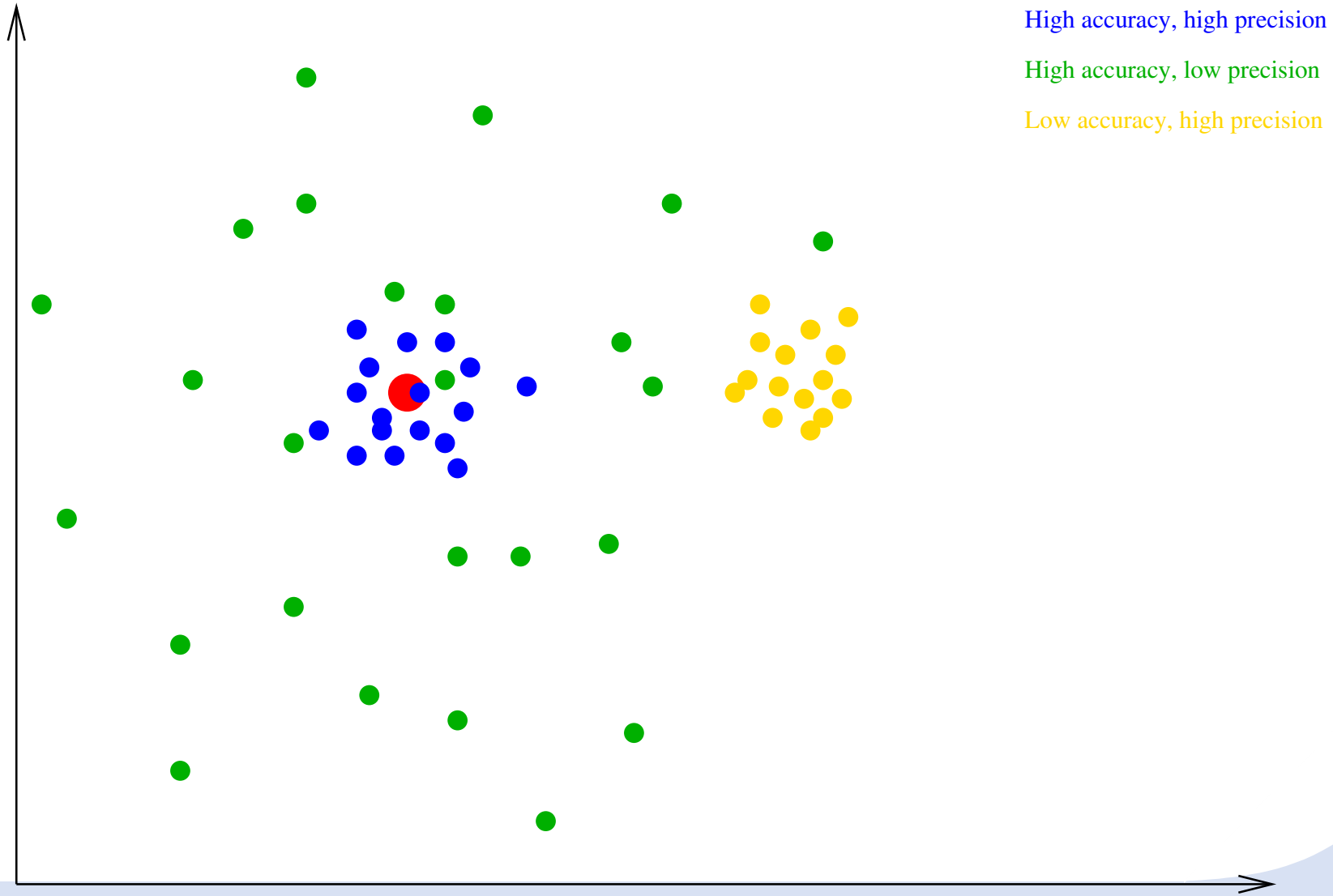
Correct value

High accuracy, high precision

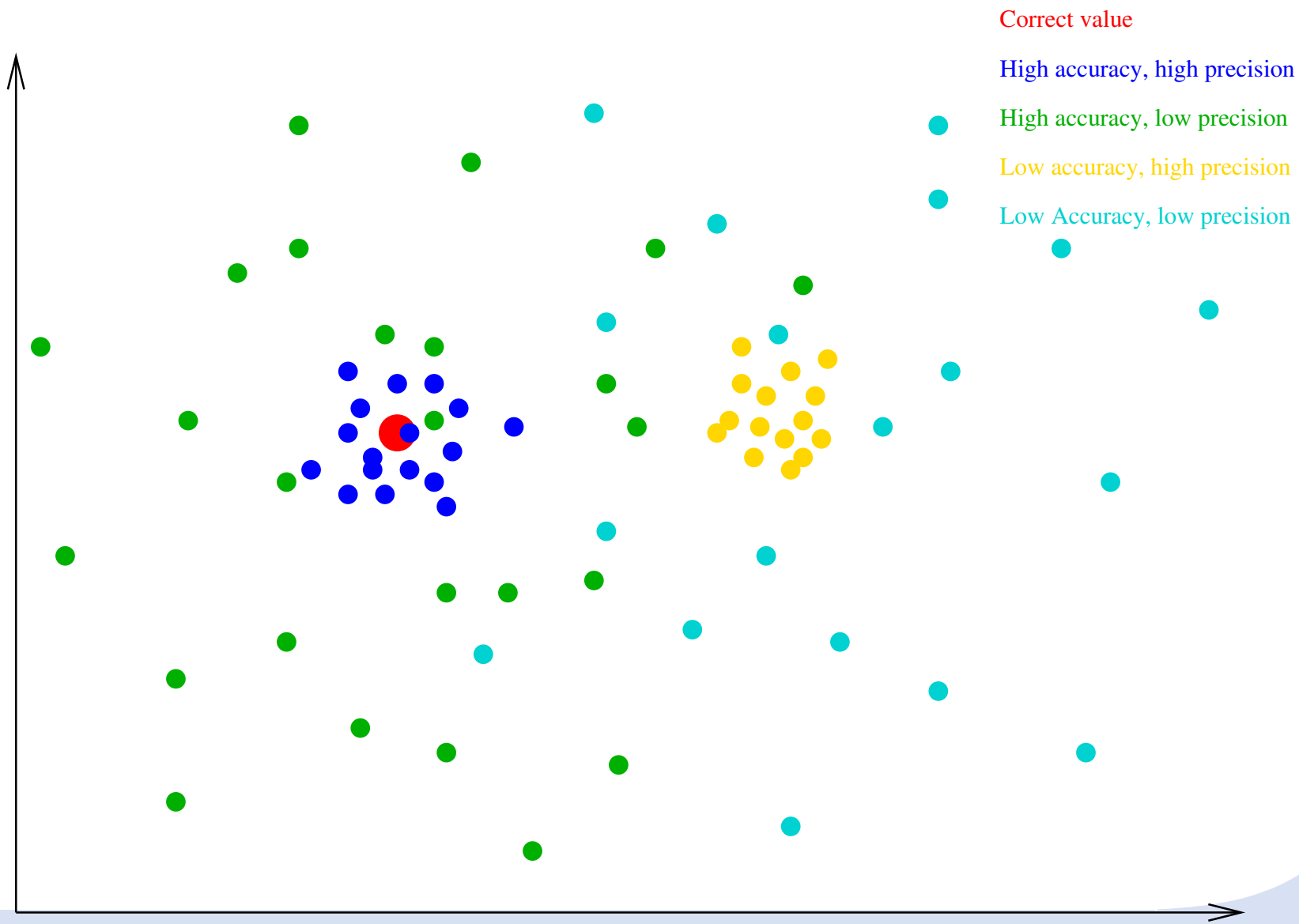
Data and Meta-Data VII



Data and Meta-Data VIII



Data and Meta-Data IX



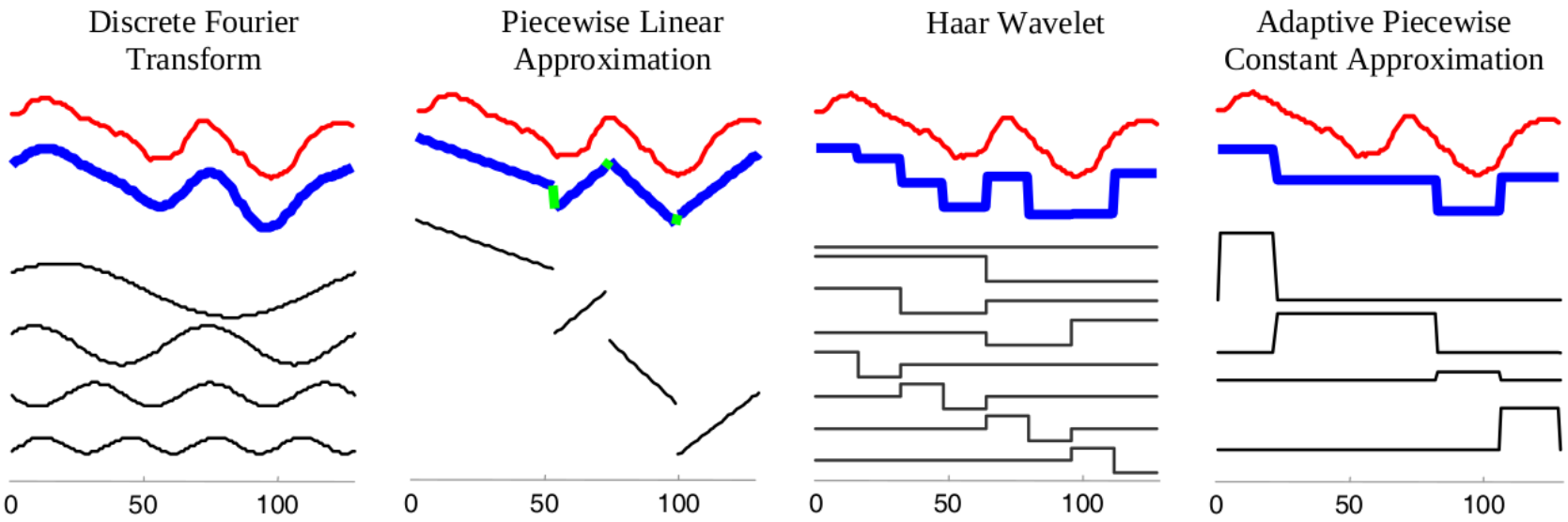
Data and Meta-Data X

- Precision can be calculated when all X' are known;
- Accuracy requires knowledge of ground truth;
- Top down hints to estimate precision and accuracy:
  - Consistency (precision)
  - Plausibility (accuracy)
  - Cross-validity (accuracy)

Symbolization of Signals

- Common representations: DFT, Piecewise Linear Approximation, Haar Wavelet, Adaptive Piecewise Constant Approximation
- SAX: Symbolic Aggregate Approximation
  - Piecewise Aggregate Approximation
  - Discretization
  - Distance Measures
  - Lower Bounding Euclidean Distance

Common Signal Representations



Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Leonardi. "Experiencing SAX: a Novel Symbolic Representation of Time Series". In: *Data Mining and Knowledge Discovery* 15.2 (Oct. 2007), pp. 107–144

Desired Properties

- Compact and memory efficient
- Efficient to compute
- Faithful to relevant properties
- Tightly, lower bounding the true distance between signals

SAX: Symbolic Aggregate Approximation

- Normalization
- Piecewise Aggregate Approximation (PAA)
- Discretization
- Distance metric
- Lower bounding Euclidean distance

Following presentation based on

Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Leonardi. “Experiencing SAX: a Novel Symbolic Representation of Time Series”. In: *Data Mining and Knowledge Discovery* 15.2 (Oct. 2007), pp. 107–144

Notation

C ... A time series $C = c_1, \ldots, c_n$

$\bar{C}$ ... A piecewise aggregate approximation of a time series $\bar{C} = \bar{c}_1, \ldots, \bar{c}_w$

$\hat{C}$ ... A symbol representation of a time series $\hat{C} = \hat{c}_1, \ldots, \hat{c}_w$

n ... Number of data points in time series C

w ... Number of PAA segments representing C

r ... Number of data points collapsed into one symbol $r = \frac{n}{w}$

A ... Alphabet size

Normalization

Normalization centers the time series on 0 and expresses the amplitude as multiples of the standard deviation.

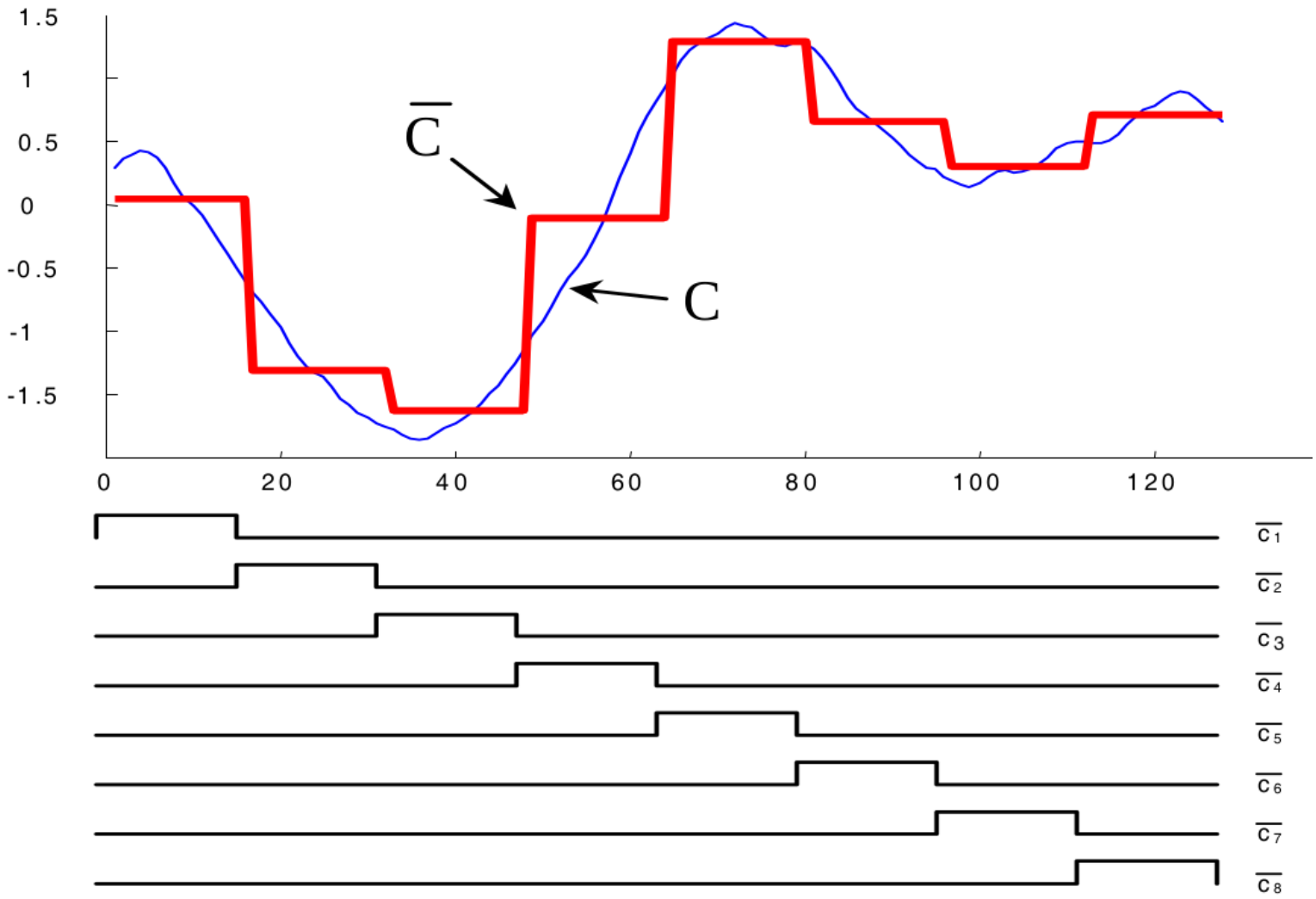
$$C = c_1, c_2, \ldots, c_n$$

$$\mu = \frac{1}{n} \sum_{i=1}^{n} c_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (c_i - M)^2}$$

$$C' = \frac{c_1 - \mu}{\sigma}, \frac{c_2 - \mu}{\sigma}, \ldots, \frac{c_n - \mu}{\sigma}$$

PAA: Piecewise Aggregate Approximation I



PAA: Piecewise Aggregate Approximation II

$$\bar{c}_i = \frac{1}{r} \sum_{j=r(i-1)+1}^{ri} c_j$$

Assume:

$$C = 0.4, 0.5, 0.4, 0, 3, 0.2, 0, 2, 0.1, 0.2, 0.2, 0.3, 0.4, 0.5$$

$$n = 12$$

$$w = 4$$

$$r = 3$$

$$\bar{C} = \frac{0.4 + 0.5 + 0.4}{3}, \frac{0.3 + 0.2 + 0.2}{3}, \frac{0.1 + 0.2 + 0.2}{3}, \frac{0.3 + 0.4 + 0.5}{3}$$

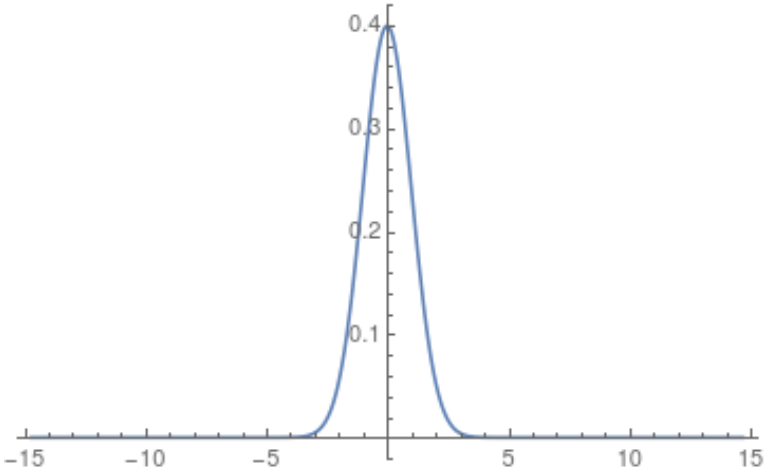
$$= 0.43, 0.23, 0.17, 0.4$$

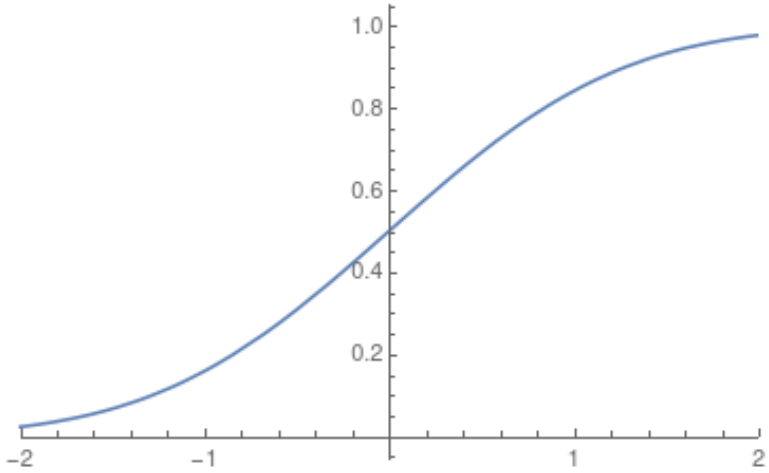
Discretization

Assignment of PAA values to symbols.

- Larger alphabet means more accuracy and less abstraction;
- Symbols should appear with similar probability;
- Normalized time series approximate Gaussian distributions;
- Breakpoints can be pre-determined based on a Gaussian distribution with $\mu = 0, \sigma = 1$;

Normal Distribution





β_i \ a	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

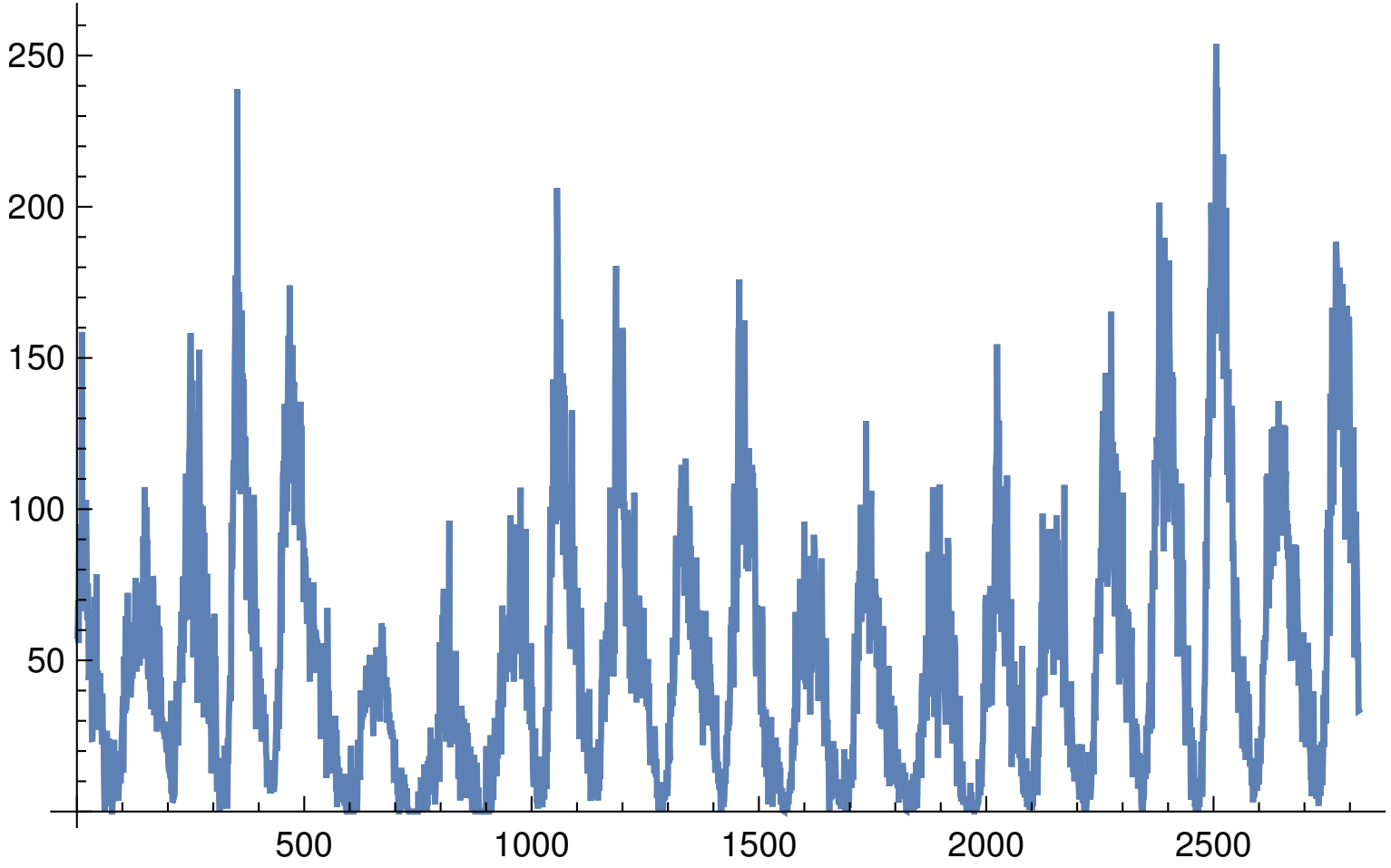
Sunspot Example I

- Monthly count of the number of observed sunspots for over 230 years (1749-1983).
- Units are a number of sunspots.
- 2,820 observations.

From `machinelearningmastery.com/time-series-datasets-for-machine-learning`

Sunspot Example II

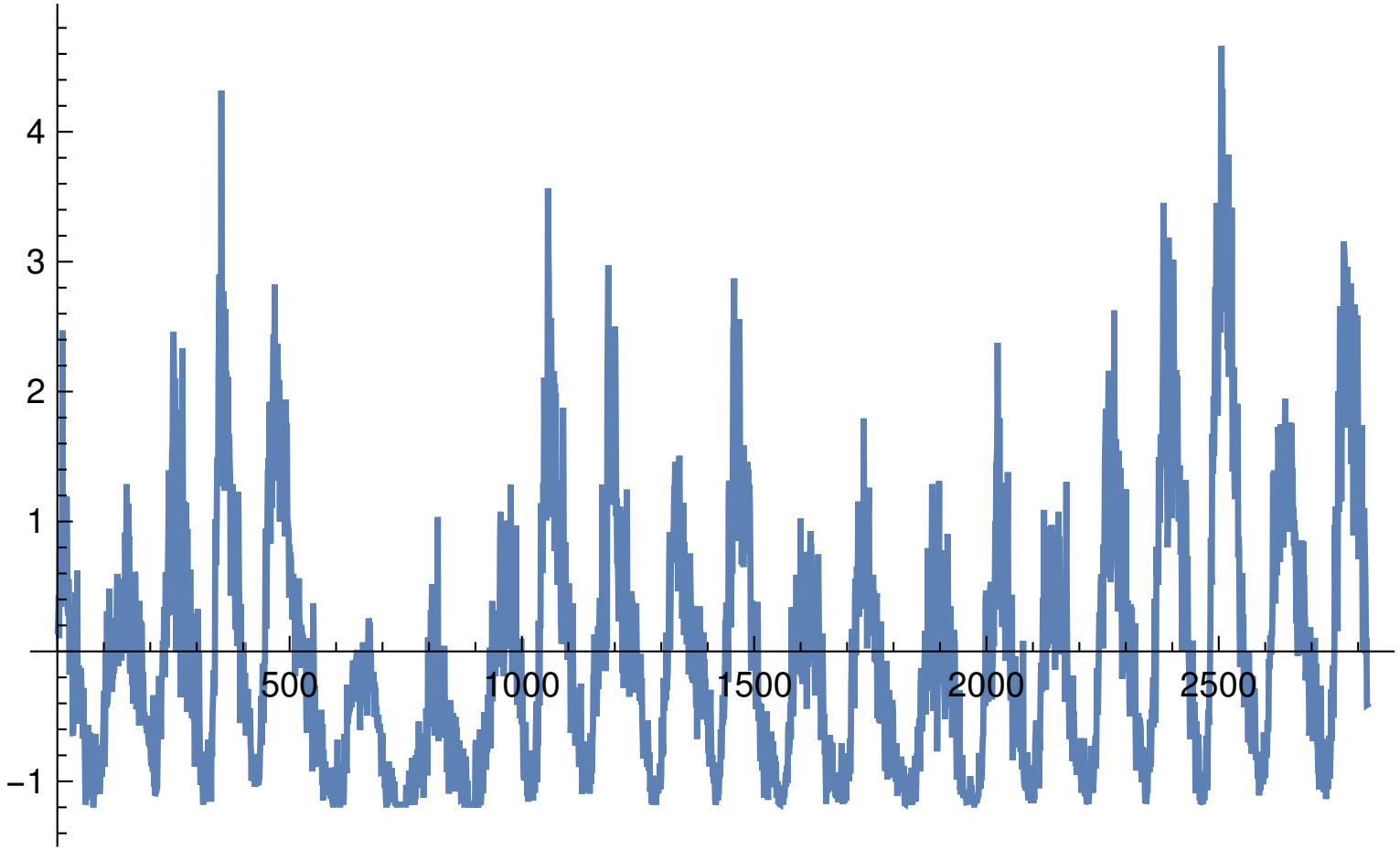
Full data plot 1749-1983:



$n = 2820, \mu = 51.3, \sigma = 43.5$

Sunspot Example III

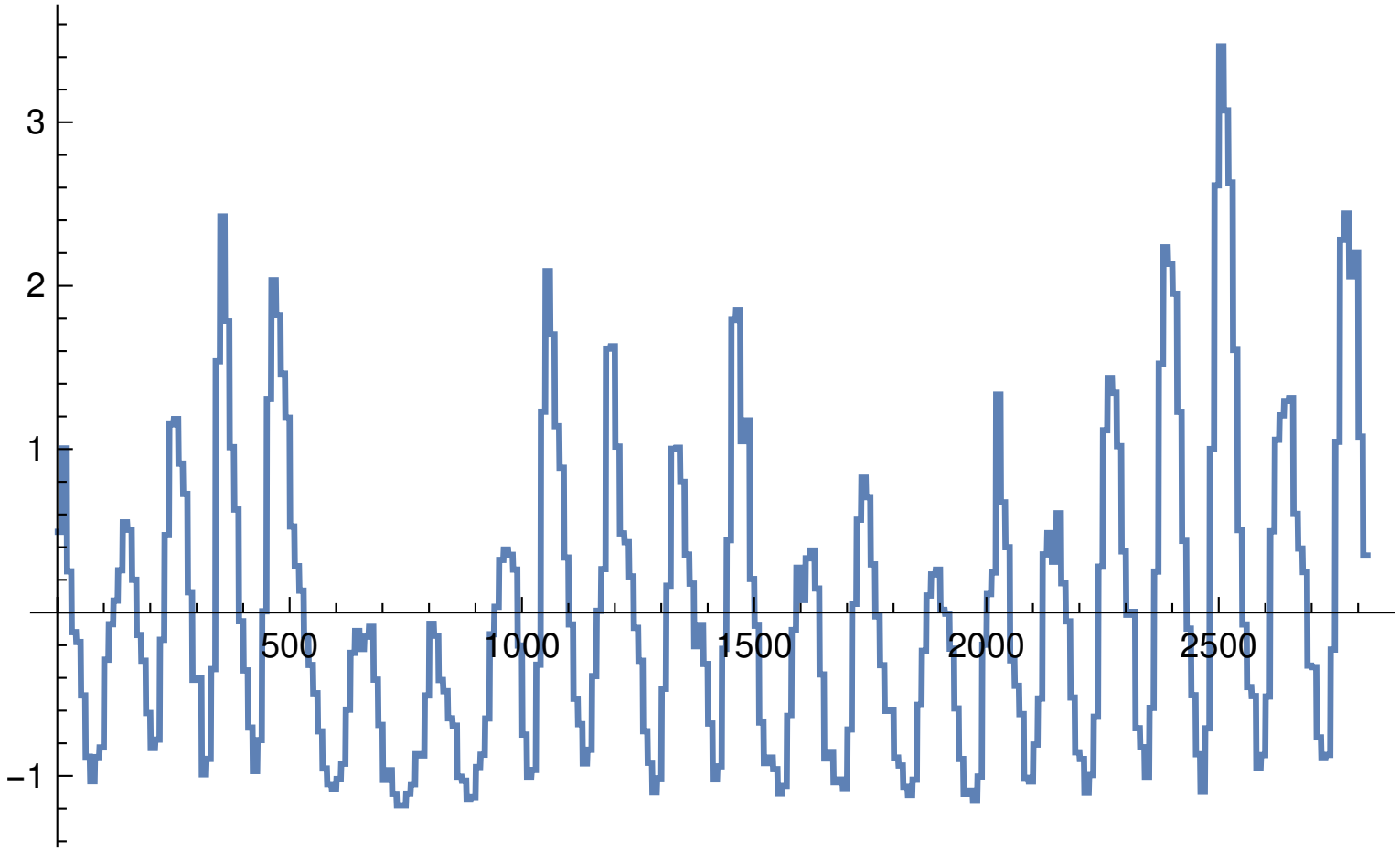
Normalized time series:



$n = 2820, \mu = 0, \sigma = 1$

Sunspot Example IV

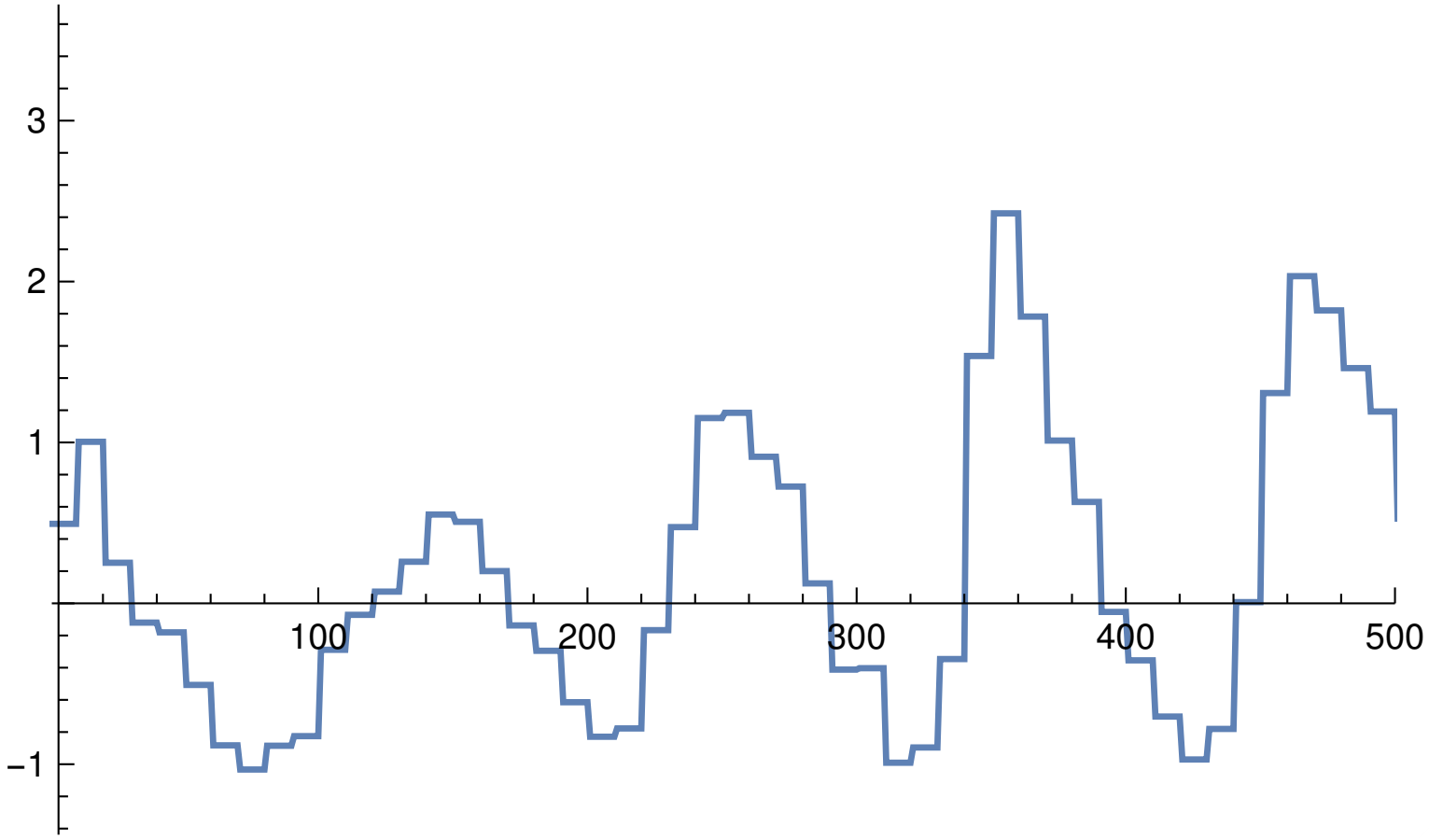
Piecewise Aggregate Approximation:



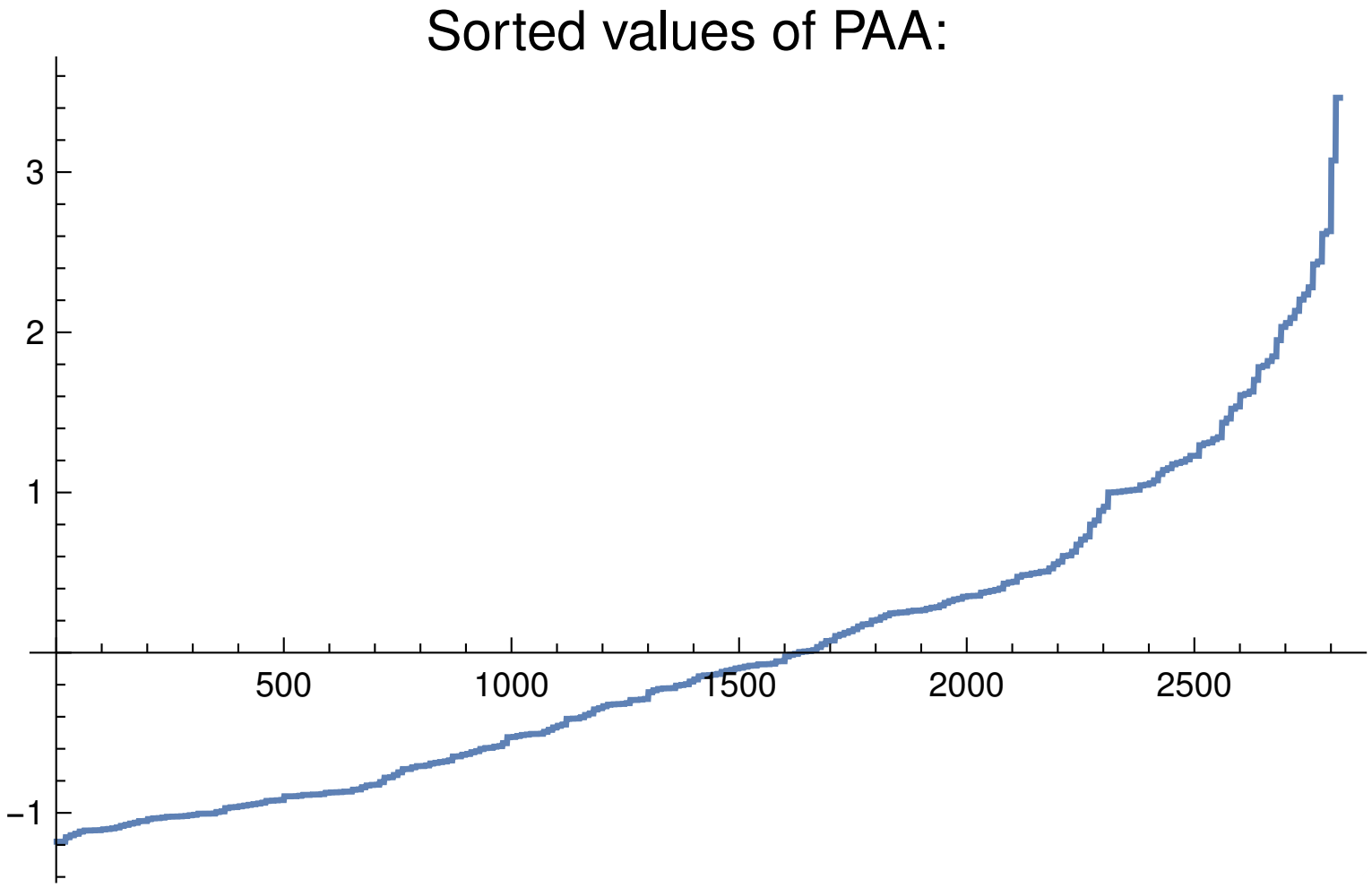
$w = 282, r = 10, \mu = 0, \sigma = 0.94$

Sunspot Example V

Piecewise Aggregate Approximation for 500 data points:

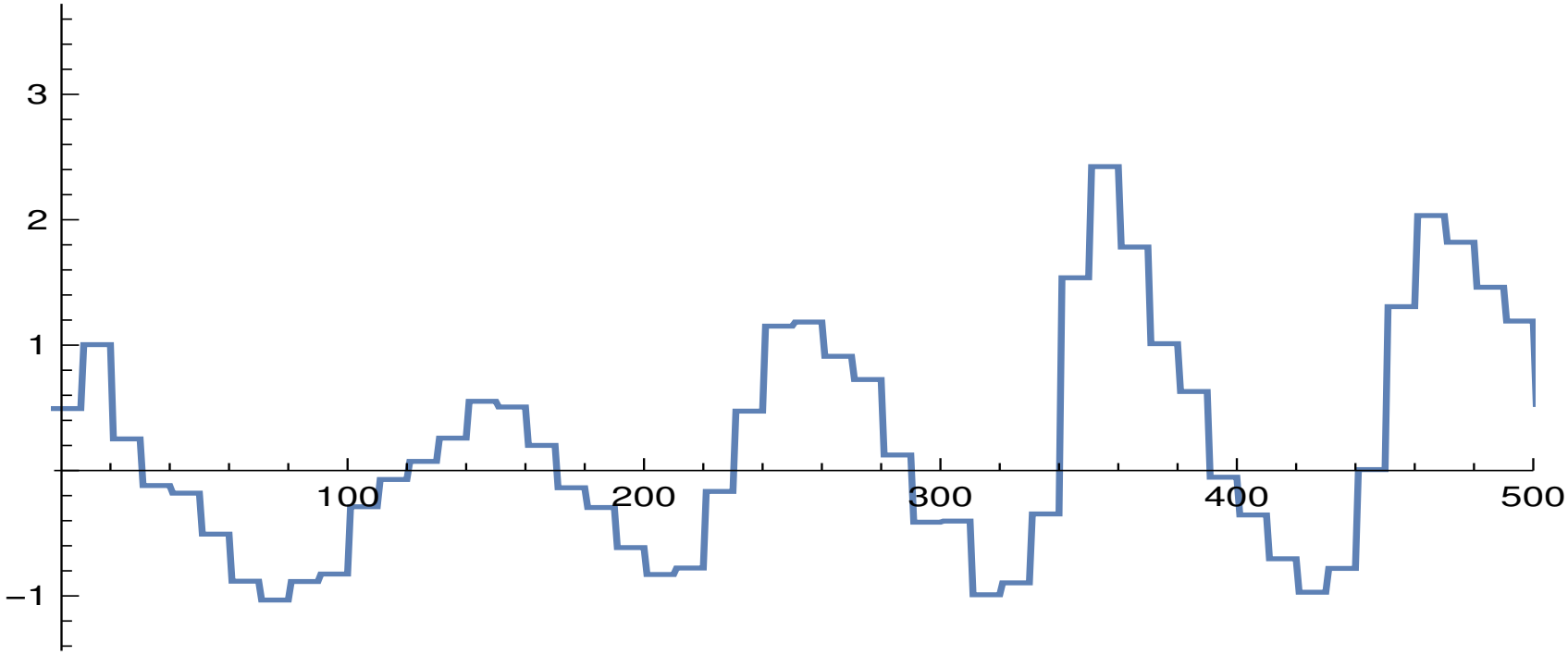


Sunspot Example VI



Sunspot Example VII

Symbolized time series:



ccbbbaaaaabbbbccbbbaaabccccccbbbaabccccccbbaaabccccc

112 a's, 96 b's, 74 c's

Distance I

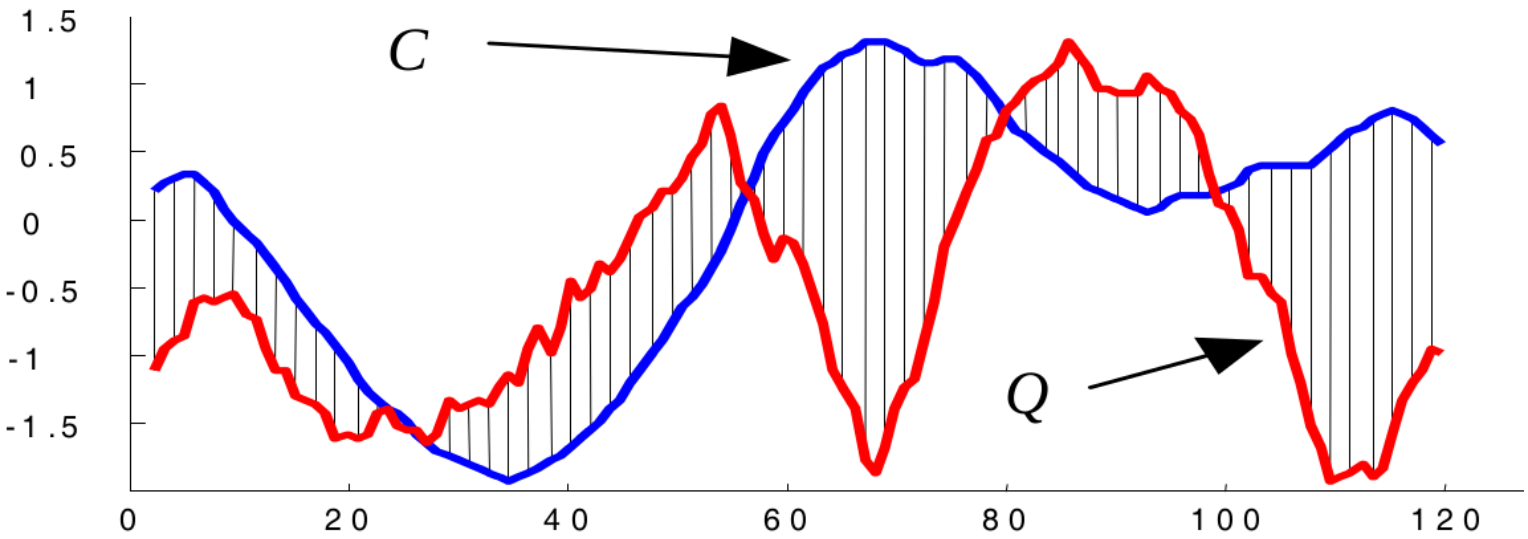
Given two time series C and Q with length n, the Euclidean distance is defined as

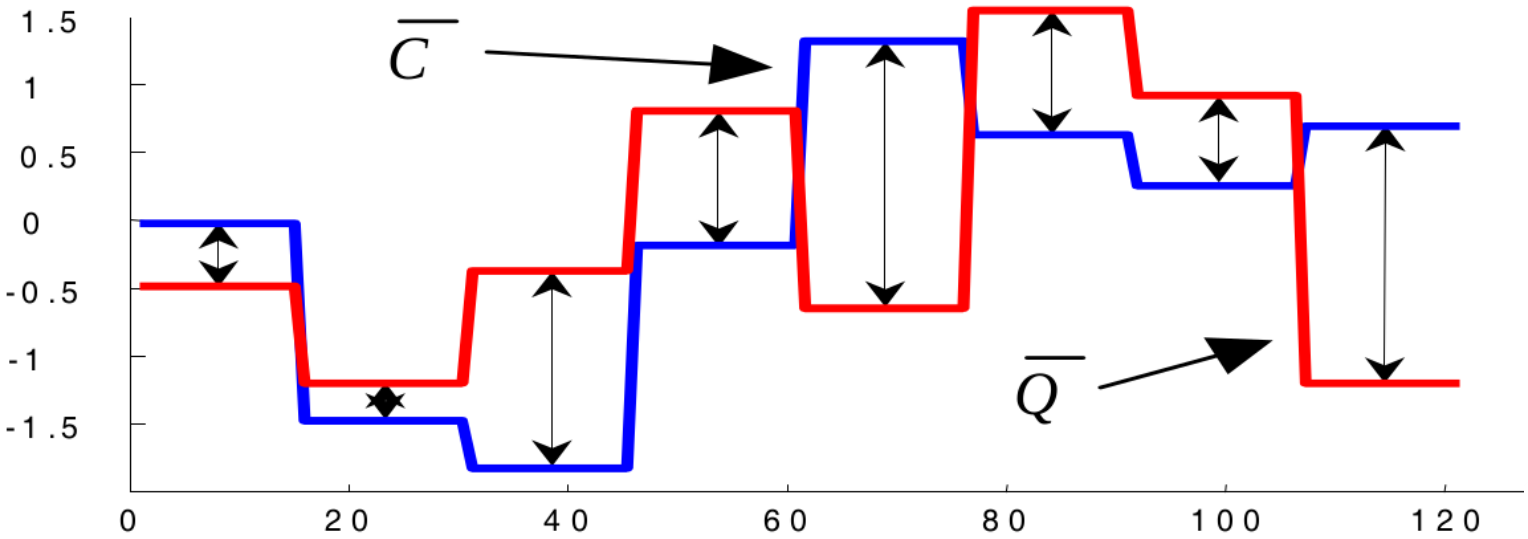
$$D(Q, C) = \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}$$

For the corresponding PAA representations $\bar{Q}$ and $\bar{C}$ the distance is defined as

$$DR(\bar{Q}, \bar{C}) = \sqrt{r \sum_{i=1}^{w} (\bar{q}_i - \bar{c}_i)^2}$$

Distance II





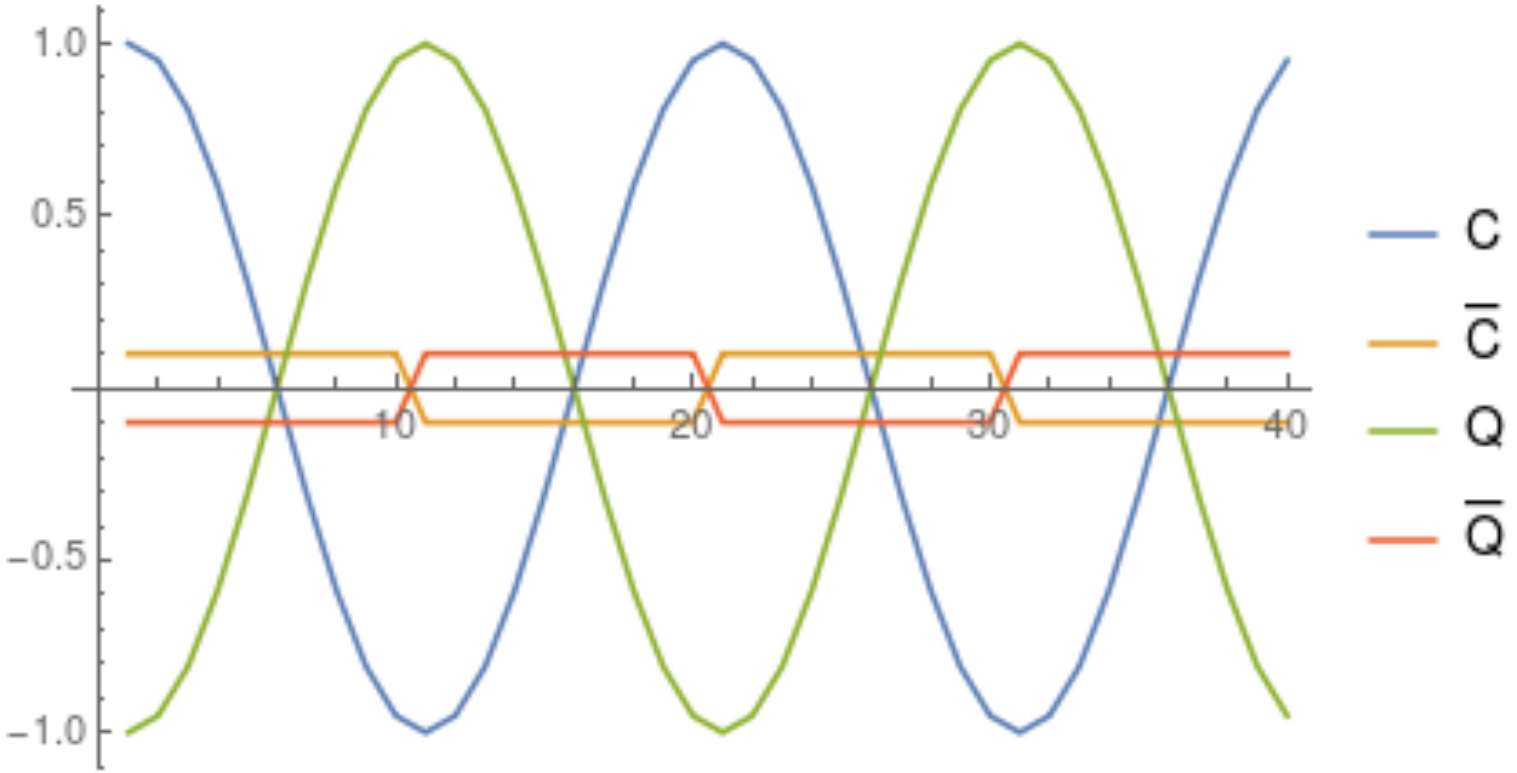
Distance III

$$DR(\bar{Q}, \bar{C}) \leq D(Q, C) \quad \forall C, D$$

DR is a lower bound of *D* because it correctly reflects large grain differences but ignores small grain differences, i.e. differences within *r* samples.

Distance IV

DR is reasonably tight in many cases but its tightness is not bounded.



Distance V

The distance in the symbol space is defined as

$$DS(\hat{Q}, \hat{C}) = \sqrt{r \sum_{i=1}^{w} \delta_A(\hat{q}, \hat{c})^2}$$

$$\delta_A(\hat{q}, \hat{c}) = \begin{cases} 0 & \text{if } |I(\hat{q}) - I(\hat{c})| \leq 1 \\ \beta_{\max(I(\hat{q}), I(\hat{c}))-1} - \beta_{\min(I(\hat{q}), I(\hat{c}))} & \text{otherwise} \end{cases}$$

with $I()$ an index function: $I(a) = 1, I(b) = 2, \ldots$

β_i the breakpoints.

Distance VI

For alphabet size $A = 4$:
Breakpoints: $\beta_1 = -0.67, \beta_2 = 0, \beta_3 = 0.67$

$$\delta_4(\hat{q}, \hat{c}) = \begin{cases} 0 & \text{if } |I(\hat{q}) - I(\hat{c})| \leq 1 \\ \beta_{\max(I(\hat{q}),I(\hat{c}))-1} - \beta_{\min(I(\hat{q}),I(\hat{c}))} & \text{otherwise} \end{cases}$$

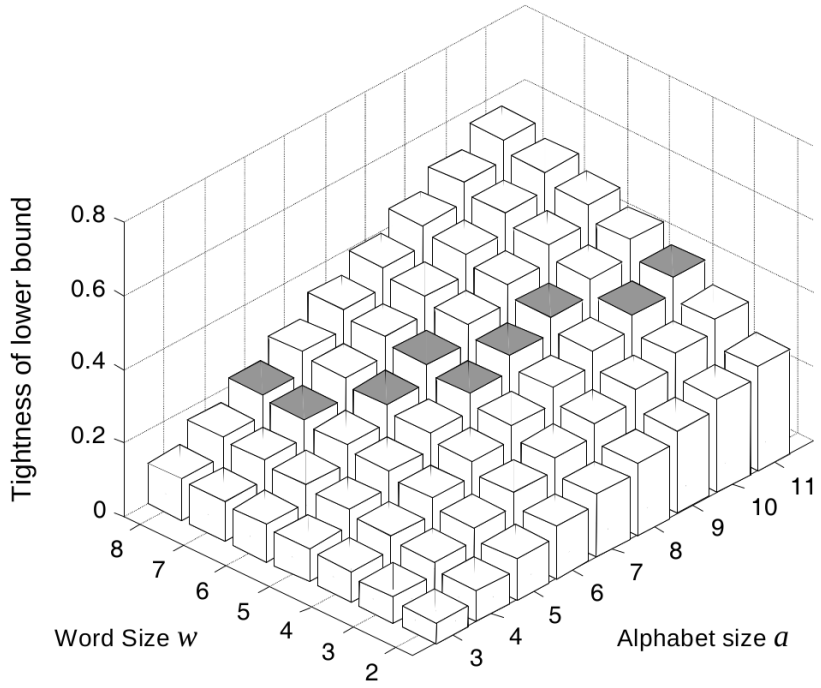
δ_4	a	b	c	d
a	0	0	$\beta_2 - \beta_1$	$\beta_3 - \beta_1$
b	0	0	0	$\beta_2 - \beta_1$
c	$\beta_2 - \beta_1$	0	0	0
d	$\beta_3 - \beta_1$	$\beta_2 - \beta_1$	0	0

	a	b	c	d
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

Distance VII

Tightness of bound:

$$\text{TightnessOfLowerBound} = \frac{DS(\hat{Q}, \hat{C})}{D(Q, C)}$$



Results from large set of experiments with the UCR time series archive from

`http://www.cs.ucr.edu/~eamonn/time_series_data/`

Distance VIII

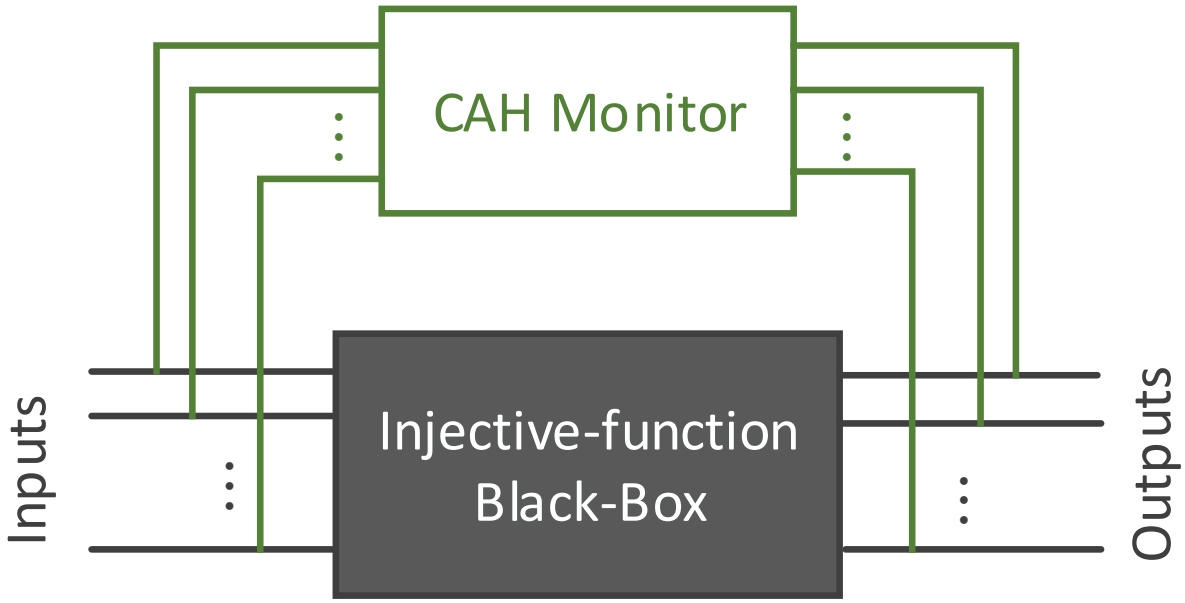
Summary

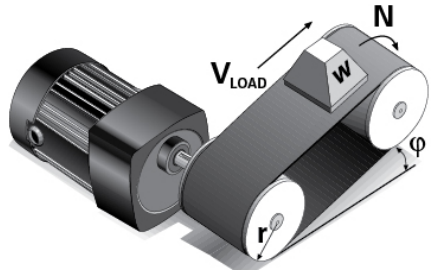
- Distance DS in the symbol space is defined based on the derivation process of symbols from time series;
- DS is provably a lower bound of D:

$$DS(\hat{Q}, \hat{C}) \leq D(Q, C) \quad \forall Q, C$$

- Tightness is reasonable in many practical cases, but theoretically unbounded.

Context Aware Health Monitoring of an AC Motor



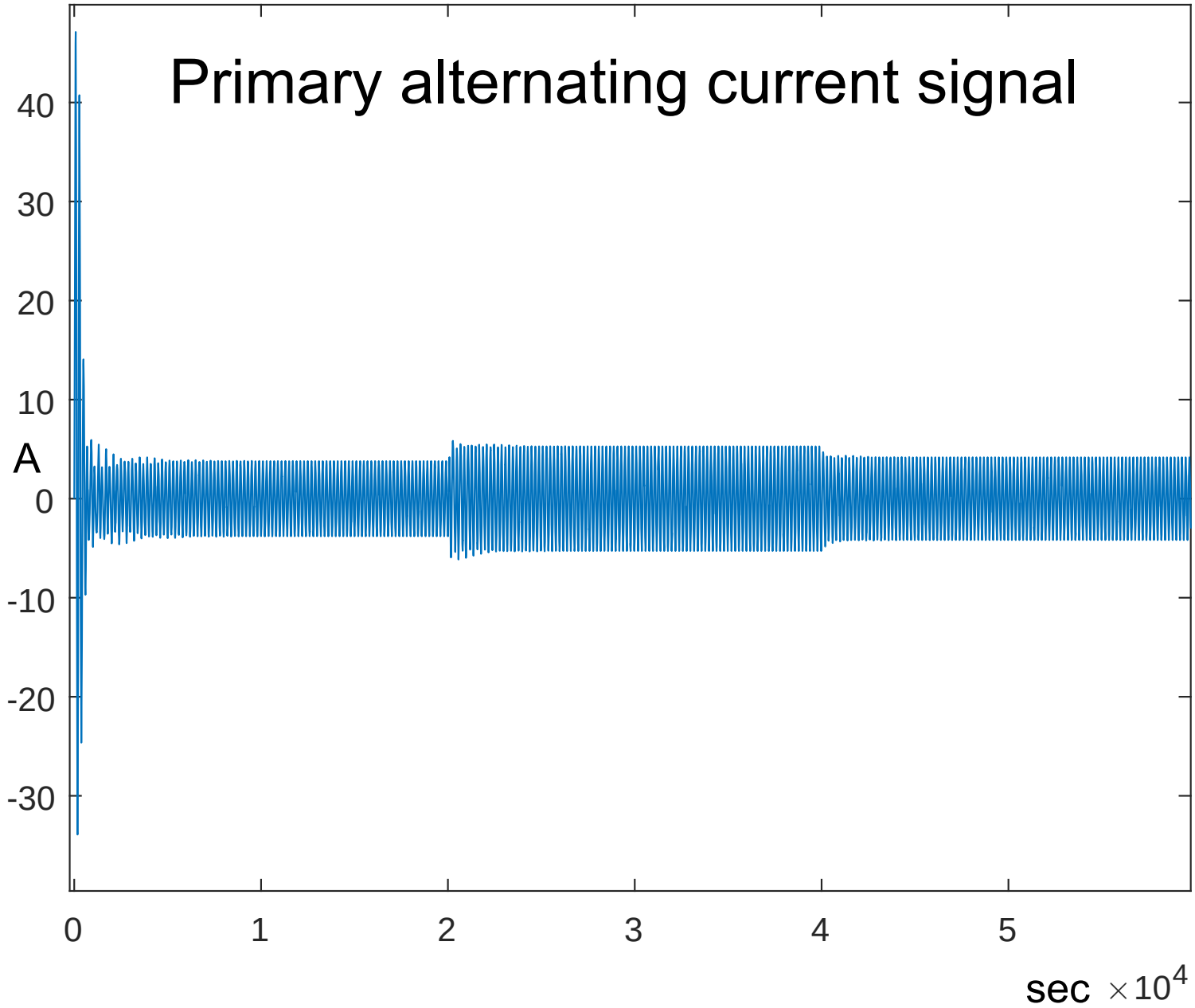


CAH Features

- No Model and minimal assumptions about the system
- Main assumption: injective function
- States are automatically inferred and learned
- Anomalies are detected when injectivity is violated

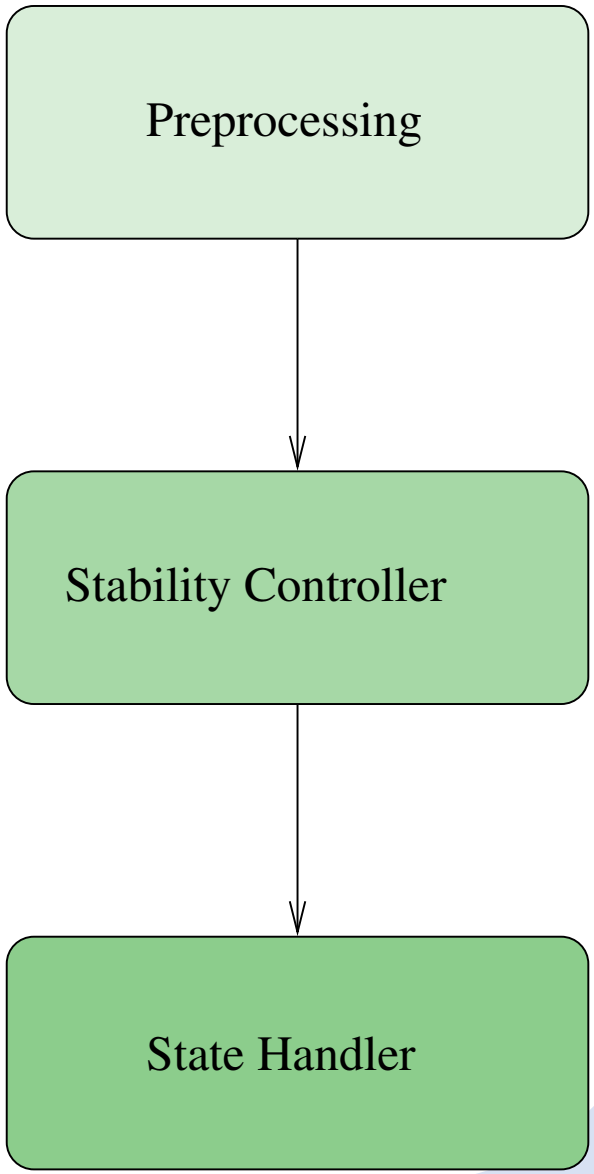
M. Götzinger, N. TaheriNejad, H. A. Kholerdi, and A. Jantsch. “On the design of context-aware health monitoring without a priori knowledge; an AC-Motor case-study”. In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Apr. 2017, pp. 1–5

CAH - Input Signal

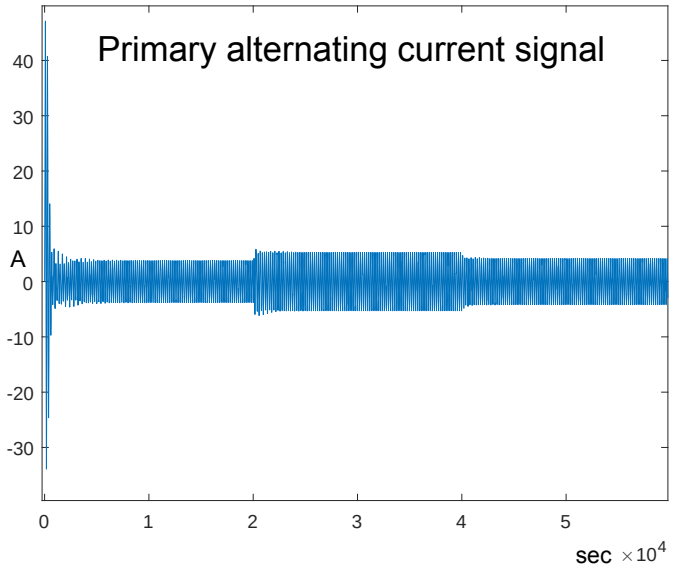


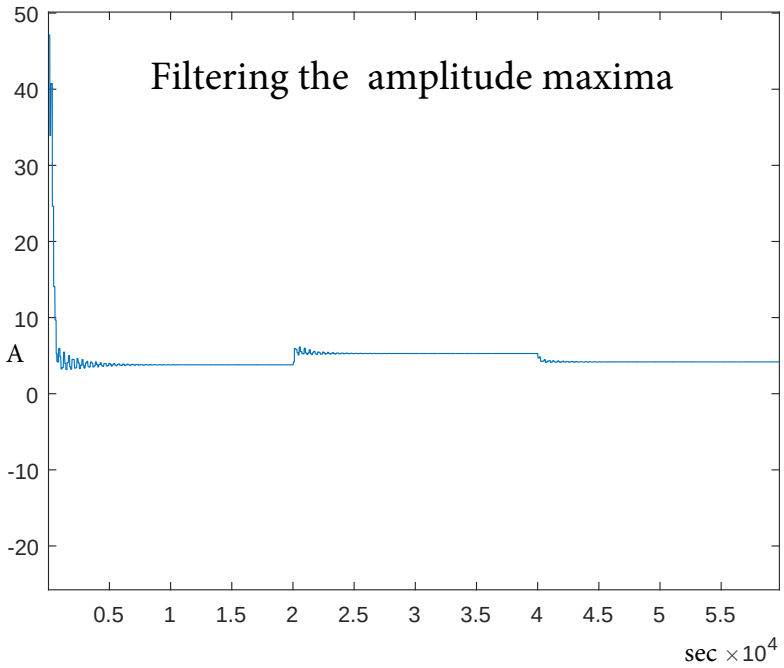
CAH System Architecture

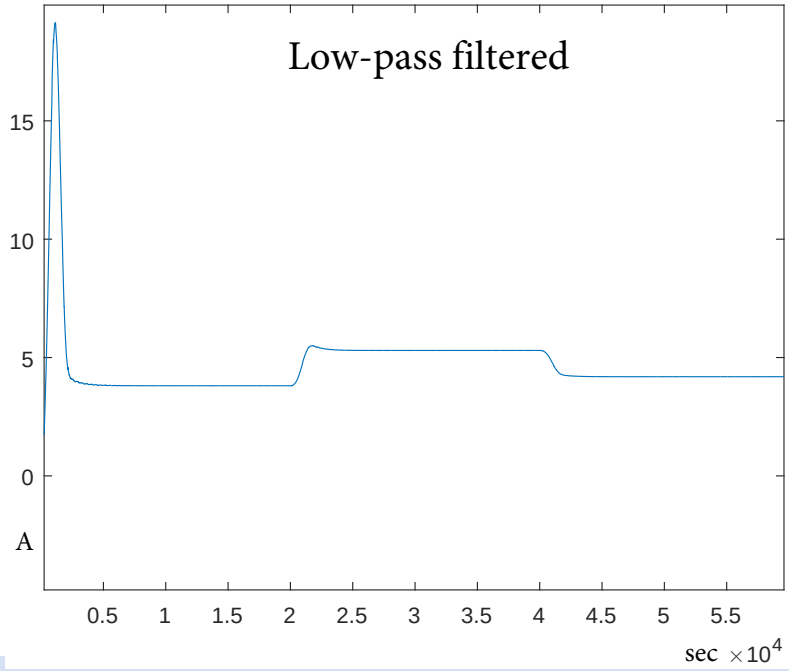
- Input signals:
  - Voltage
  - Frequency
  - Mechanical torque
- Output signals:
  - Current
  - Motor Speed
  - Electrical torque
  - Vibrations



CAH Preprocessing







CAH Stability Abstraction

- The system is stable if all n signals $s_1, \cdots, s_n$ are stable.
- A signal s_i is stable if a new sample of that signal s_i' is "similar" to the samples in a sliding history window.
- Two samples s_i' and s_i'' of the signal s_i are similar to a each other if $d(s_i', s_i'') \leq D_{\text{Similarity}}$

$$d(s_i', s_i'') = \frac{s_i' - s_i''}{s_i'}$$

- A signal s_i is stable if s_i' is similar to at least N_{Similar} stored samples in the history window.

W ... window size
$D_{\text{Similarity}}$ ... Similarity distance threshold
N_{Similar} ... Similarity count threshold

CAH State Abstraction

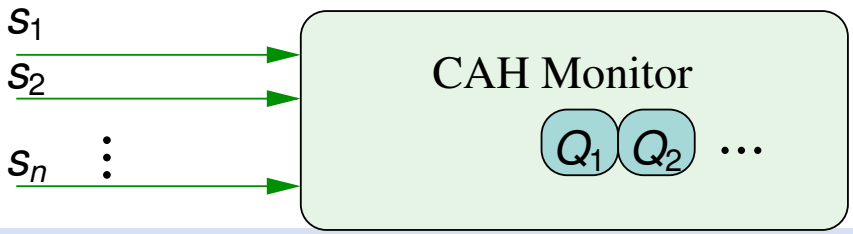
- A state $Q = (Q_{s_1}, Q_{s_2}, \cdots, Q_{s_n})$ is defined by the average values Q_{s_i} of samples of the signals s_i that have been identified to belong to the state.
- The distance of a new sample s'_i to a state Q is defined as

$$d(s'_i, Q) = \frac{s'_i - Q_{s_i}}{s'_i}$$

- If

$$d(s'_i, Q) \leq D_{\text{State}}$$

for all s_i that defines the state Q the new sample set $(s_1, \cdots, s_n)$ is considered to belong to state Q.



CAH State Abstraction

- If a new sample set s'_i belongs to the currently active state Q, the state remains active and the average values Q_{s_i} are updated.
- If the new sample set does not belong to the active state, a state change is considered.
- The new state is *normal* if both input and output signals have changed, otherwise it is an error.
- If the number of samples in a state is $\geq N_{Samples}$ the state is considered *valid*, otherwise it is *transient*.

D_{State} ... State similarity threshold
$N_{Samples}$ ... State size threshold for valid states

CAH Drift Detection

- For each state the average sample values of two sliding windows called *Digital Average Block (DAB)*, are maintained, DAB_1 and DAB_2.
- DAB_1 maintains average of samples $s_i^{t-1}, \cdots, s_i^{t-W_{\text{DAB}}}$ and DAB_2 maintains average of samples $s_i^{t-W_{\text{DAB}}-1}, \cdots, s_i^{t-2 \cdot W_{\text{DAB}}}$.
- The average for signal s_i in DAB_k is denoted as $\text{DAB}_k(s_i)$.
- If for any signal s_i we have

$$\frac{\text{DAB}_1(s_i) - \text{DAB}_2(s_i)}{\text{DAB}_1(s_i)} \geq D_{\text{Drift}}$$

the signal is considered to be drifting.

D_{Drift} ... Drift threshold
W_{DAB} ... DAB window size

CAH - Parameter Settings

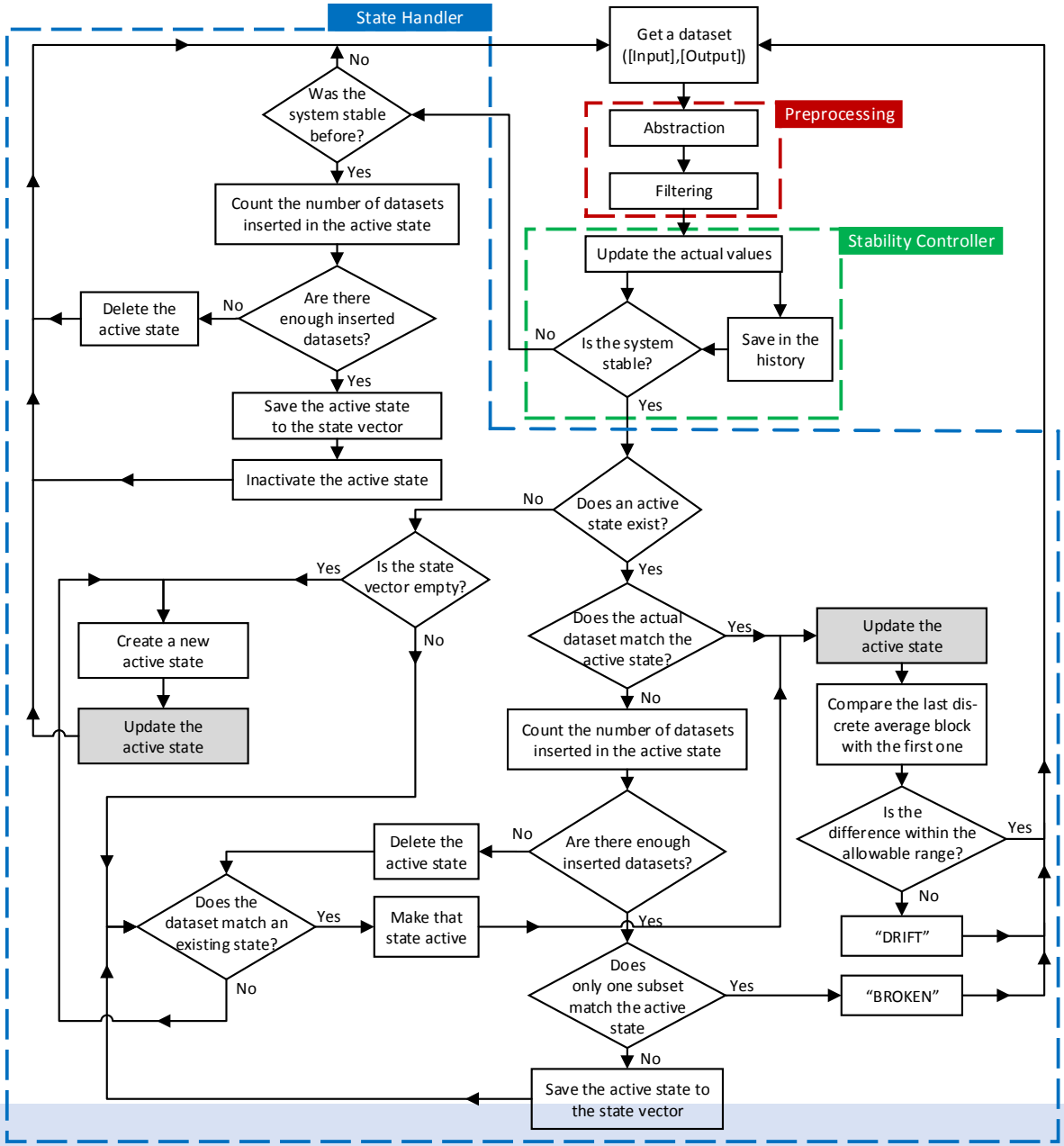
3	$\leq$	W			Window size
3	$\leq$	N_{Similar}	$\leq$	W	Similarity count threshold
0.01	$\leq$	$D_{\text{Similarity}}$	$\leq$	0.08	Similarity distance threshold
0.11	$\leq$	D_{State}	$\leq$	0.19	State similarity threshold
3	$\leq$	W_{DAB}			*DABwindowsize*
0.11	$\leq$	D_{Drift}	$\leq$	0.19	Drift threshold

Empirical rules based on case studies on AC Motor and Hydraulic pipe systems

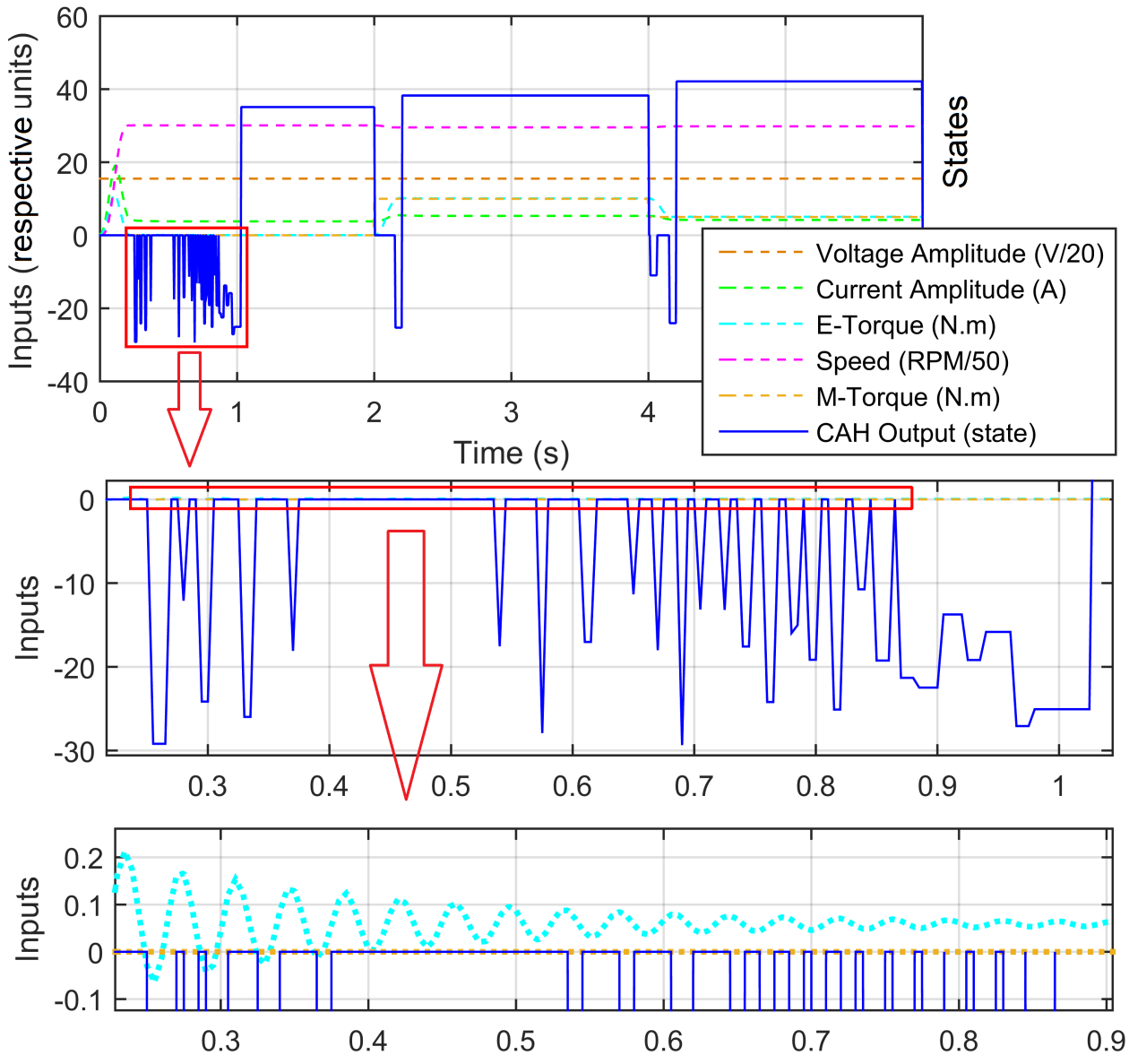
M. Götzinger, N. TaheriNejad, H. A. Kholerdi, and A. Jantsch. "On the design of context-aware health monitoring without a priori knowledge; an AC-Motor case-study". In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Apr. 2017, pp. 1–5

Maximilian Götzinger et al. "Applicability of Context-Aware Health Monitoring to Hydraulic Circuits". In: *The 44th Annual Conference of the IEEE Industrial Electronics Society*. 2018

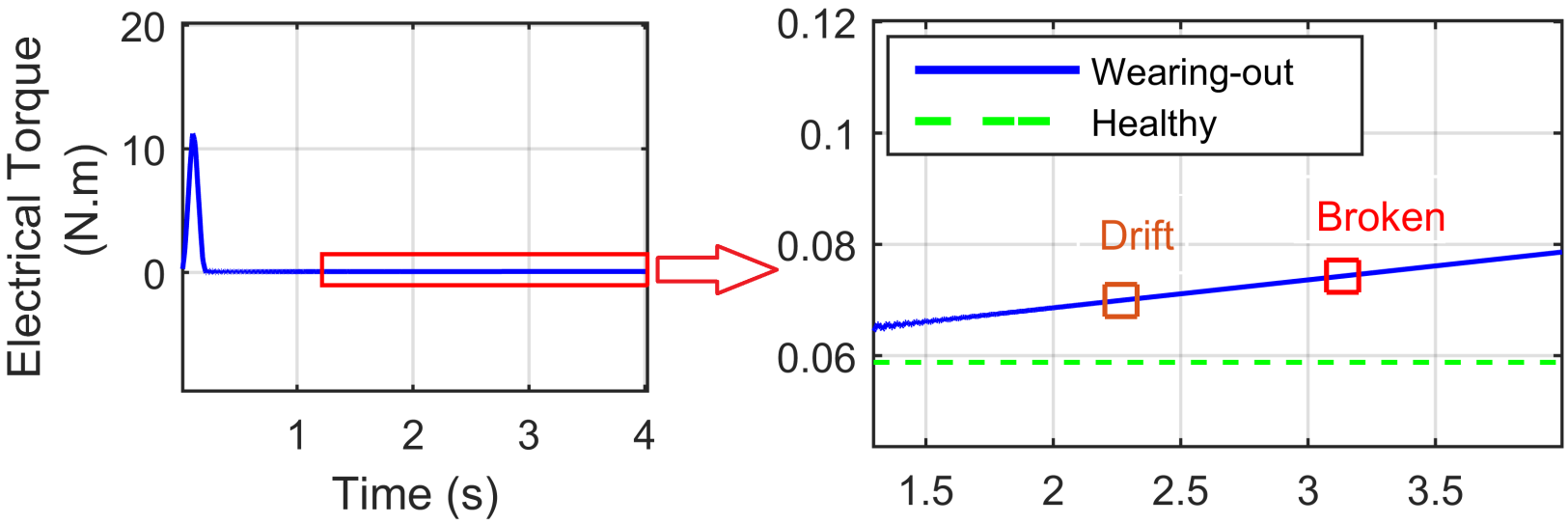
CAH State Handler



CAH Normal Mode



CAH Anomaly Detection



CAH Summary

Context-Aware Health monitoring (CAH) motivated by:

- The need for low cost health monitoring
- Suitable for resource and data limited scenarios.
- Assumptions on the black-box:
  - Injective Function,
  - Steady state behavior.
- In the AC motor case study CAH successfully recognized
  - normal states,
  - state changes,
  - drift, and
  - broken behavior..

Confidence

Classification with Confidence

- Multi-Classifier System
- SVM based Quality Estimation
- Iterative CNNs with Quality Estimation

Multi-Classifier System

- Deployment of several classification algorithm.
- Self-assessment (**confidence**) of each algorithm.
- Confidence based control of classification.
- Improves robustness and works well for small training sets.

Hedyeh A. Kholerdi, Nima TaheriNejad, and Axel Jantsch. “Enhancement of Classification of Small Data Sets Using Self-awareness - An Iris Flower Case-Study”. In: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence, Italy, May 2018

Iris Flower Classification

By Diliff - Own work, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=33037509

- UCI Iris Flower Dataset, with 150 labeled images
- Each to be classified in one of three classes of iris plants.

Iris Flower Classification Algorithms

Neural Network

Support Vector Machine

Naive Bayesian

Setosa

Versicolor

Virginica

Setosa

Versicolor

Virginica

Setosa

Versicolor

Virginica

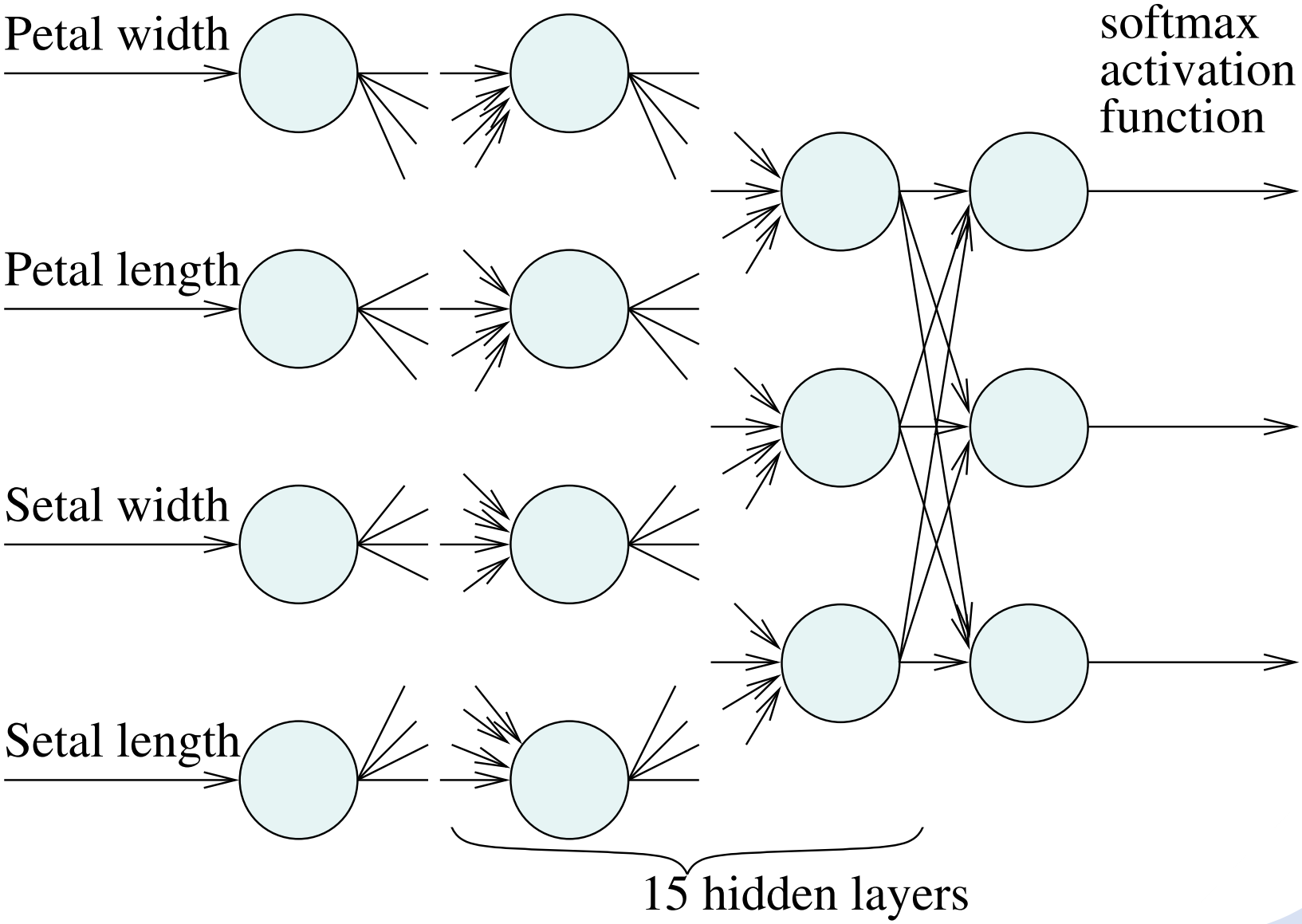
Confidence Metric

Confidence c of an algorithm A_j for a particular classification of sample x_i to class k_l is the probability that the classification is correct:

$$c\left(A_j(x_i, k_l)\right) = p\left(E_{x_i}^{A_j} == T_{x_i}^{k_l}\right)$$

- x_i ... Data sample
- k_l ... Class
- A_j ... Algorithm j
- $T_{x_i}^{k_l}$ ... the True Class of a sample x_i (Ground Truth)
- $E_{x_i}^{A_j}$ ... the class estimated by algorithm A_j for x_i

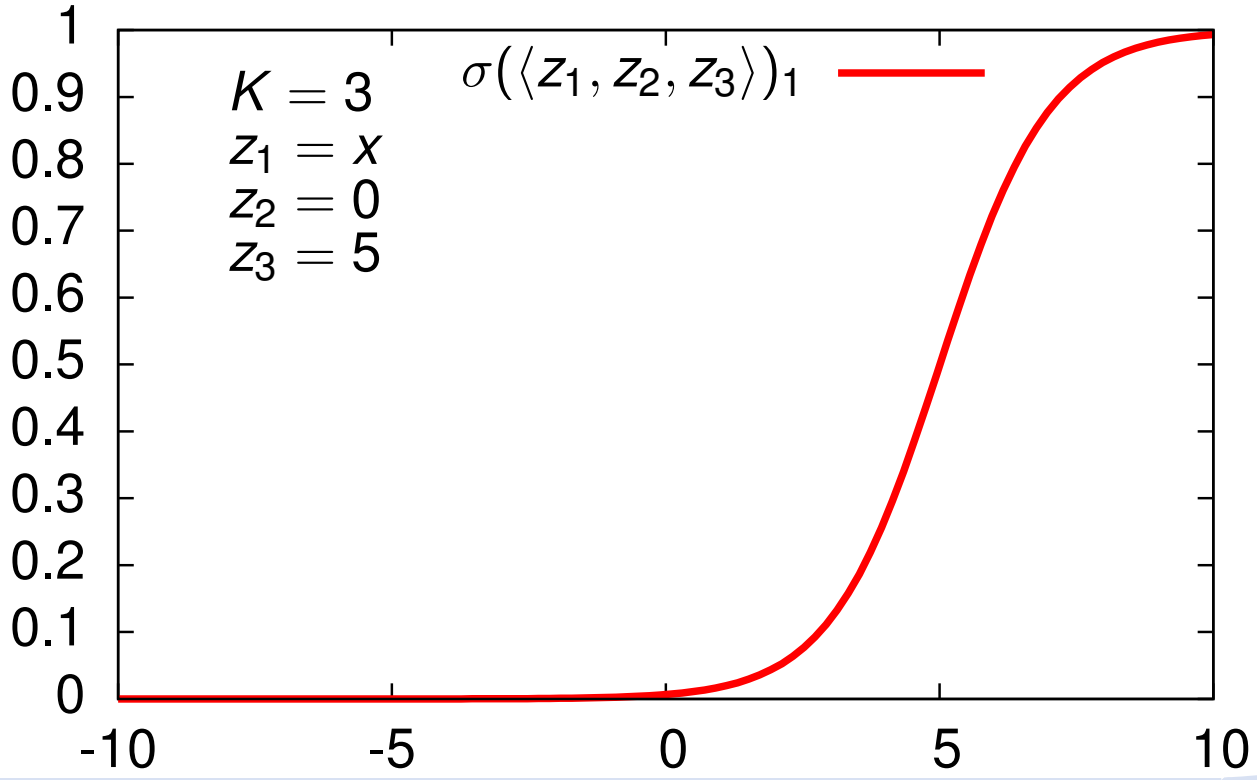
Confidence of a Neural Network Classifier I



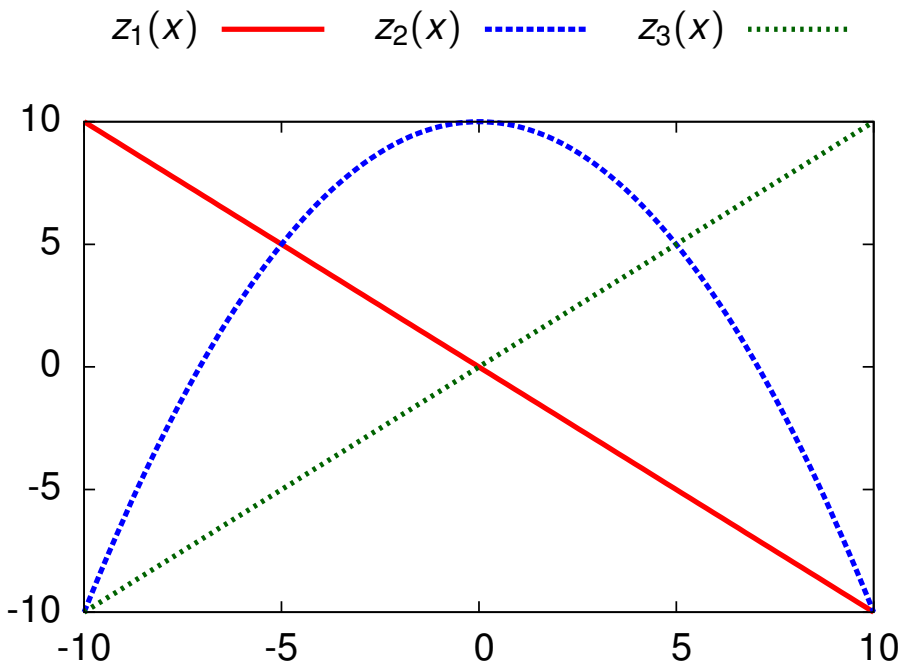
Confidence of a Neural Network Classifier II

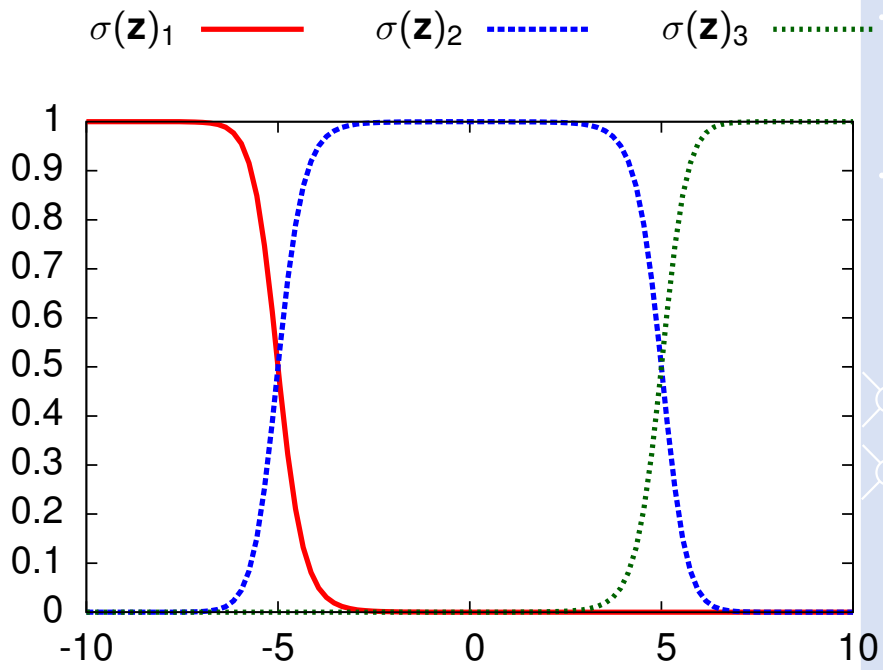
The softmax function is a normalized exponential function that computes a vector with values in $[0, 1]$ and with the sum $= 1$.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1 \ldots K$$



Confidence of a Neural Network Classifier III





Neural Network classification output and corresponding probabilities.

Confidence of a Naïve Bayesian Classifier I

$$P(k|x) \propto P(k) \prod_{i=1}^{F} P(f_i|k)$$

$x = \langle f_1, \ldots, f_F \rangle$	... given input feature set
k	... category
$P(k\|x)$	... posterior probability that x falls into class k
$P(k)$	... prior probaility of class k
F	... number of features
f_i	... feature
$P(f_i\|k)$	... conditional probability that feature f_i occurs in an input falling into class k

Confidence of a Naïve Bayesian Classifier II

Naïve Bayesian classifier for iris flowers:

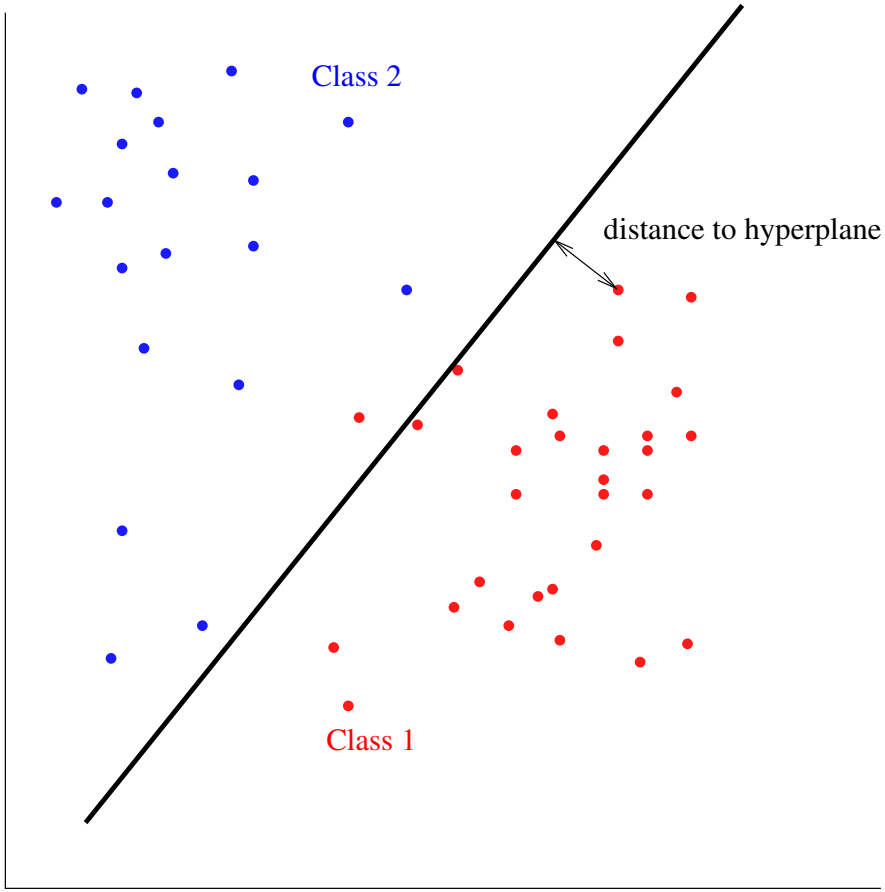
Input feature set:
$x = \langle \text{petal width}, \text{petal length}, \text{setal width}, \text{setal length} \rangle$

Categories: $\langle \text{Setosa}, \text{Versicolor}, \text{Virginica} \rangle$

$$c\Big(A_{\text{NB}}(x_i, k_l)\Big) = p\Big(E_{x_i}^{A_{\text{NB}}} == T_{x_i}^{k_l}\Big) = P(k_l|x_i)$$

Confidence of Support Vector Machine Classifier

SVM based classifier constructs hyper planes to separate categories.



The normalized distances of inputs x are used as confidence.

Confidence Metric

Confidence c of an algorithm A_j for a class k_l is:

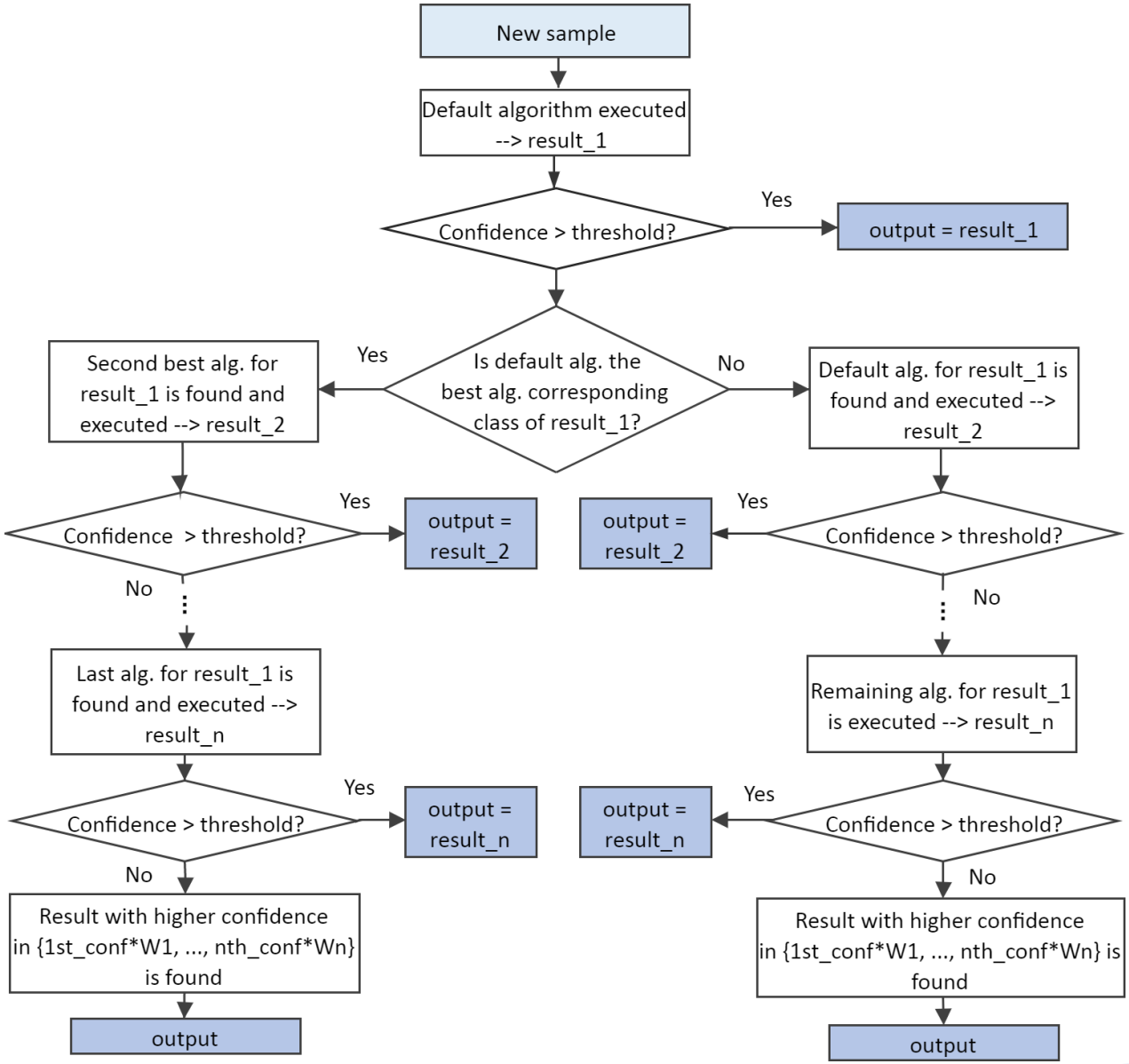
$$c\left(A_j^{k_l}\right) = \frac{1}{n}\sum_{i=0}^{n} c\left(A_j(x_i, k_l)\right)$$

Confidence of an algorithm A_j for all m classes is:

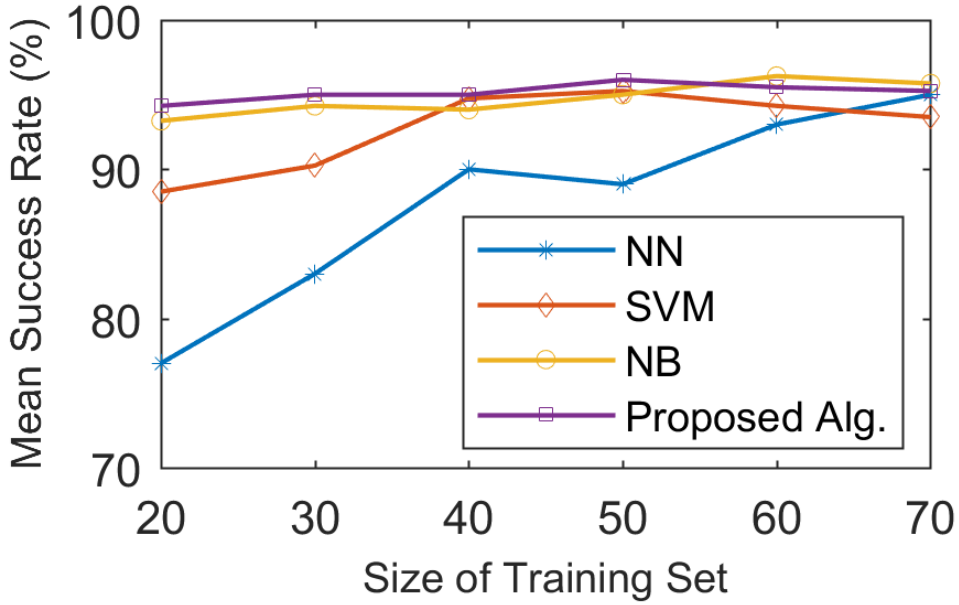
$$C_{A_j} = \frac{1}{m}\sum_{k_l=0}^{m} c\left(A_j^{k_l}\right).$$

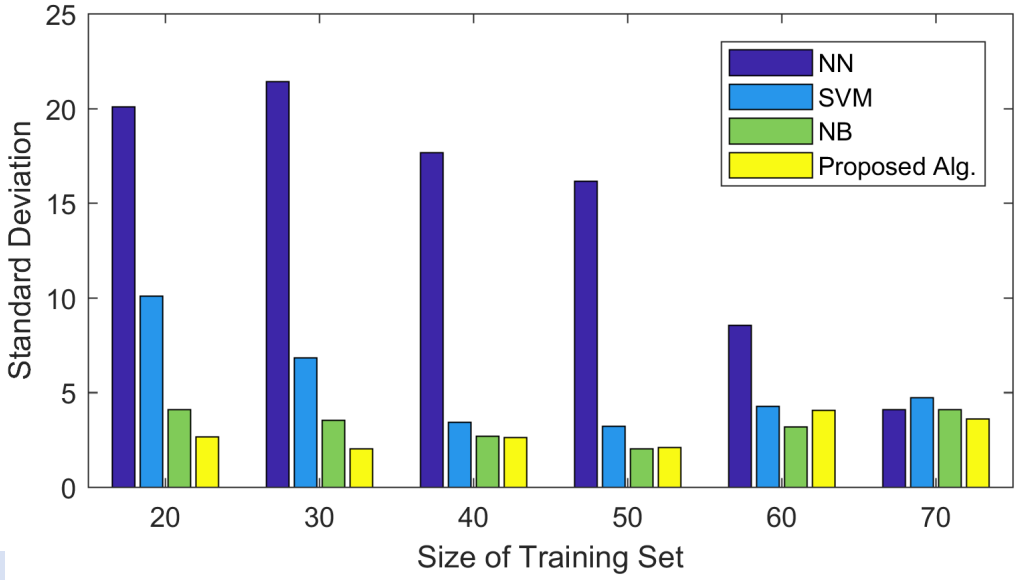
The default algorithm is the one with highest overall confidence.

Confidence Based Classification



Classification Success Rate





Multi-Classifier System

Summary

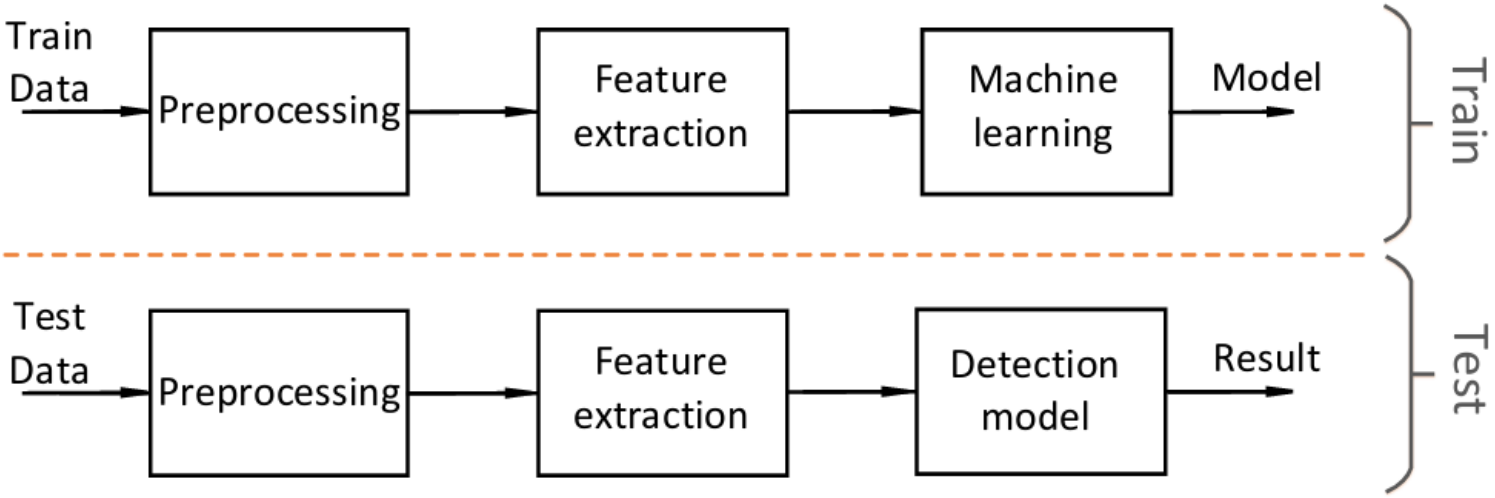
- Self-assessment, expressed as a **confidence metric** improves classification.
- It is superior to each individual algorithm.
- It is robust.
- It works well for small training sets.
- Overhead:
  - *n* classification algorithms are implemented and trained.
  - 27% run-time overhead in this case study.

Epileptic Seizure Detection I

- Single lead Electrocardiogramphy (ECG)
- Three SVMs used:
  - Simple SVM Model
  - Full SVM Model
  - Confidence SVM Model

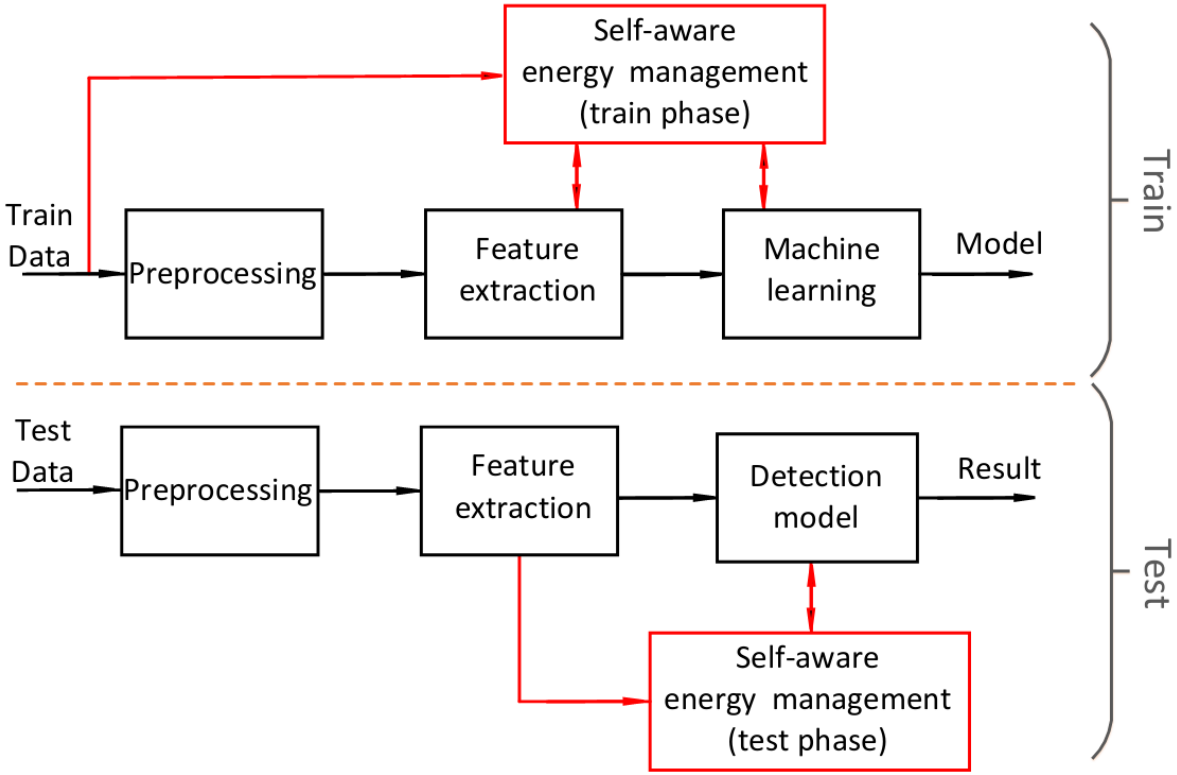
Farnaz Forooghifar, Amir Aminifar, and David Atienza Alonso. “Self-Aware Wearable Systems in Epileptic Seizure Detection”. In: *21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic, August 29-31, 2018*. 2018, pp. 426–432

Epileptic Seizure Detection II



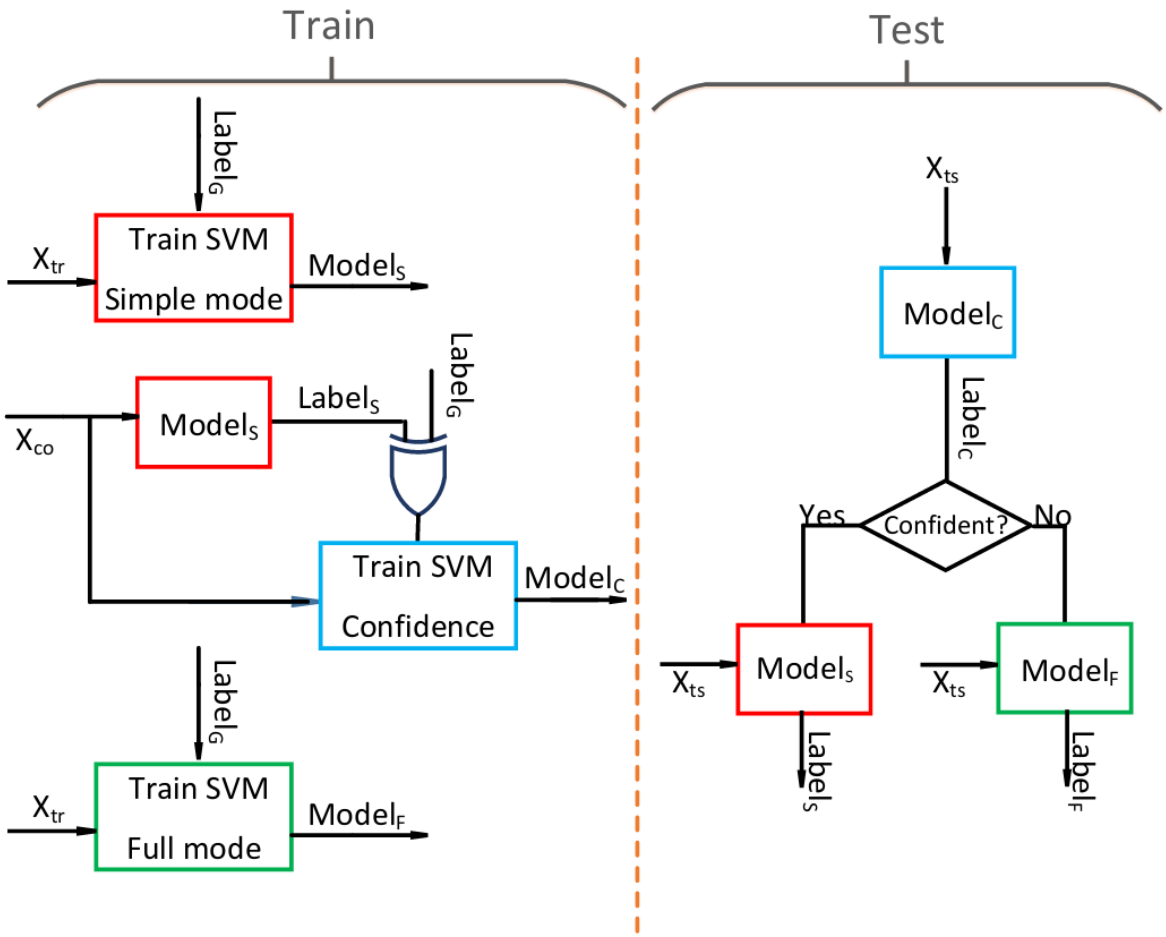
Traditional wearable system for monitoring pathological health conditions

Epileptic Seizure Detection III



Self-aware enhanced ECG monitoring

Epileptic Seizure Detection IV



Epileptic Seizure Detection V

Results

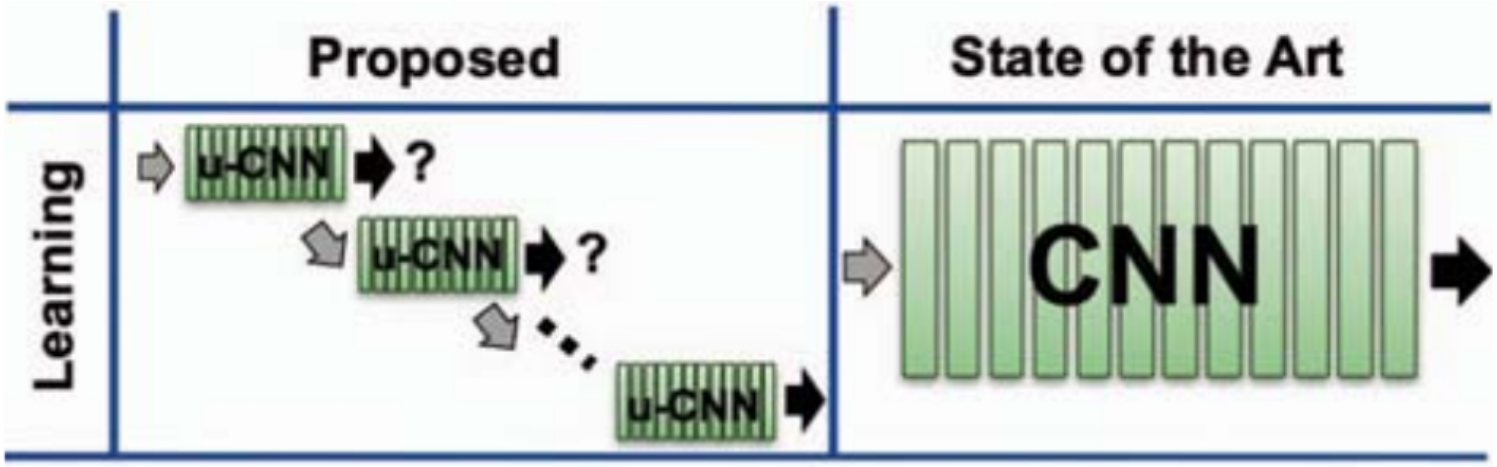
- Simple SVM model:
  - uses subset of features
  - 30ms execution time
  - Invocation in 37% of cases
- Full SVM Model:
  - uses full feature set
  - 840ms execution time
  - Invocation in 63% of cases
- Confidence SVM model:
  - uses full feature set
  - few ms execution time
- Overall performance:
  - 540ms execution time (37% reduction)
  - 89.4% correct predictions (vs 88.7% with only the full SVM model)

ICNN: Iterative Convolutional Neural Network I

- The CNN AlexNet is broken into a set of μCNNs
- The μCNNs are sequentially executed followed by an estimate of prediction accuracy (confidence)
- if confidence < lower threshold $\rightarrow$ stop and fail;
- if confidence > upper threshold $\rightarrow$ stop and classify;
- otherwise continue with next μCNN

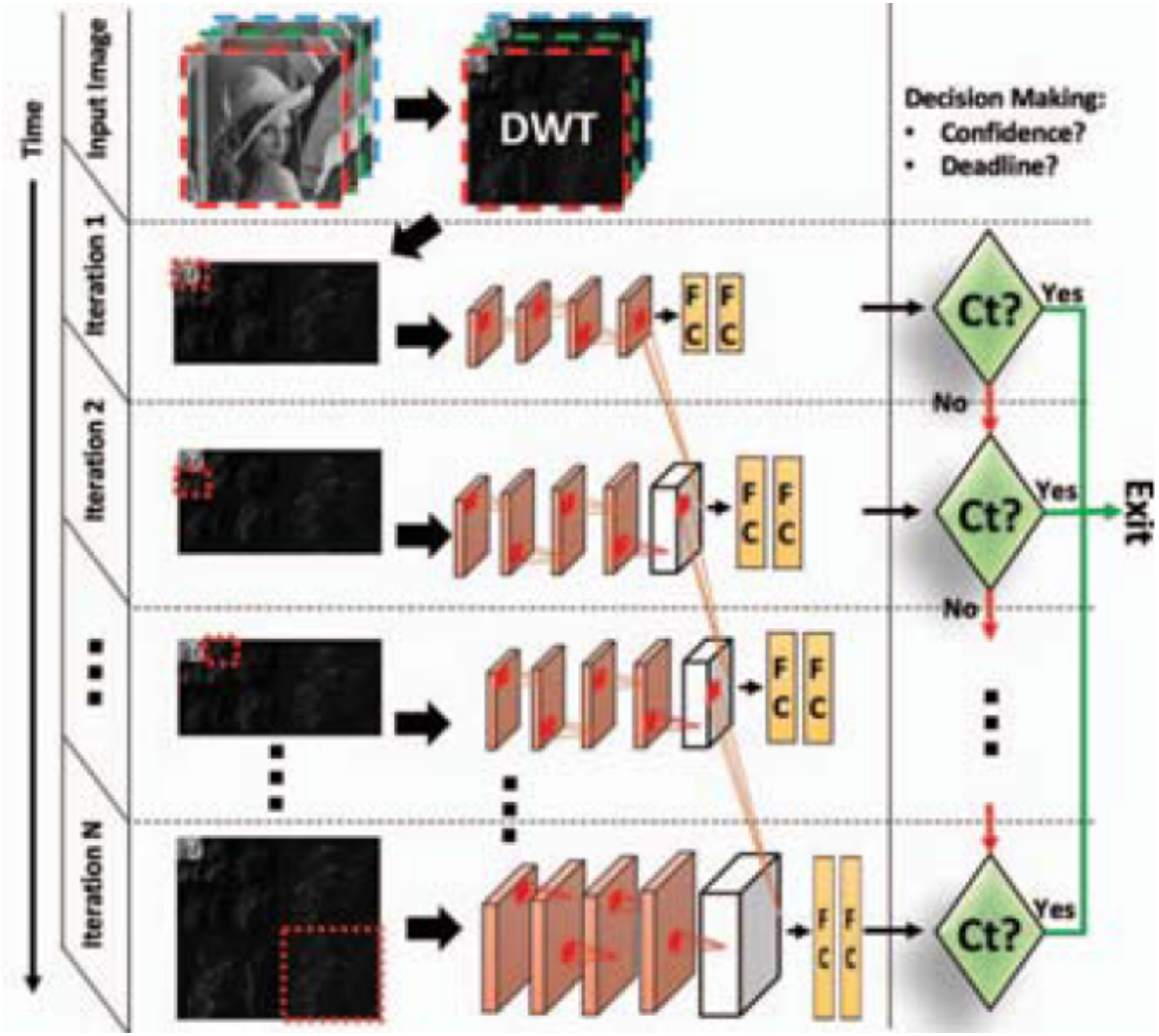
Katayoun Neshatpour, Farnaz Behnia, Houman Homayoun, and Avesta Sasan. "ICNN: An iterative implementation of convolutional neural networks to enable energy and computational complexity aware dynamic approximation". In: *2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018*. 2018, pp. 551–556

ICNN: Iterative Convolutional Neural Network II



One large CNN is replaced by a sequence of μCNNs

ICNN: Iterative Convolutional Neural Network III

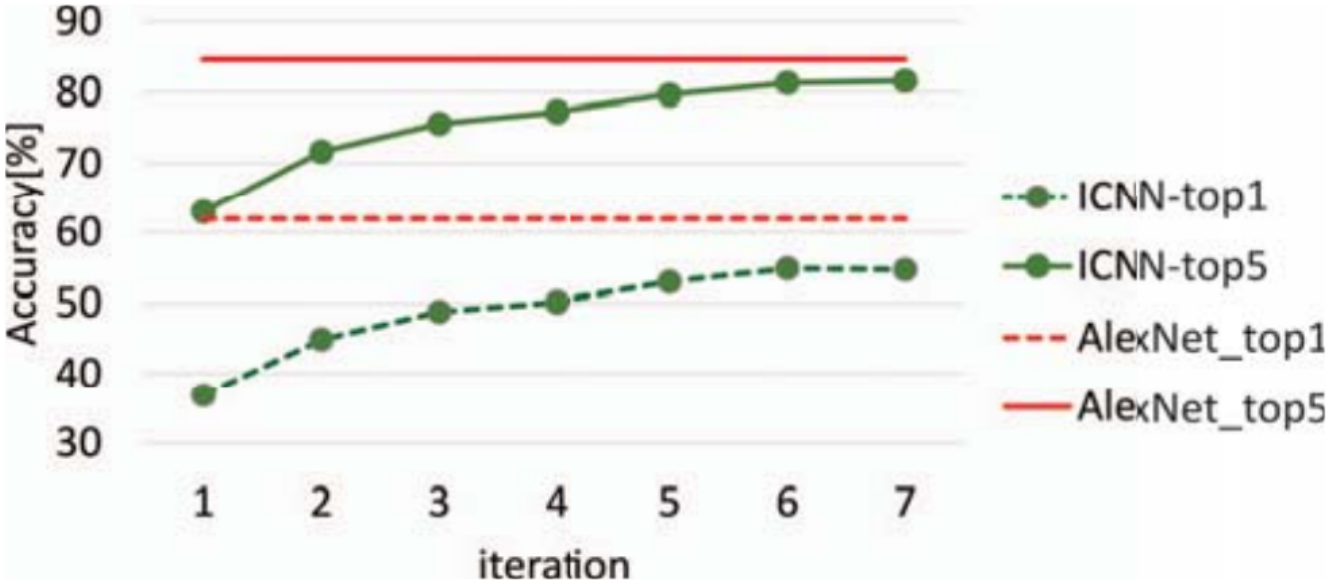


The computation can be stopped for various reasons.

ICNN: Iterative Convolutional Neural Network IV

Computational cost os always less than the original CNN.

ICNN: Iterative Convolutional Neural Network V



The accuracy iteratively approximates the original CNN's.

ICNN: Iterative Convolutional Neural Network VI

Summary

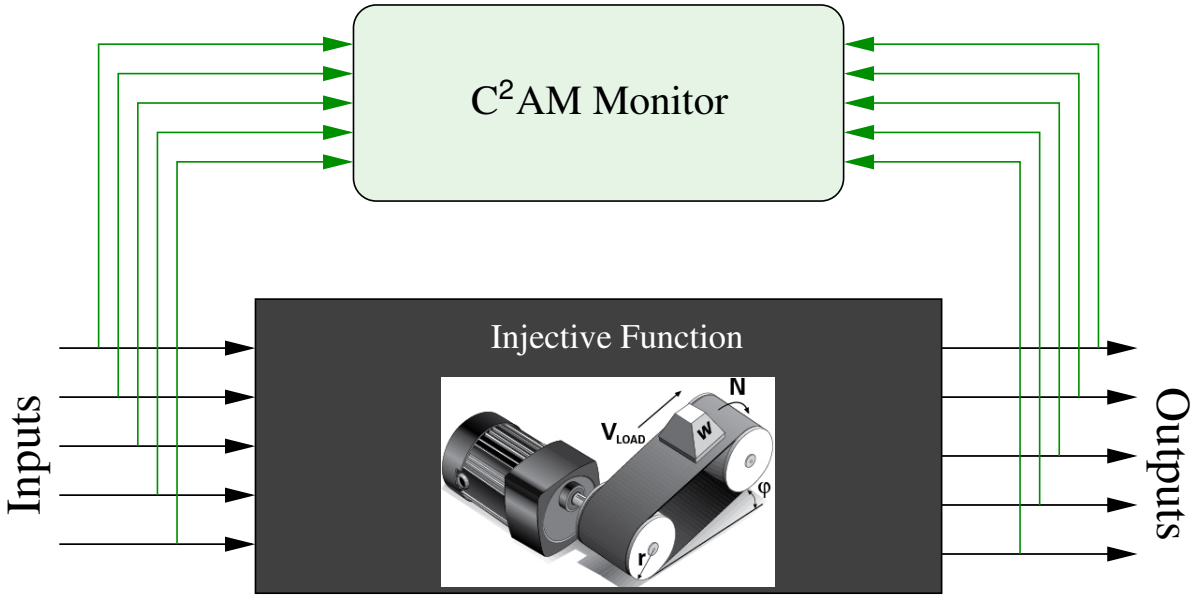
- The ICCN iteratively approximates the original CNN, both in therms of computation cost and accuracy;
- The trade-off between effort and accuracy can be made application and specification specific;
- The ICNN continuously assess its own performance.

Monitoring with Confidence

- Motivation
- Definition of Confidence, distance, and fuzzy logic based functions
- The CAM system, state handling
- AC Motor case study
- Hydraulic pipe system case study
- Conclusions

Maximilian Götzinger et al. “Model-free Condition Monitoring with Confidence”. In: *International Journal of Computer Integrated Manufacturing* 32.4-5 (2019)

Confidence Based Monitoring



Monitoring with Confidence

Confidence is defined as distance between ground truth f and an estimation g:

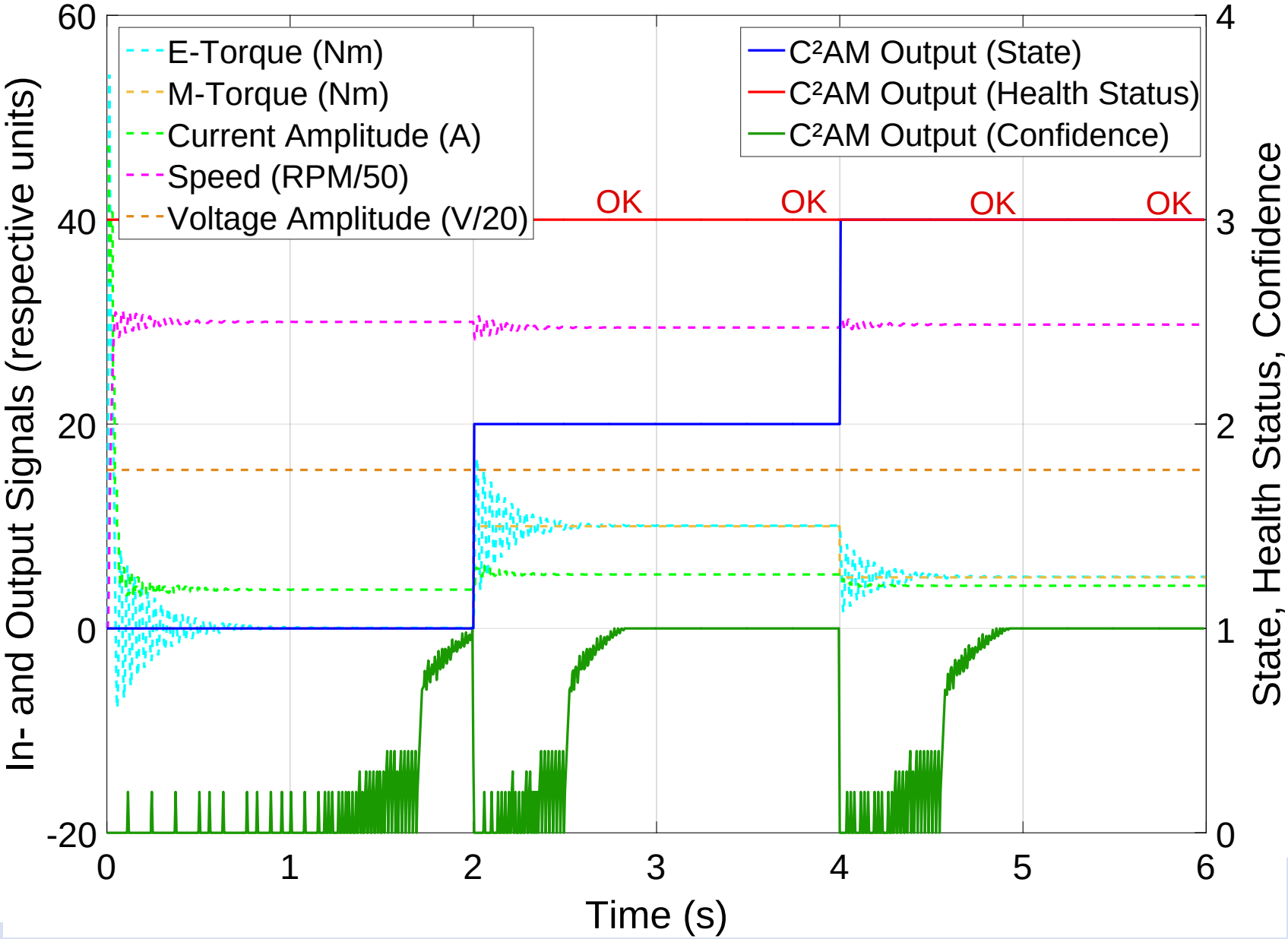
$$c(g(x_i)) = 1/\Big(\Delta\big(f(x_i), g(x_i)\big)\Big)$$

$$C(g) = \frac{1}{n}\sum_{i=0}^{n} c(g(x_i))$$

$$0 \leq c(.), C(.) \leq 1$$

$f(x)$ ... Ideal function
$g(x)$ ... Estimation or Approximation of f
Δ ... Appropriate distance metric

Sampling Data into States



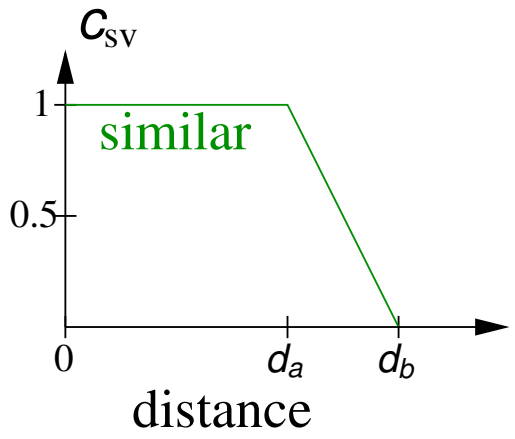
Sampling Data into States

$$d = \left| \frac{v_{\text{new}} - v_h}{v_{\text{new}}} \right|$$

v_{new} ... New data sample
v_h ... Data sample in history
d ... Relative distance between new and previous data points

Sampling Data into States

When is a new datum “similar” to a previously seen one?



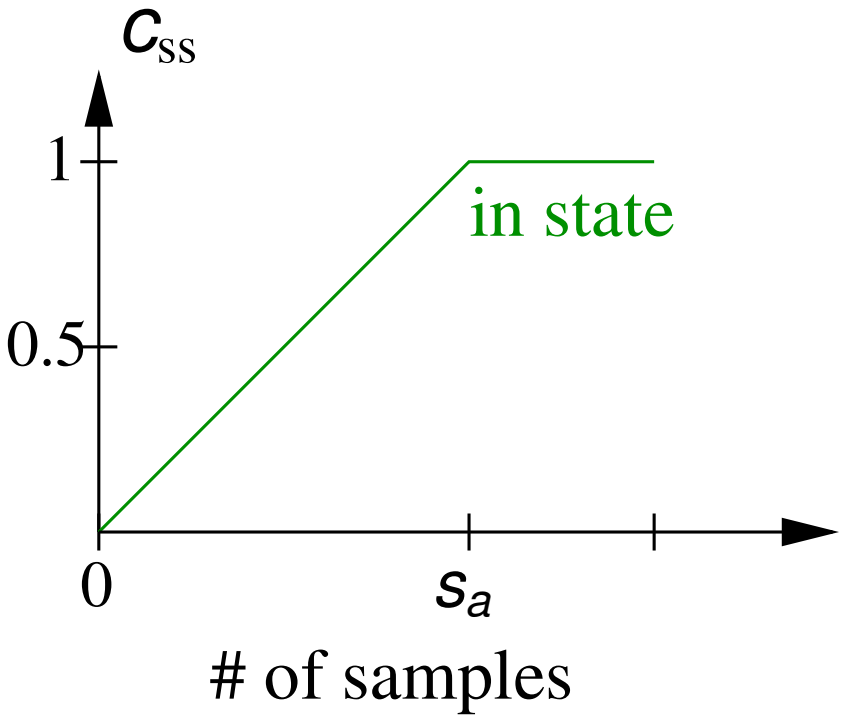
$$d = \left| \frac{v_{new} - v_h}{v_{new}} \right|$$

$$\frac{1}{\Delta'_{sv}} = c_{sv} = \begin{cases} 1 & \text{if } d \leq d_a \\ \frac{d - d_b}{d_a - d_b} & \text{if } d_a < d < d_b \\ 0 & \text{otherwise.} \end{cases}$$

Sampling Data into States

When does a new datum belong to an existing state?

- The state is defined by a set of previous data samples;
- The new datum is assigned to this state if sufficiently many data samples are in proximity.



$$\frac{1}{\Delta'_{ss}} = c_{ss} = \begin{cases} 1 & \text{if } k \geq s_a \\ \frac{k}{s_a} & \text{if } k < s_a \end{cases}$$

c_{ss} ... Confidence that a new data belongs to a data set

k ... No of samples similar to new datum

s_a ... Threshold

Sampling Data into States

What is the confidence that a new datum belongs to a given state?

$$c_b = (\bigwedge_{j=1}^{n} c_{sv}) \wedge c_{ss}$$

- c_b ... Confidence that a new data belongs to a state
- c_{sv} ... Proximity between data points
- c_{ss} ... Confidence that a new data belongs to a data set based on the number of similar data points in the state
- n ... No of samples in history

Identification of Stable States

A state is valid if the datapoints are sufficiently similar and there are sufficiently many similar points in the state.
Confidence that a state is valid:

$$c_{val} = c_{sl} \wedge c_{sz}$$

c_{val} ... Confidence that a state is valid
c_{sl} ... Proximity between data points
c_{sz} ... Confidence due to size of data in the state

Malfunctions I

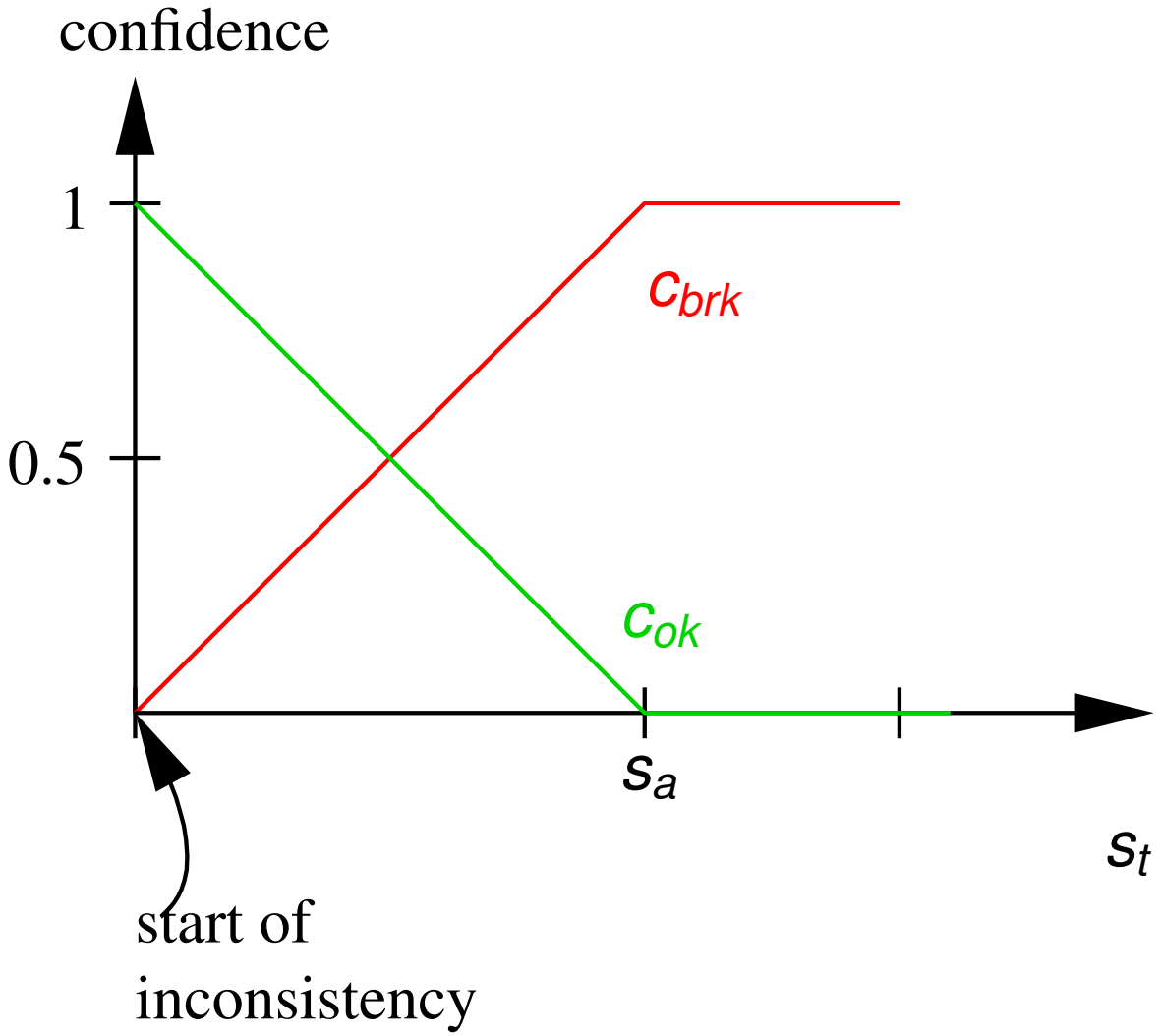
Assuming the system is a bijective function, input and outputs have to correlate.

$$c_{brk} = \begin{cases} 1 & \text{if } s_t \geq s_a \\ \frac{s_t}{s_a} & \text{if } 0 \leq s_t < s_a \end{cases}$$

$$c_{ok} = \begin{cases} 0 & \text{if } s_t \geq s_a \\ \frac{s_a - s_t}{s_a} & \text{if } 0 \leq s_t < s_a \end{cases}$$

c_{brk} ... Confidence that the system is broken
c_{ok} ... Confidence that the system is okay
s_a ... time constant depending on the type of system
s_t ... time elapsed after input-output inconsistency

Malfunctions II



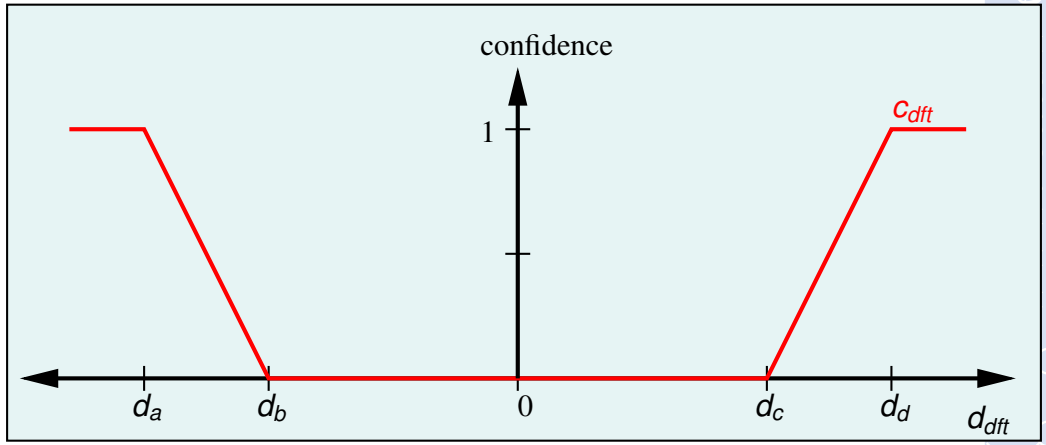
Drift Detection I

- Drifting values are a challenge for self-adapting monitoring without fixed, predefined thresholds.
- Drifting values are an important symptom for malfunctions or maintenance requirements.
- C^2AM stores *Discrete Average Blocks (DAB)* for each state.
- The first DAB of a state is compared with the latest DAB to detect a drift.

Drift Detection II

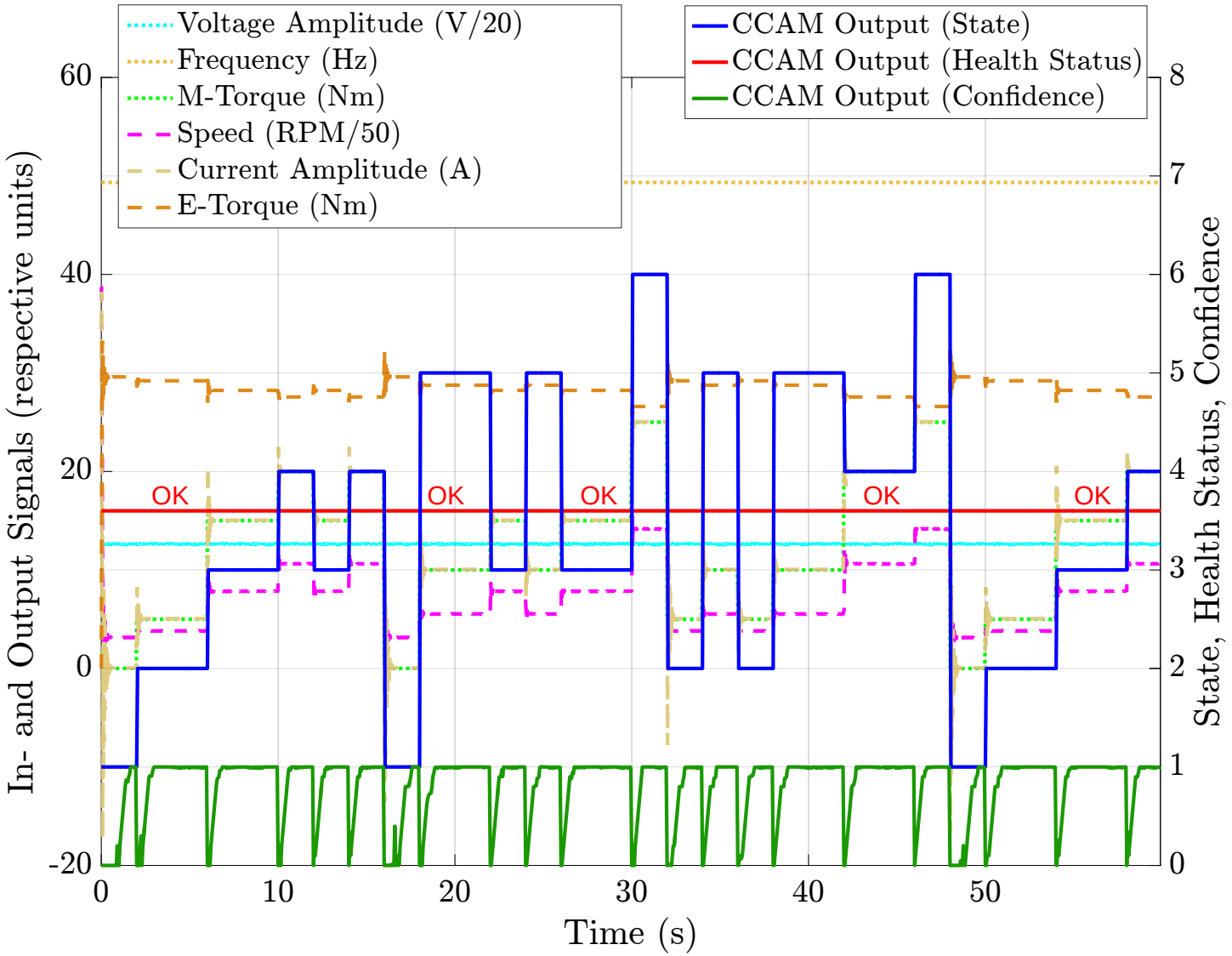
$$d_{dft} = \left| \frac{v_{avg,DAB_{first}} - v_{avg,DAB_{new}}}{v_{avg,DAB_{first}}} \right|$$

$$c_{dft} = \begin{cases} \frac{d_b - d_{dft}}{d_b - d_a} & \text{if } d_a < d_{dft} < d_b \\ 0 & \text{if } d_b \leq d_{dft} \leq d_c \\ \frac{d_{dft} - d_c}{d_d - d_c} & \text{if } d_c < d_{dft} < d_d \\ 1 & \text{otherwise} \end{cases}$$



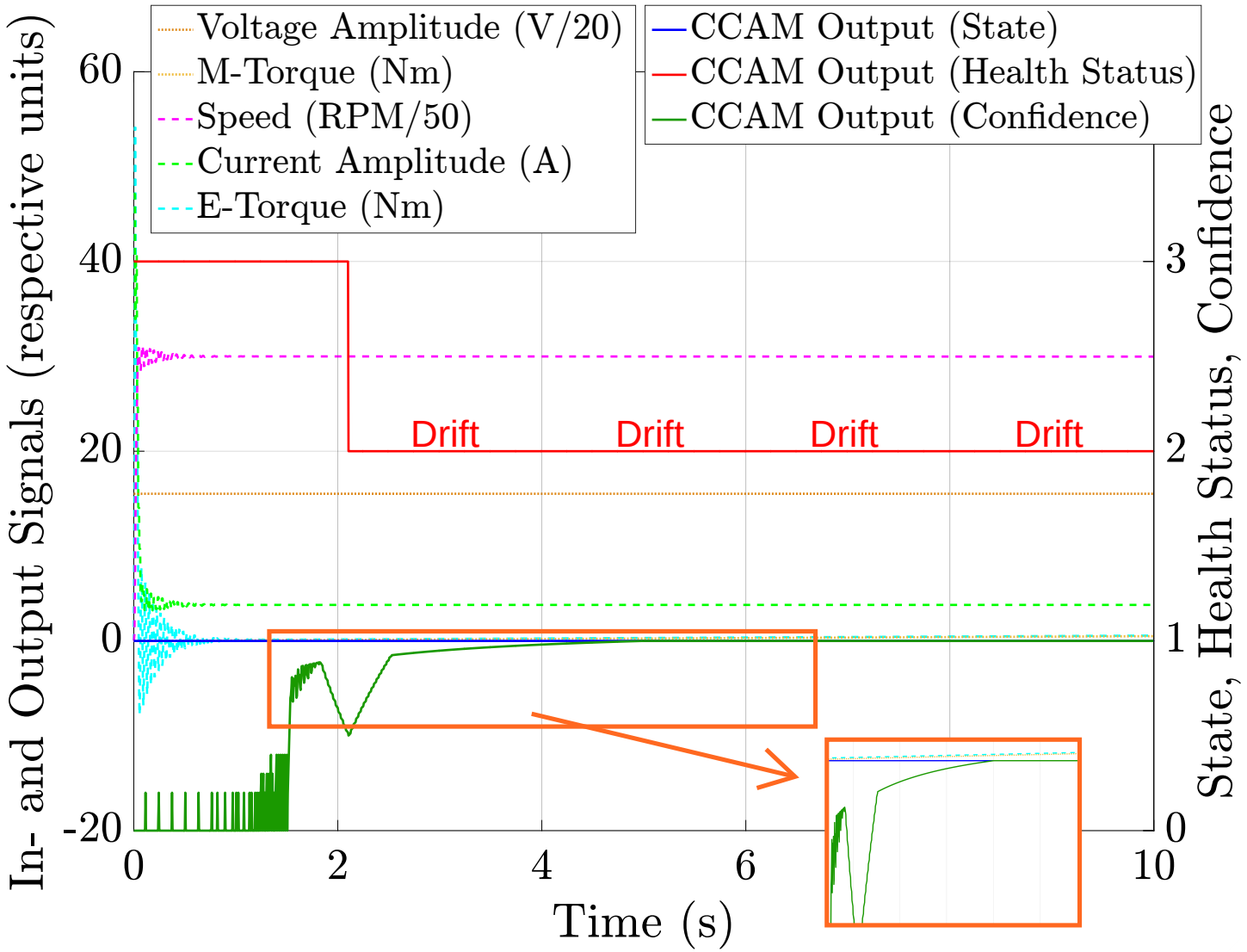
d_{dft} ... distance between first and latest DAB
$v_{avg,DAB}$ ... average of DAB
d_a, d_b, d_c, d_d ... constants
c_{dft} ... confidence that drift has occurred

C²AM Case Study: AC Motor I



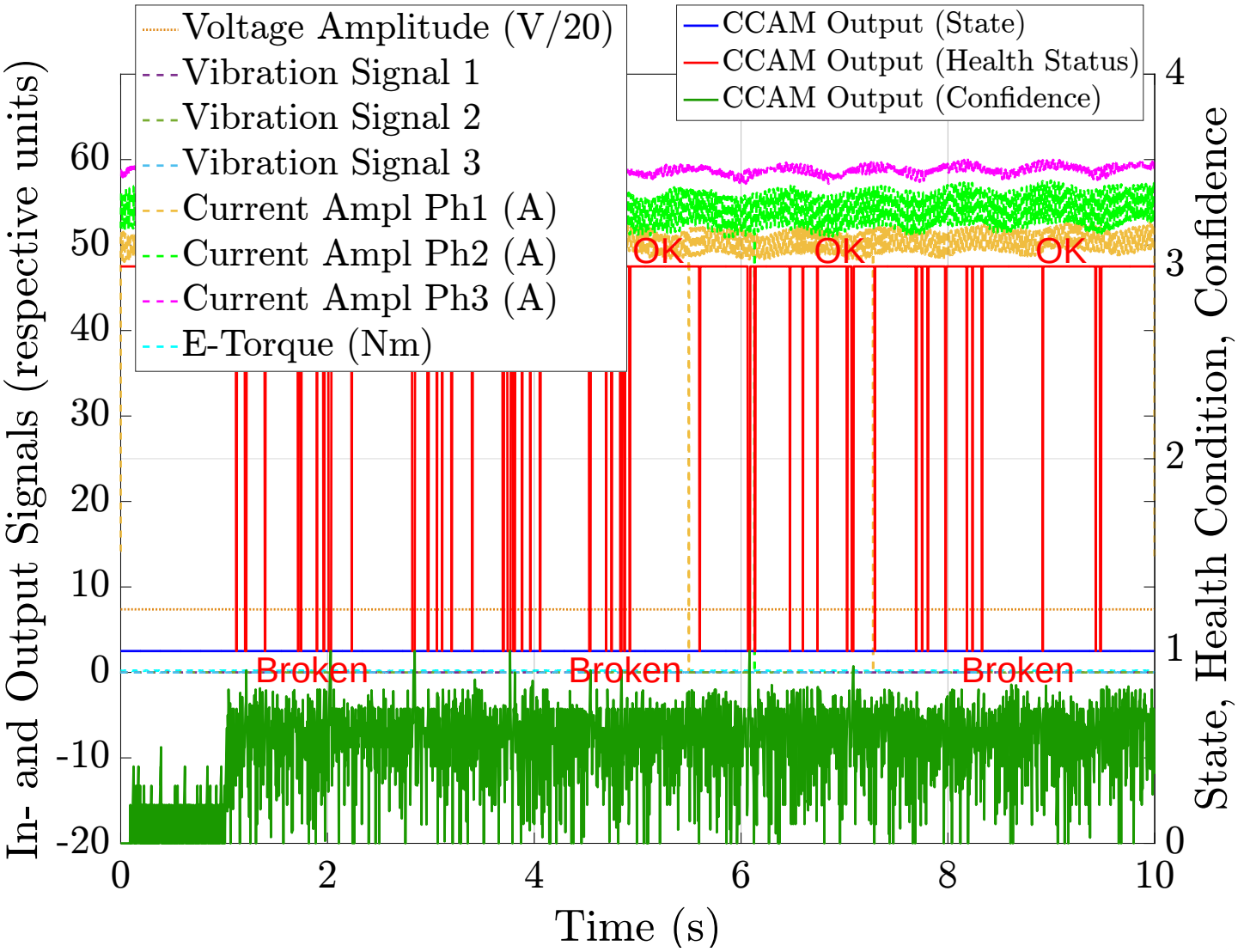
C²AM during many state changes of the AC motor

C²AM Case Study: AC Motor II



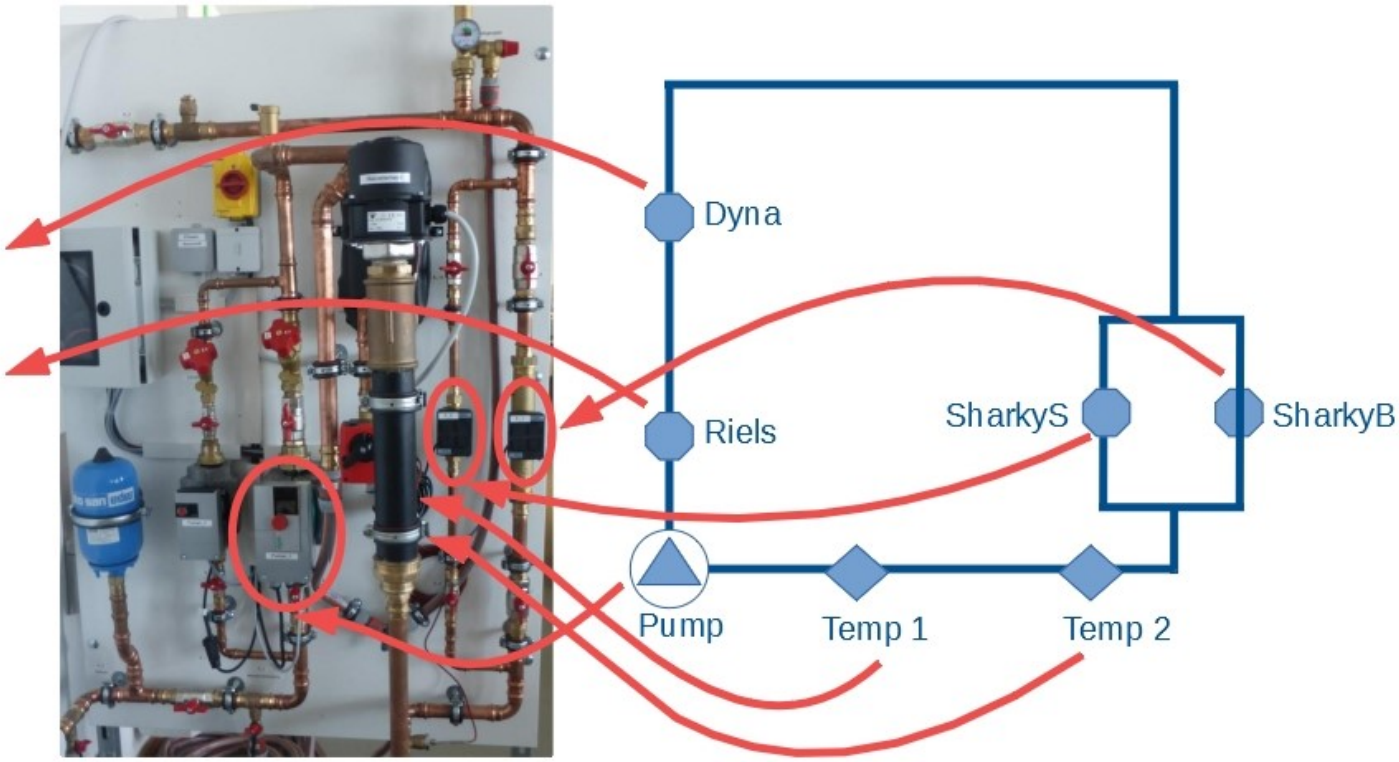
Drift observation

C²AM Case Study: AC Motor III

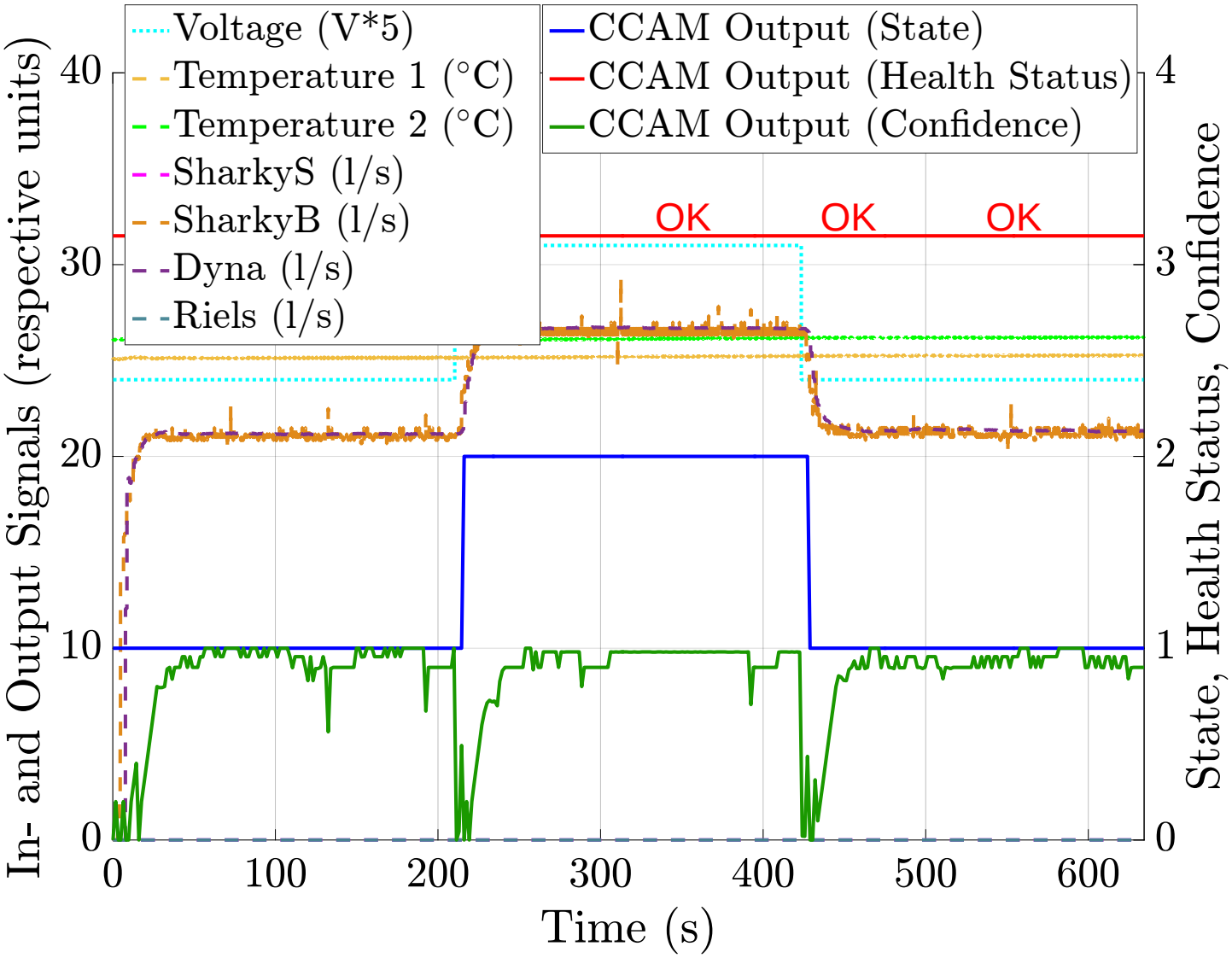


The AC Motor has a bearing defect

C²AM Case Study: Water Pipe System I

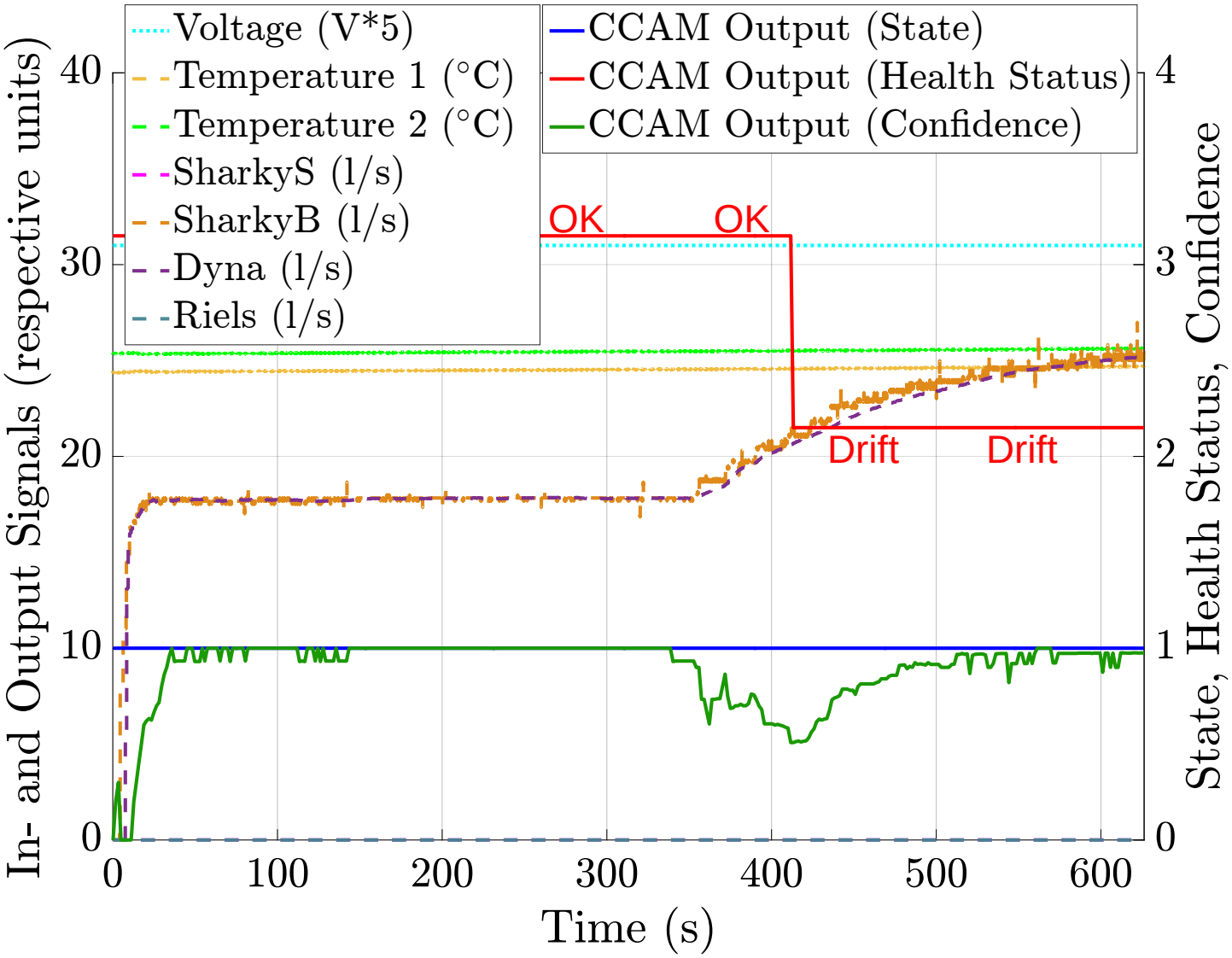


C²AM Case Study: Water Pipe System II



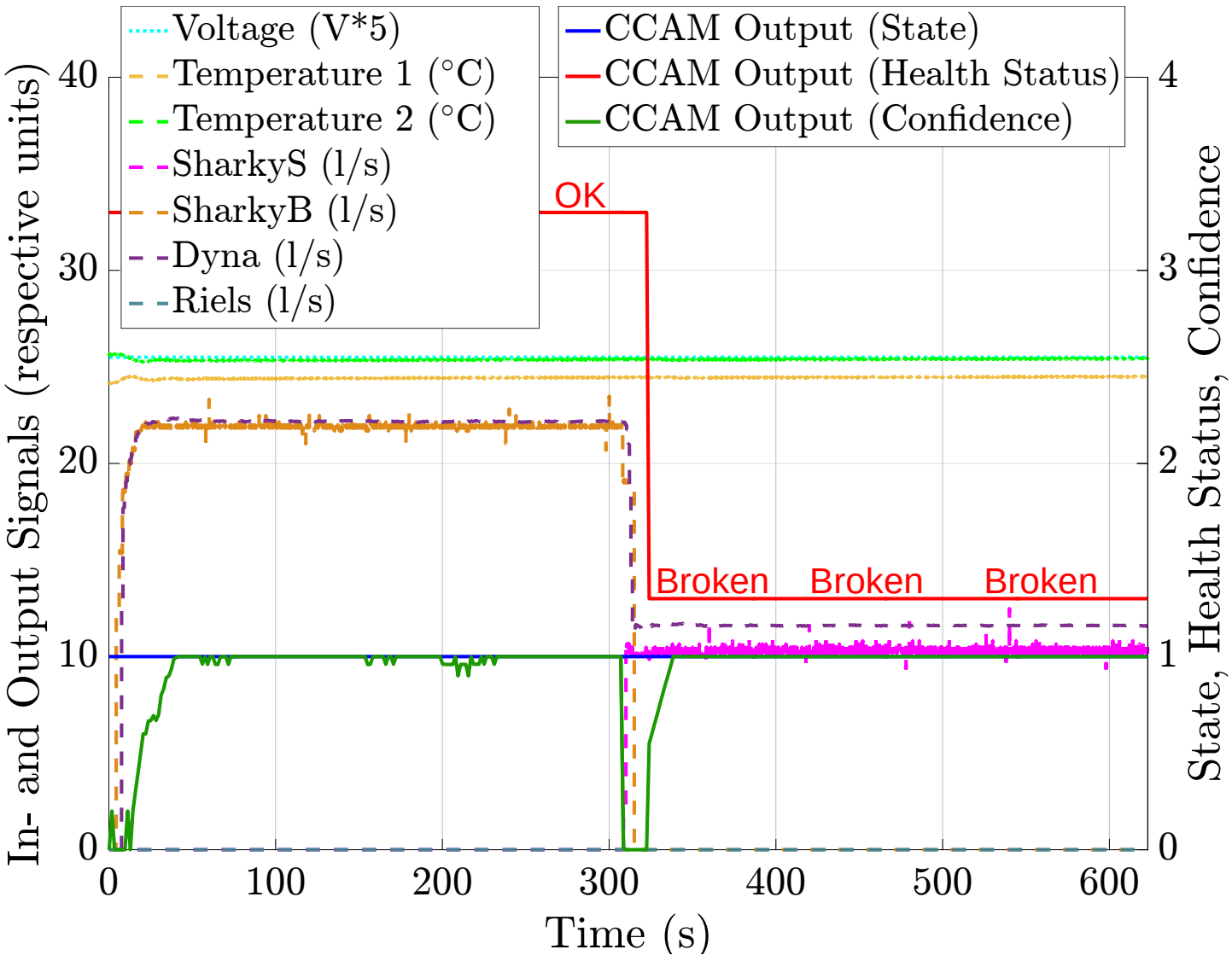
C²AM during state changes of the water pipe system.

C²AM Case Study: Water Pipe System III



Observation of a drift due to pipe leakage

C²AM Case Study: Water Pipe System IV

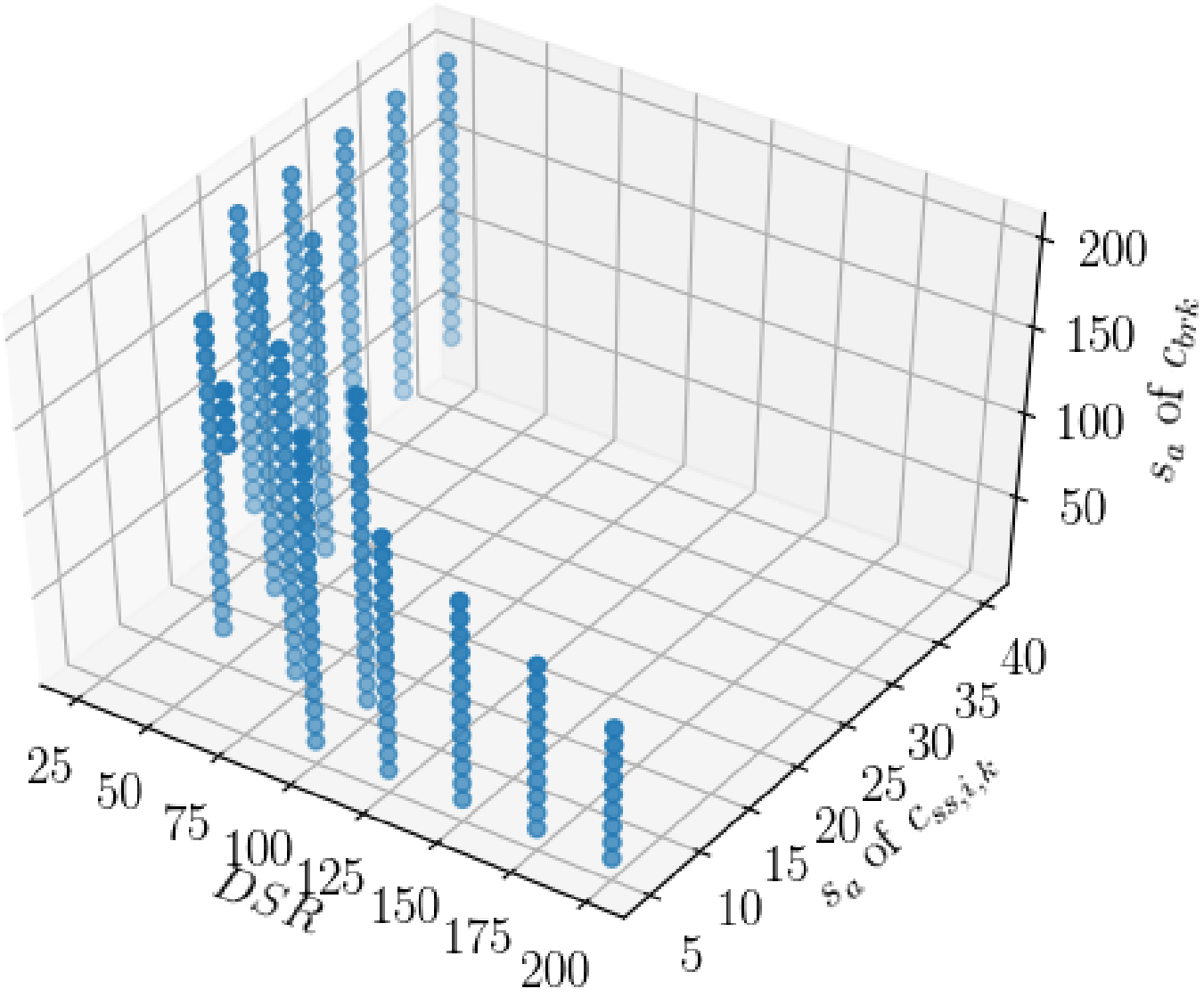


Broken system due to sudden appearance of a whole in a pipe.

C^2AM Parameter Sensitivity Analysis I

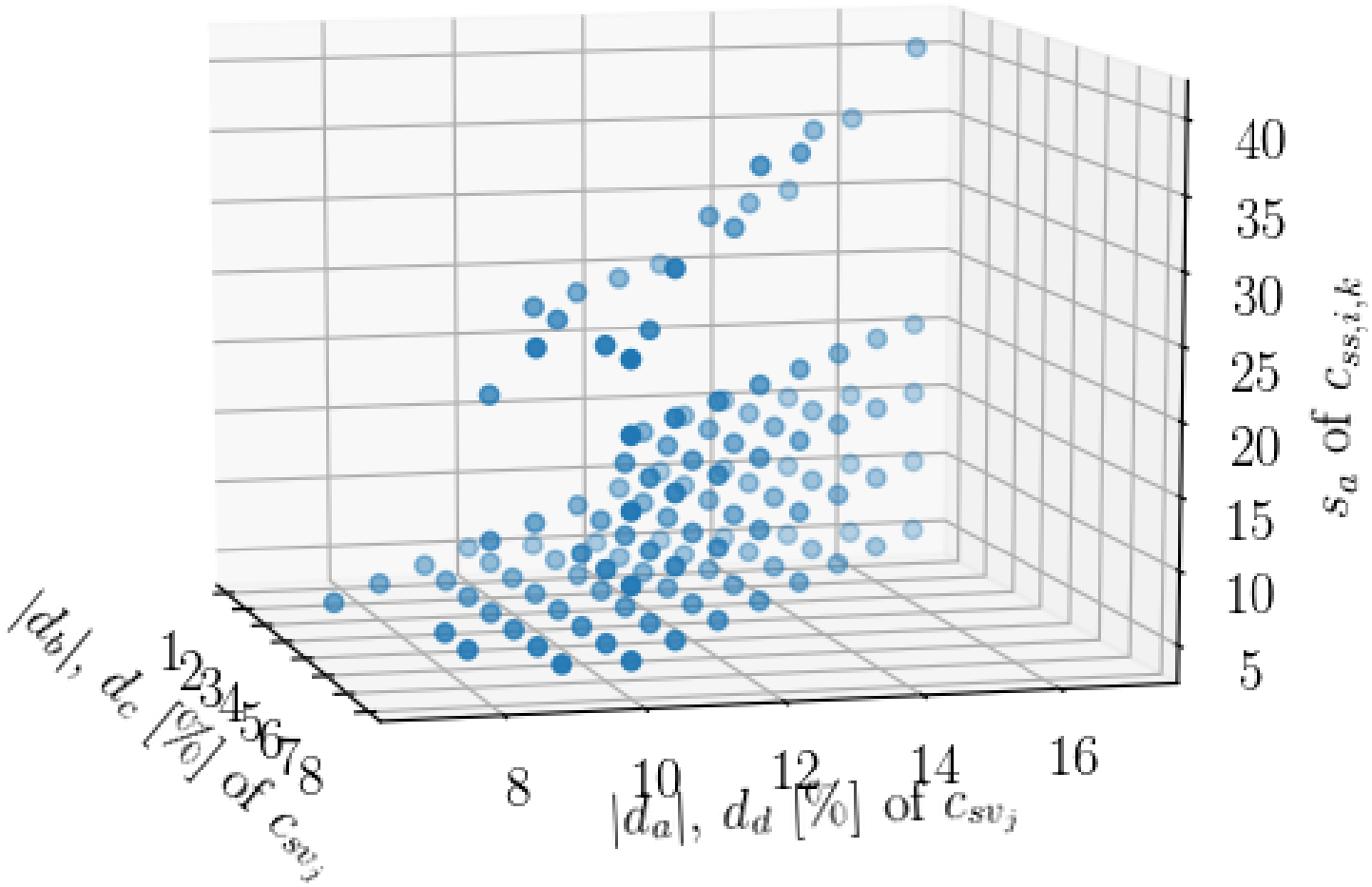
- C^2AM uses a number of empirically established parameters.
- They have to be adjusted for different applications.
- How sensitive is C^2AM to the settings of these parameters?

C²AM Parameter Sensitivity Analysis II



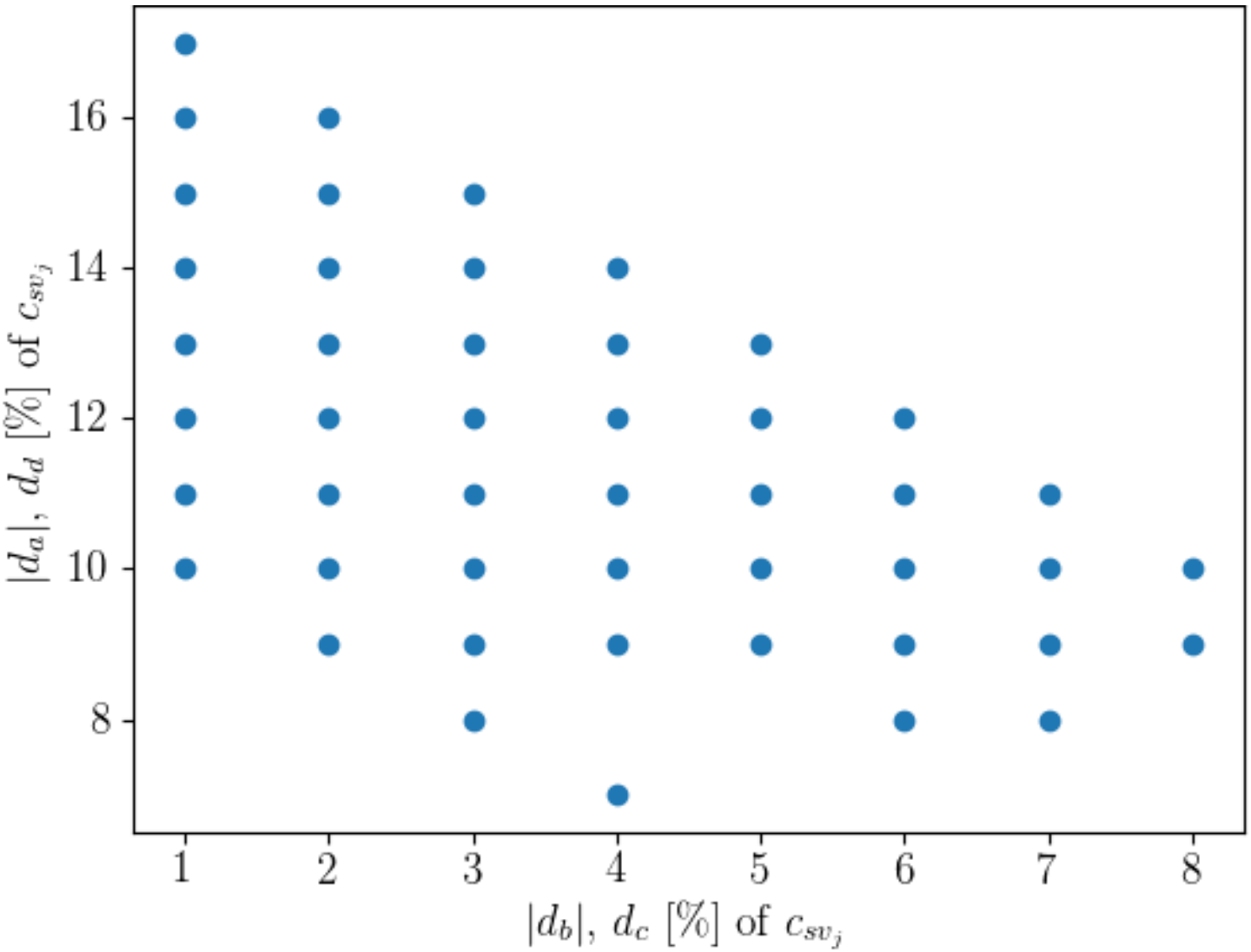
Effect of the Down Sampling Rate (DSR)

C²AM Parameter Sensitivity Analysis III



Mutual dependencies of fuzzy boundaries

C²AM Parameter Sensitivity Analysis IV



Effect of fuzzyfication: high d_b and d_c values mean steep membership functions.

Summary

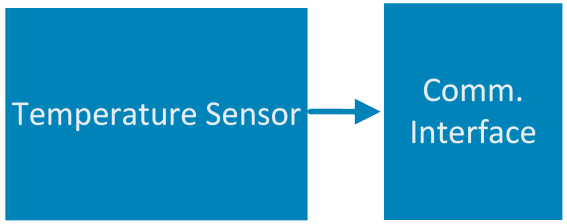
- Confidence has been used for self-assessment in various applications.
  - C^2AM for EWS, AC motor, water pipe systems
  - SVM based epileptic seizure monitoring
  - Multi-Classifier systems
  - Iterative CNNs
- Defined as distance
- Defined as probability

Nima TaheriNejad and Axel Jantsch. “Improved Machine Learning using Confidence”. In: *IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. Edmonton, Canada, May 2019

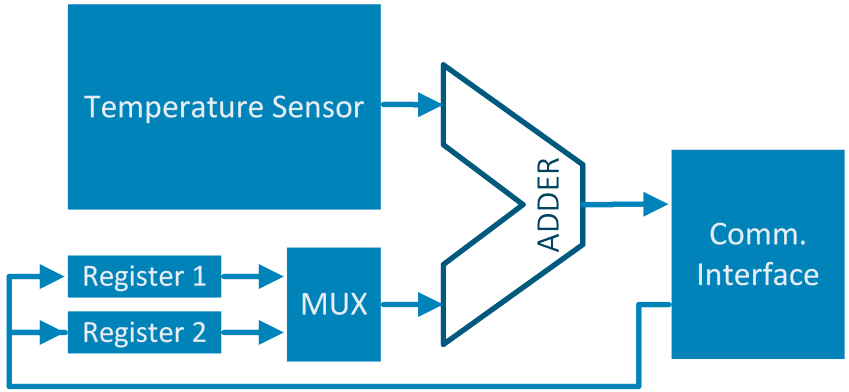
Situation Awareness and Attention

Attention Based Temperature Measurement

- How many temperature measurements are required in an MPSoC?
- It varies over several orders of magnitude depending on activity and current temperature.



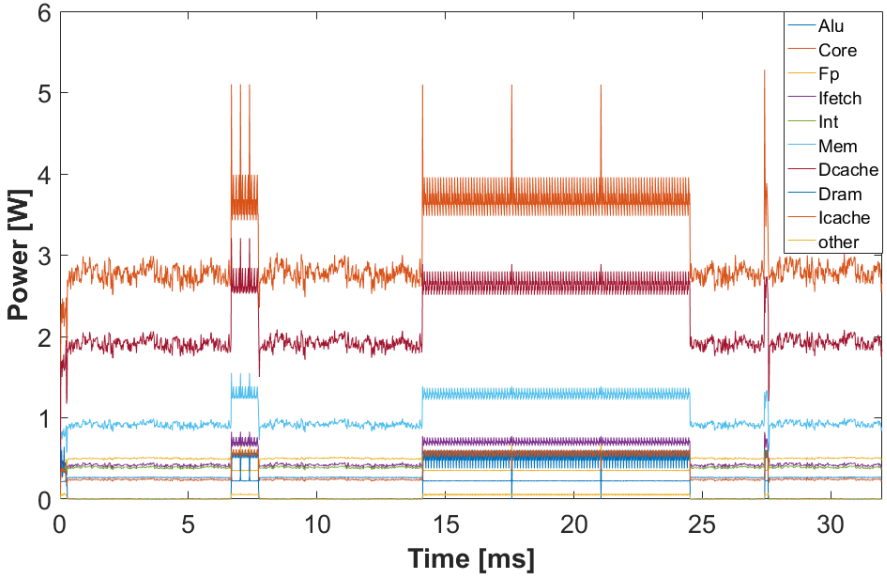
Conventional Architecture

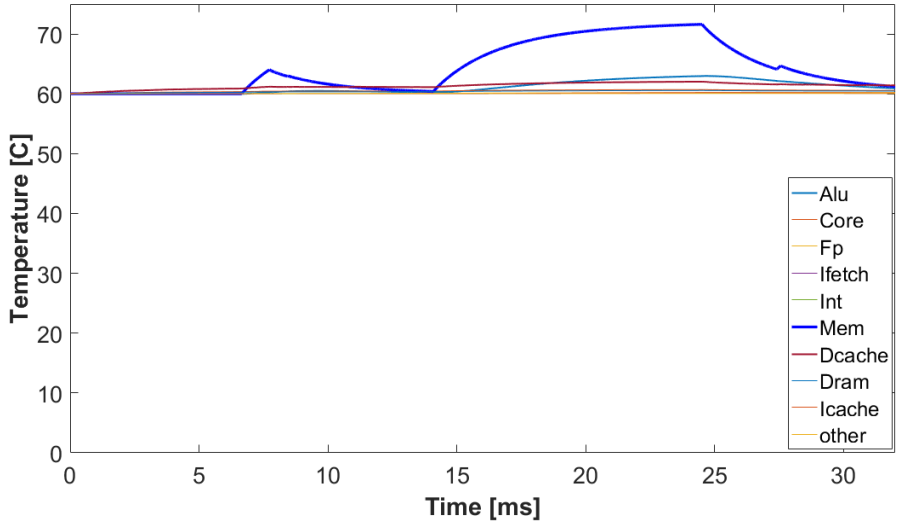


Proposed Architecture

Nima TaheriNejad, M. Ali Shami, and Sai Manoj P. D. "Self-aware sensing and attention-based data collection in Multi-Processor System-on-Chips". In: *15th IEEE International New Circuits and Systems Conference (NEWCAS)*. June 2017, pp. 81–84

Attention Based Temperature Measurement





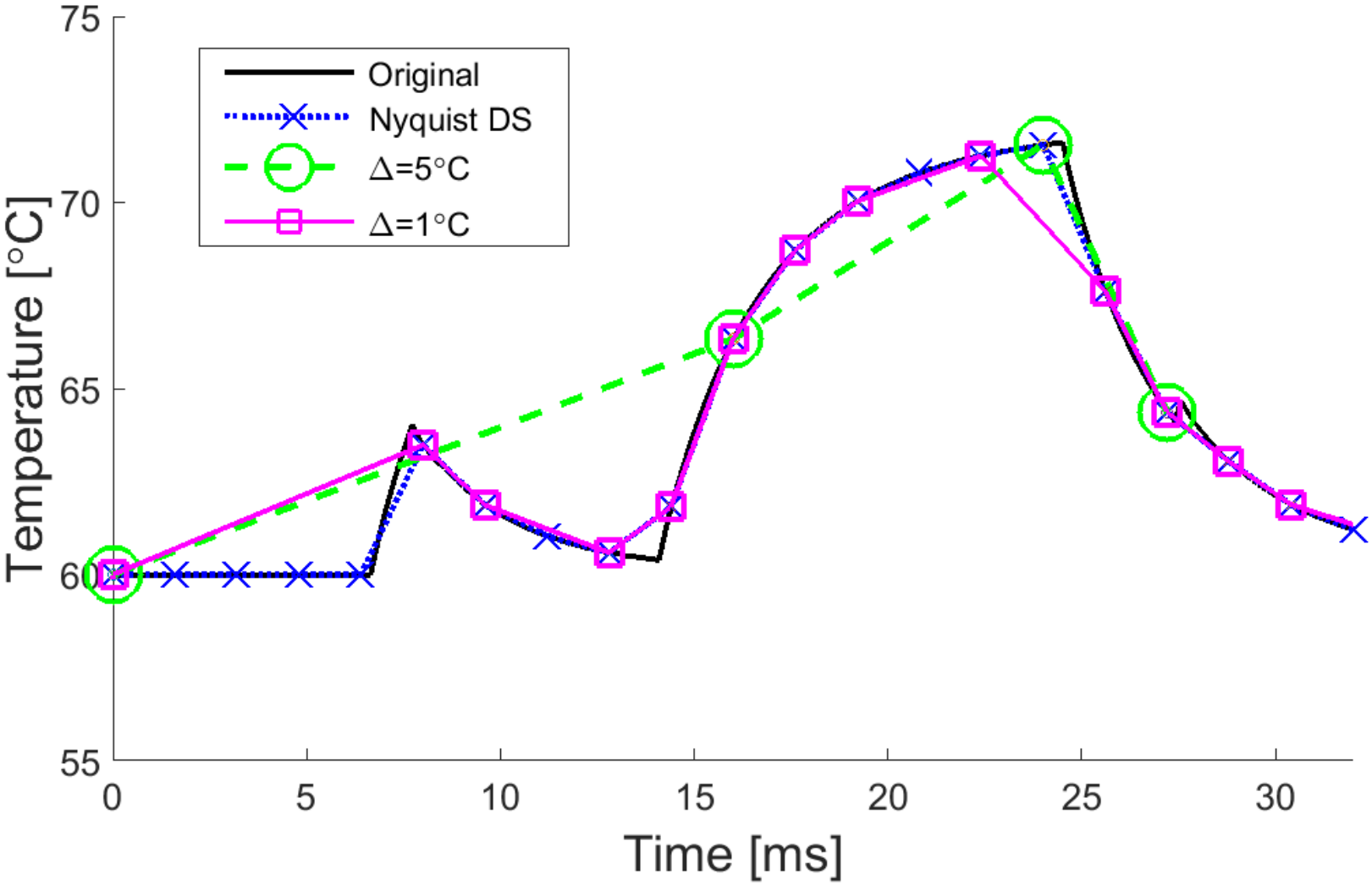
Intel Nehalem processor, running Barnes from SPLASH-2 Benchmarks, using Snipersim and Hotspot.

Attention Based Temperature Measurement

- When only differences $> \Delta = 1°C$ are reported, 7 out of 10 sensors send only 1 value in this experiment.
- Reduction of temperature reports for Memory, ALU and D-Cache:

Unit	$\Delta = 1$	Imp.	$\Delta = 2$	Imp.	$\Delta = 5$	Imp.
Memory	13	35%	9	55%	4	80%
ALU	4	80%	2	90%	1	95%
D-Cache	2	90%	2	90%	1	95%
All others	1	95%	1	95%	1	95%

Attention Based Temperature Measurement

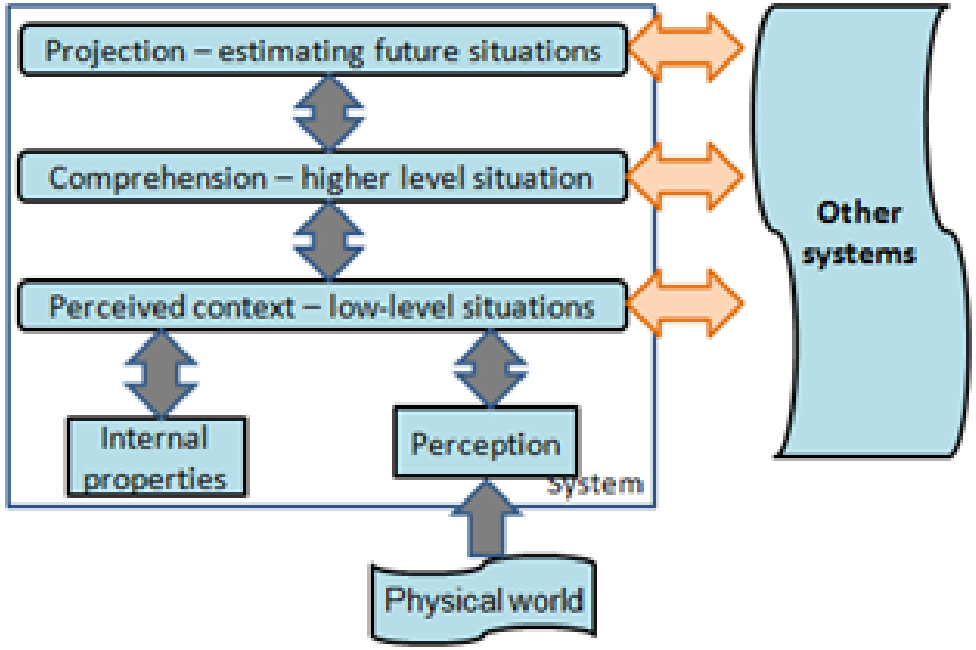


Attention Based Temperature Measurement

- Rate of temperature reporting can be significantly reduced and fine tuned;
- Can depend on
  - relative difference,
  - absolute difference,
  - absolute value,
  - system level mode;
- Potential benefits:
  - reduced processing,
  - reduced communication,
  - reduced measurements.

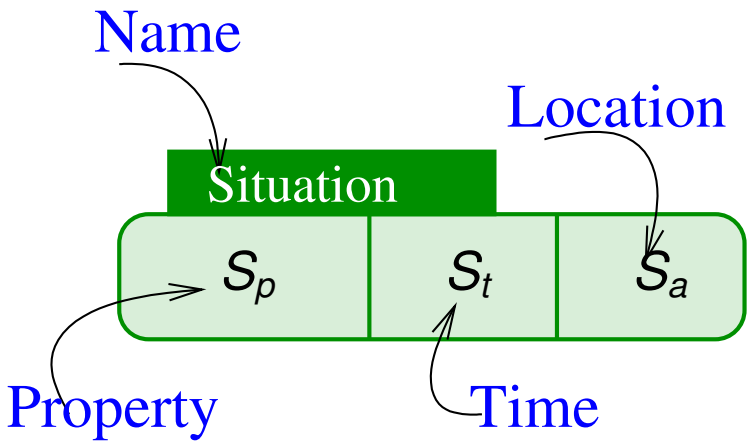
Situation Aware Health Monitoring I

- CPSs live in complex, varying environments.
- Situation assessment is critical for both efficient and effective decision making.
- Multiple sensors provide parts of the necessary information.



J.-S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis. "The benefits of self-awareness and attention in fog and mist computing". In: *IEEE Computer, Special Issue on Self-Aware/Expressive Computing Systems* (July 2015), pp. 37–45

Situation Aware Health Monitoring II



- Situation is modeled as a 3-tuple.
- S_t denotes the temporal information.
- S_a denotes the spatial information.
- S_p is a set of properties.
- Each property can be atomic with a value, or compound.

Situation Aware Health Monitoring III

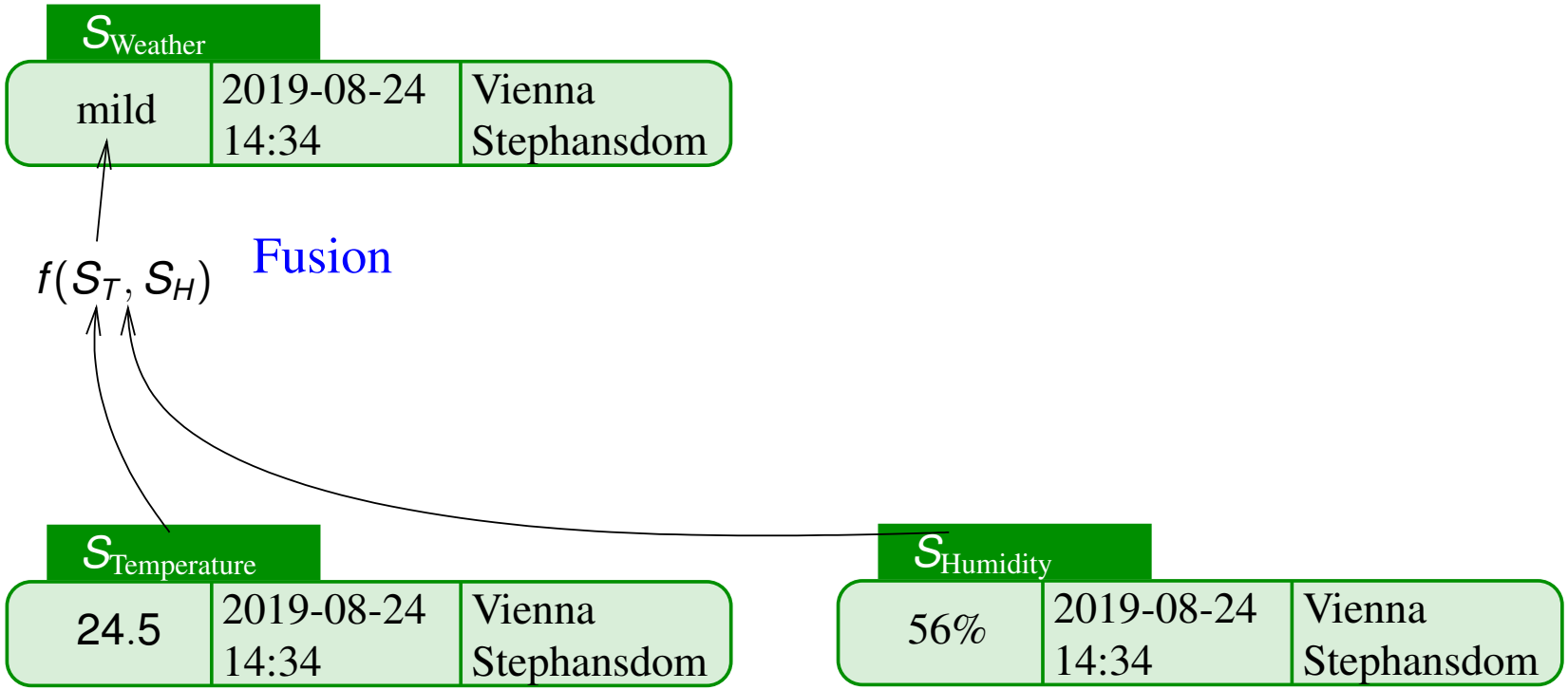
$S_{\text{Temperature}}$

24.5	2019-08-24 14:34	Vienna Stephansdom

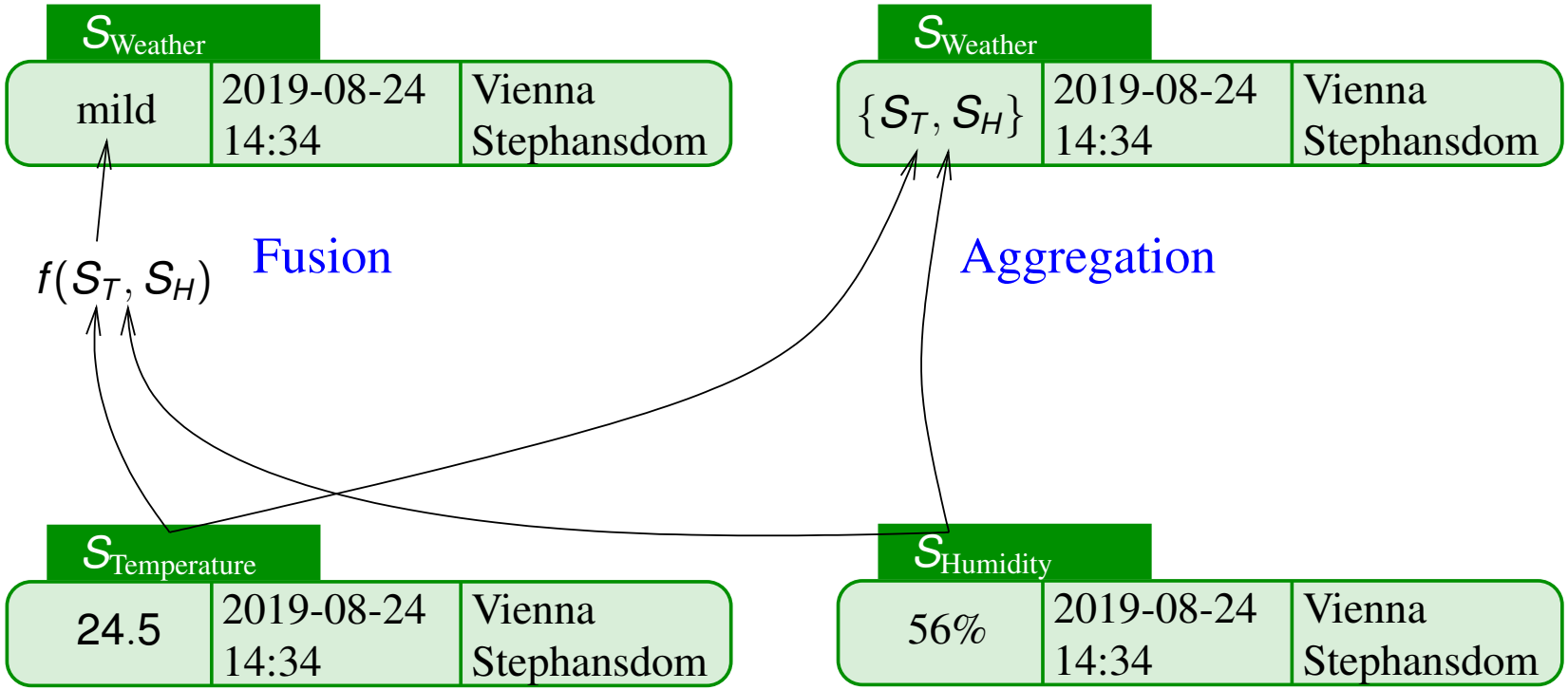
S_{Humidity}

56%	2019-08-24 14:34	Vienna Stephansdom

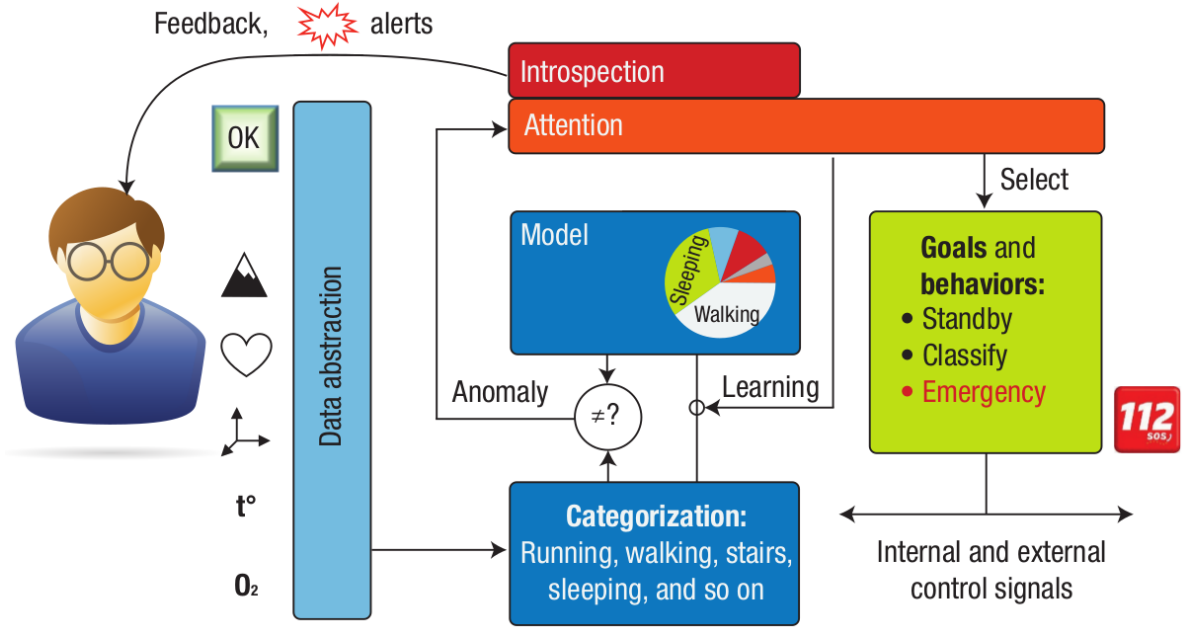
Situation Aware Health Monitoring IV



Situation Aware Health Monitoring V

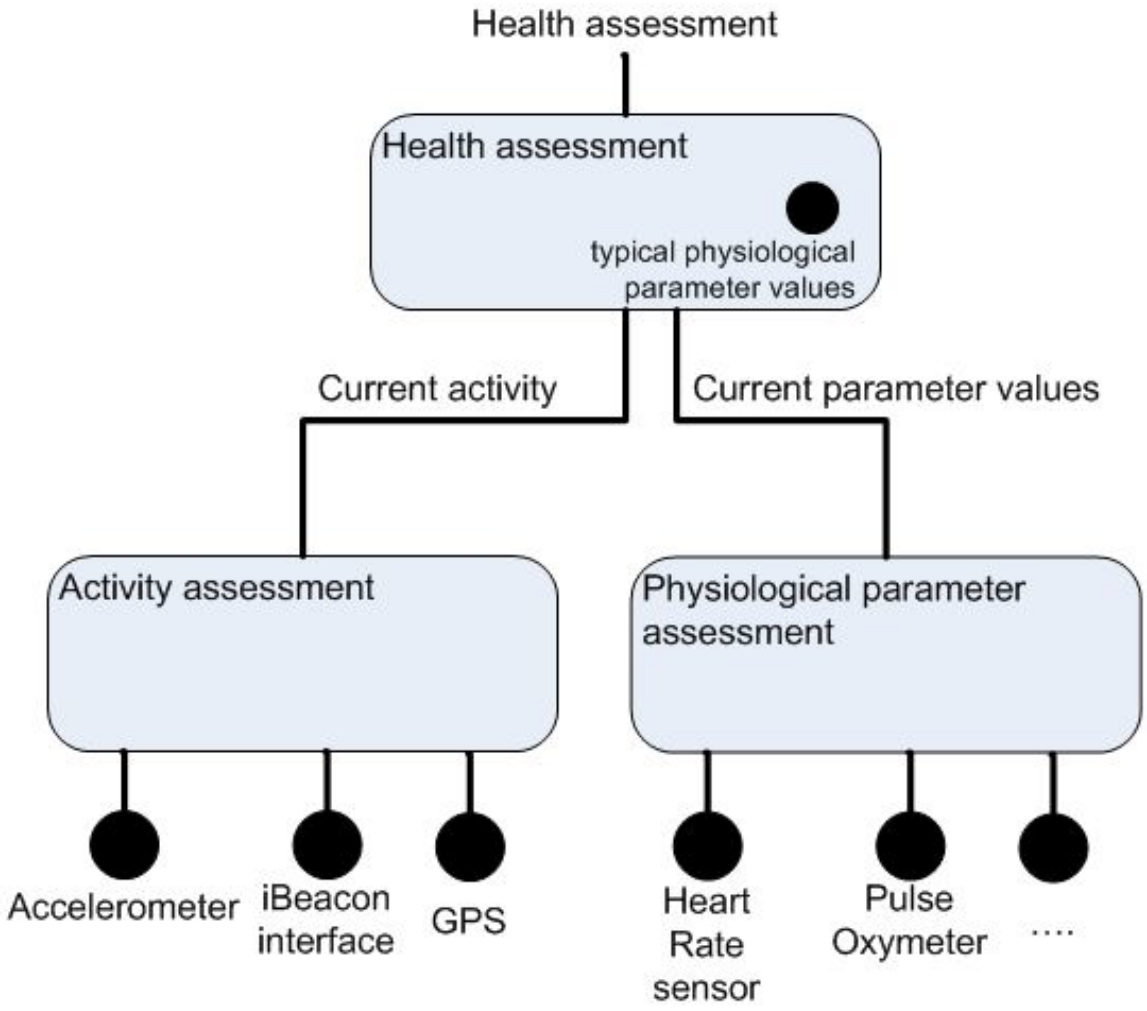


Situation Aware Health Monitoring VI

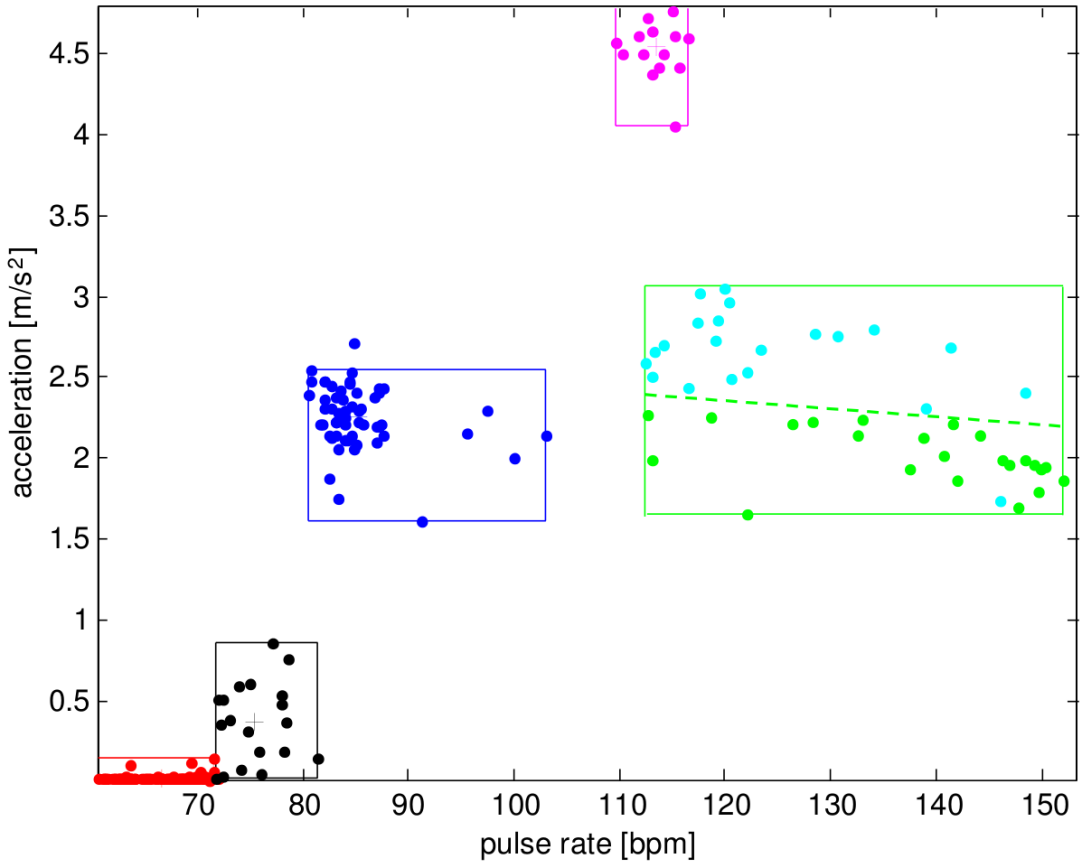


- Mobile health monitoring with situation dependent assessment.
- Sensors: accelerometer, GPS, heart rate monitor, a pulse oximeter, altitude.
- Activities: resting on a couch, working at a table, walking slowly indoors, climbing stairs indoors, walking slowly outdoors and walking at a rapid pace outdoors.

Situation Aware Health Monitoring VII



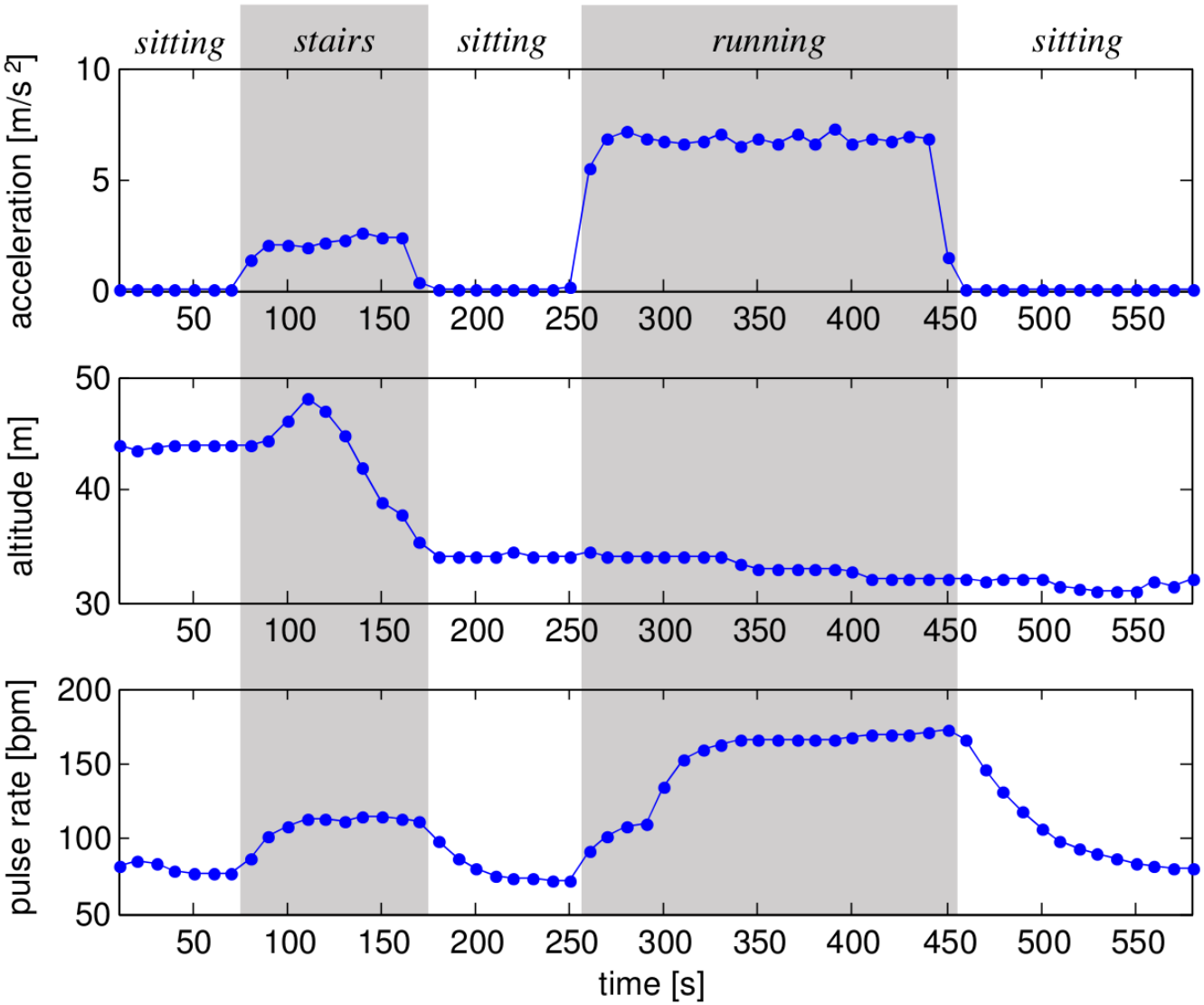
Situation Aware Health Monitoring VIII



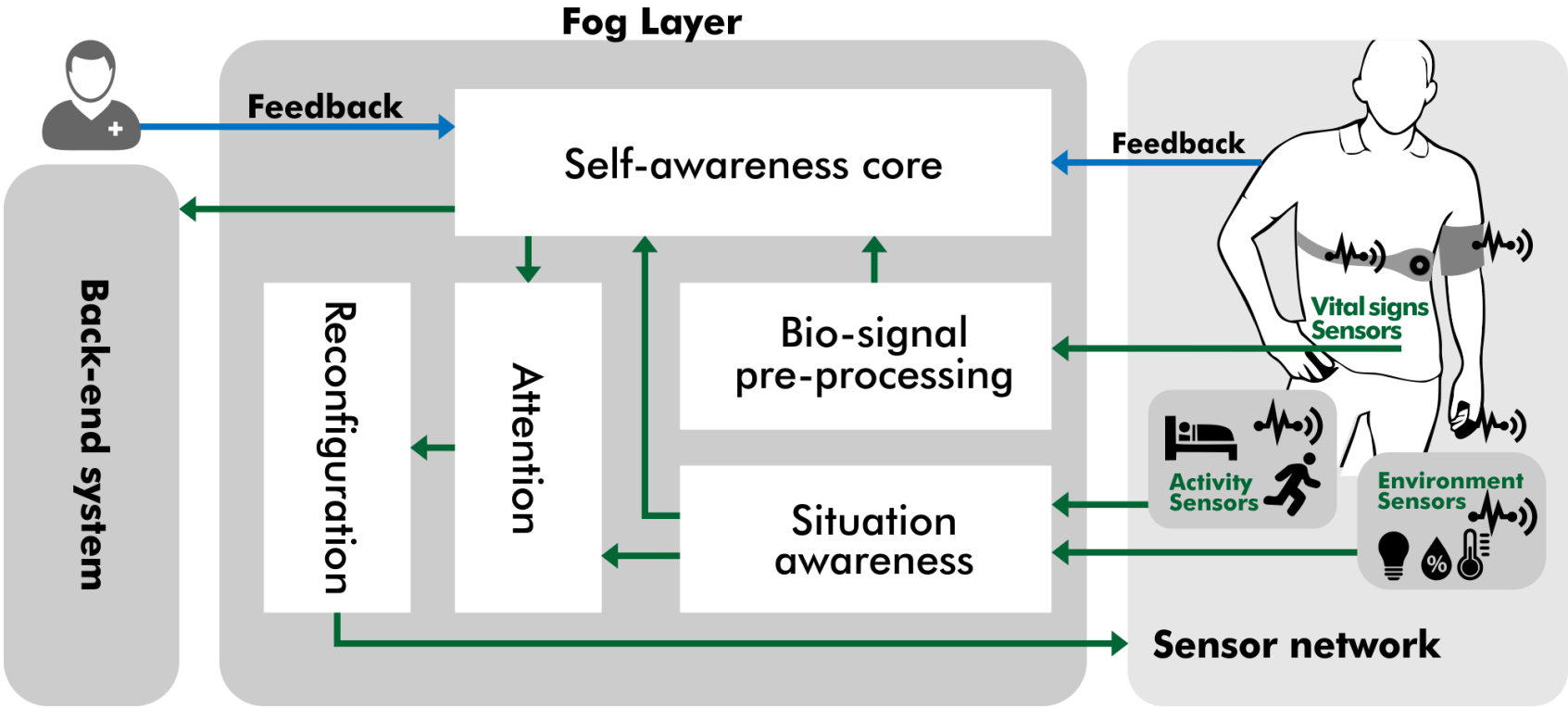
- sitting or lying
- car driving
- indoor slow walking
- outdoor rapid walking
- climbing stairs up and down

- Accelerometer and pulse rate are sufficient for most activities.
- Altitude measurements were required to separate outdoor walking from stair climbing.

Situation Aware Health Monitoring IX



Enhanced Early Warning Score I



Arman Anzanpour et al. “Self-Awareness in Remote Health Monitoring Systems using Wearable Electronics”. In: *Proceedings of Design and Test Europe Conference (DATE)*. Lausanne, Switzerland, Mar. 2017

Enhanced Early Warning Score II

1. Prioritize different situations

Enhanced Early Warning Score III

1. Prioritize different situations
2. Distinguish different modes of urgency

Emergency Level:	Score:0 Normal				Score:1-3 Low				Score:4-6 Medium				Score>6 High			
	Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor	
	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night
Sleeping	E	E	E	E	C	D	D	D	B	C	C	C	A	A	B	B
Resting	D	D	D	D	C	C	C	C	B	B	B	B	A	A	B	B
Walking	C	C	C	C	B	C	C	C	B	B	B	B	A	A	A	B
Jogging	C	C	C	C	B	B	B	C	B	B	B	B	A	A	A	B
Running	C	C	C	C	B	B	B	B	B	B	B	B	A	A	A	A

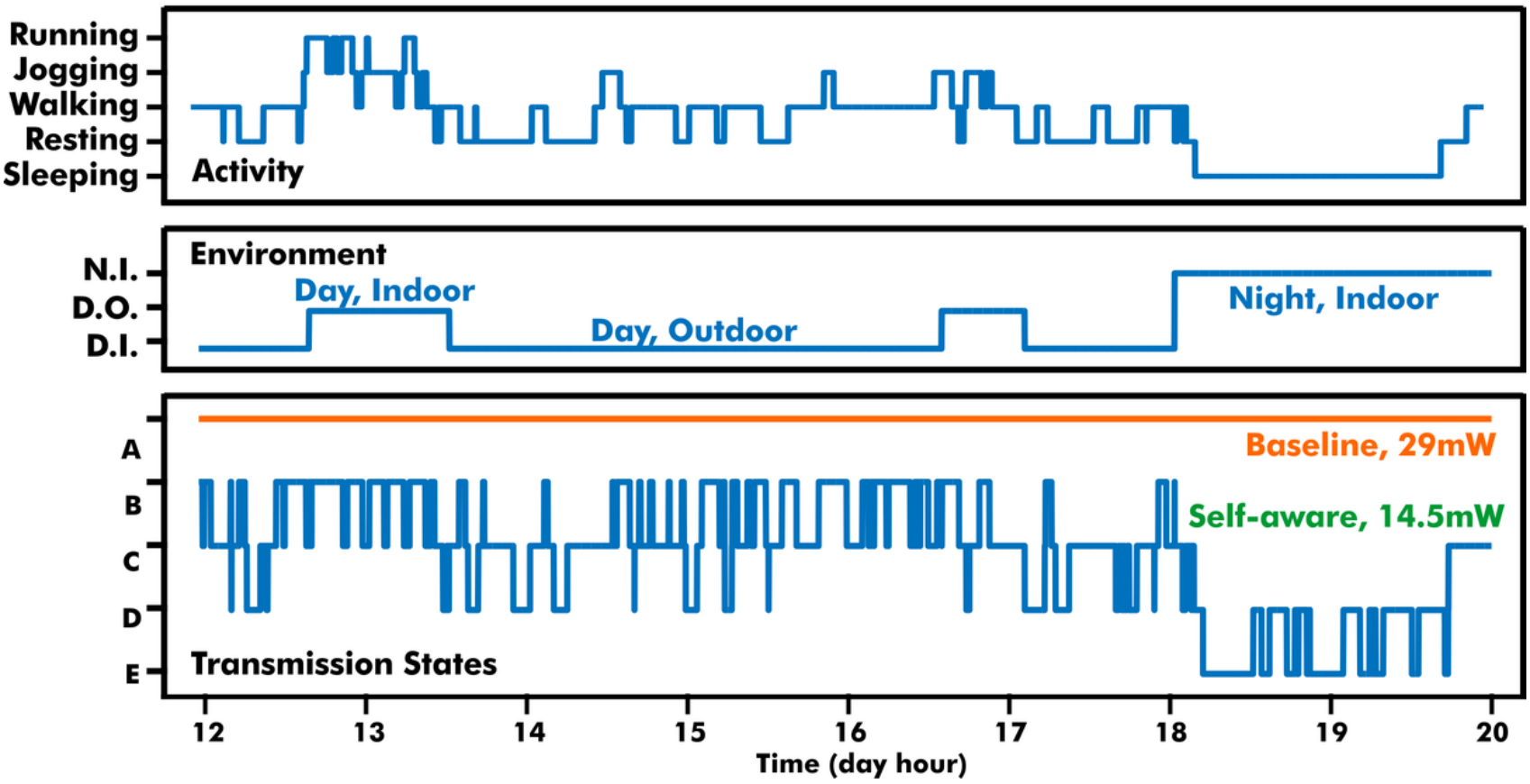
Enhanced Early Warning Score IV

1. Prioritize different situations
2. Distinguish different modes of urgency
3. Define sensing activity for each mode

State	Respiration Rate Activity	Blood Pressure	Heart Rate, SpO2, and Body Temp.	Transmission Power Consumption
A	Continuous	Every hour at day Disabled at night	Every sec.	29 mW
B	2 min continuous 8 min OFF	Every hour at day Disabled at night	Every sec.	26.8 mW
C	2 min continuous 3 min OFF	Every 3 hours at day Disabled at night	Every min.	12.5 mW
D	2 min continuous 8 min OFF	Every 3 hours at day Disabled at night	Every min.	7 mW
E	2 min continuous 18 min OFF	Disabled	Every min.	4.3 mW

Enhanced Early Warning Score V

Over a day half the energy can be saved.



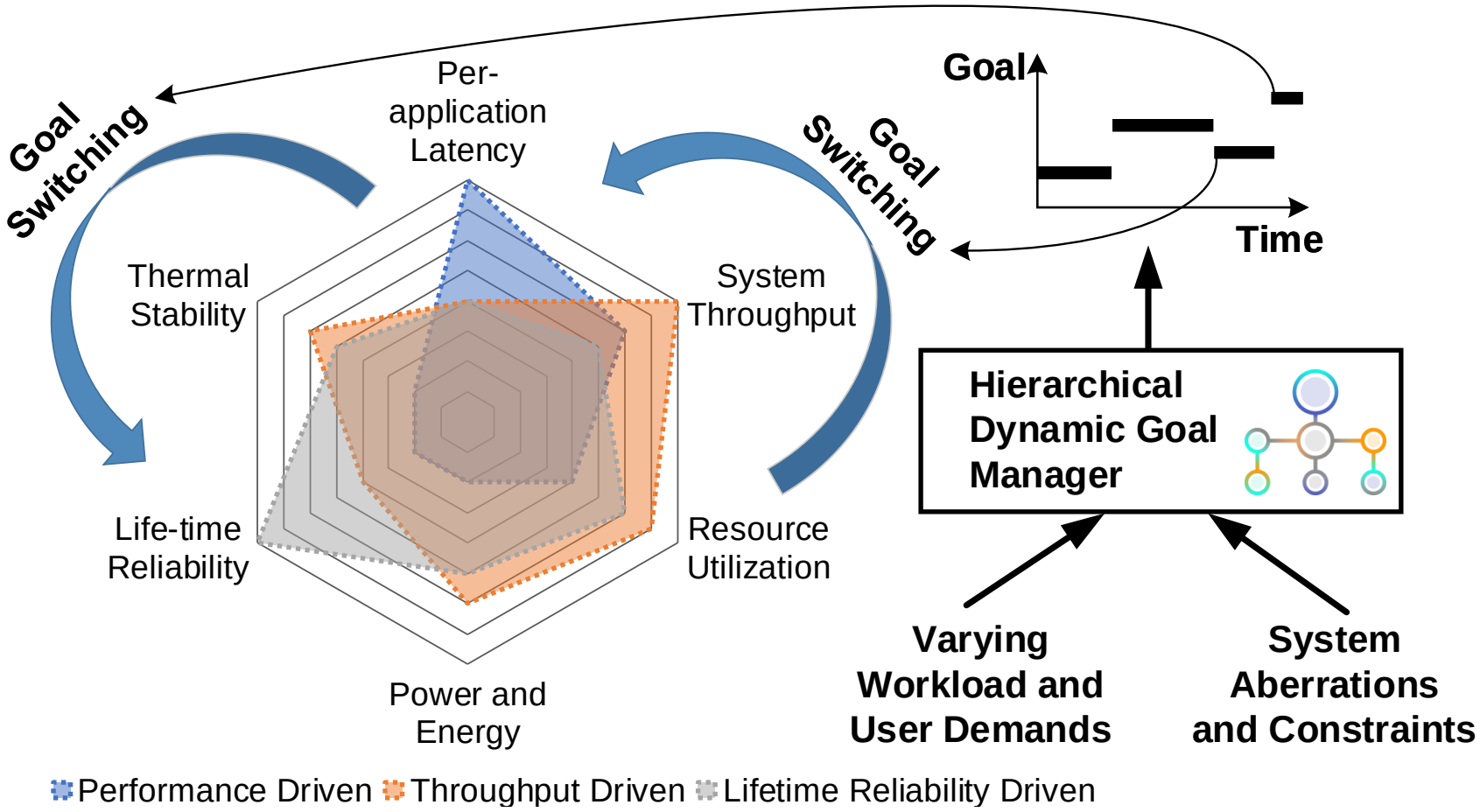
Attention and Situation Awareness

Summary

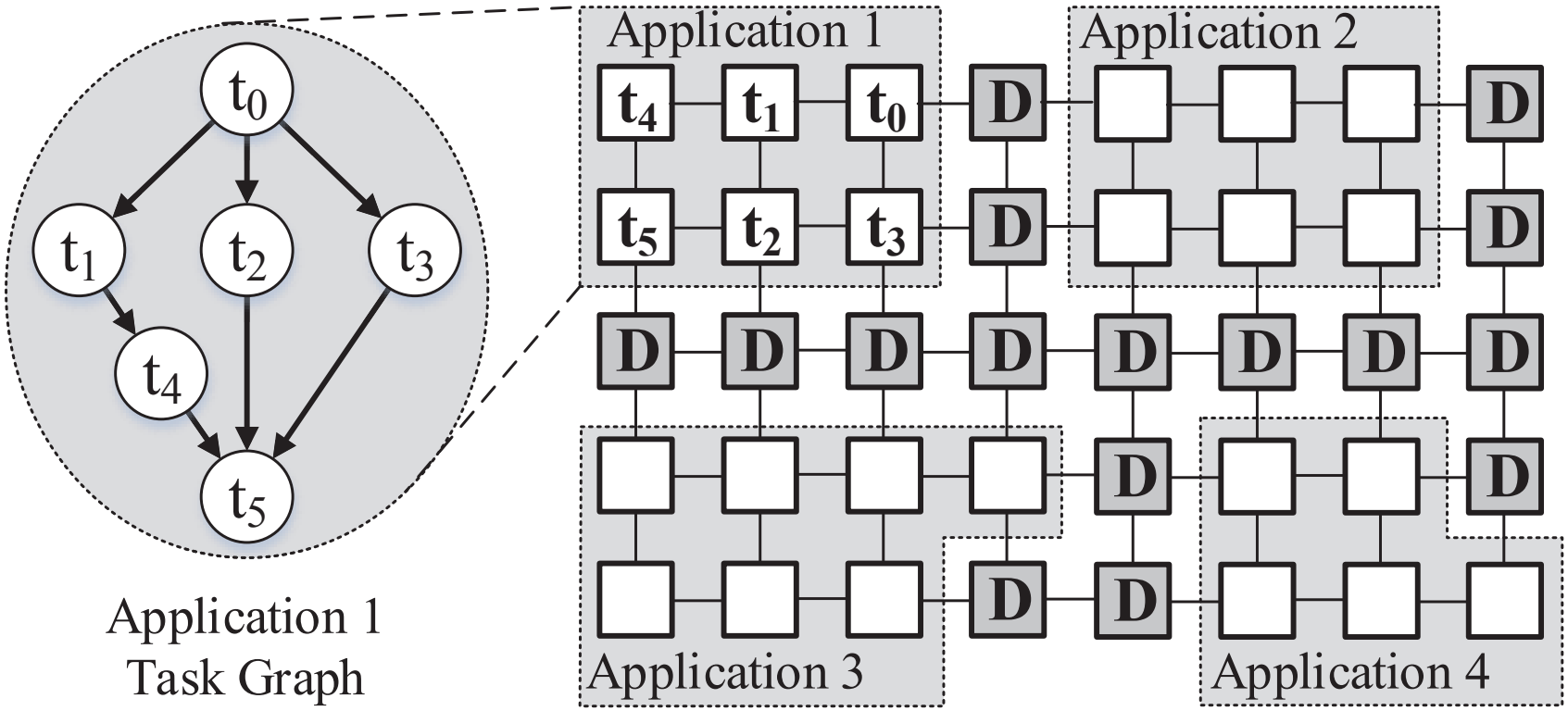
- Situation awareness is inferred from sensory data.
- Various sensors contribute.
- Sensor data may be redundant, faulty, irrelevant.
- Attention allows to focus on what is relevant and urgent.
- Situation awareness and Attention facilitates
  - good decision making.
  - efficient usage of resources.

Goal Management

Goals for Dynamic Task Mapping



Dynamic Task Mapping



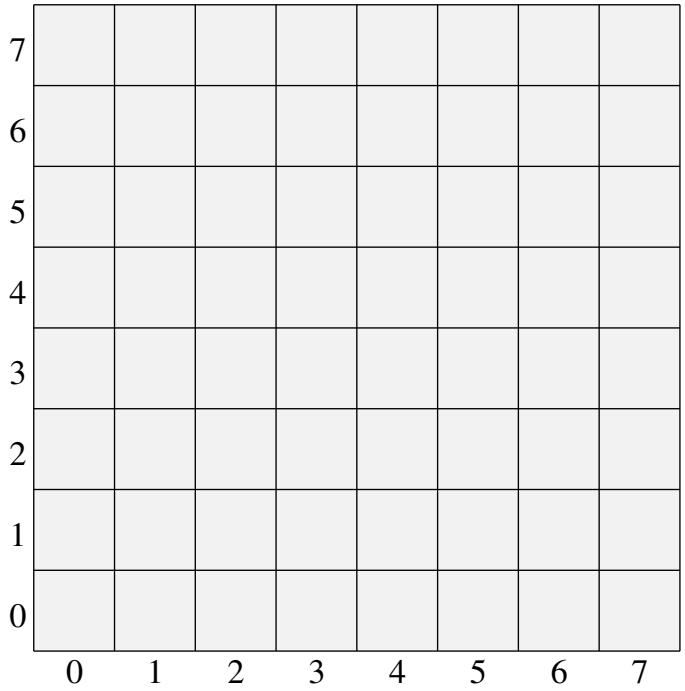
Example 1: Performance Driven Task Mapping

MapPro Objectives:

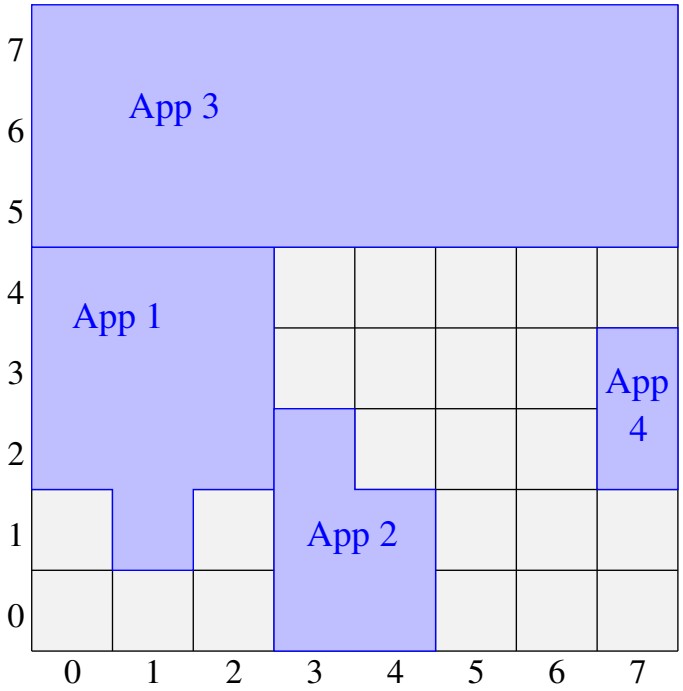
- Maximize performance for all applications;
- Minimize communication latency in the new application;
- Minimize fragmentation.

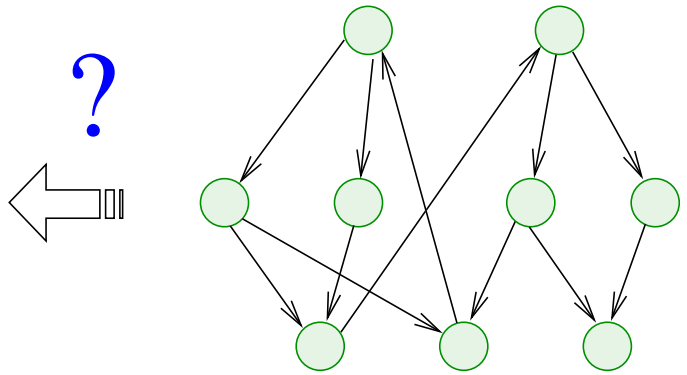
Mohammad-Hashem Haghbayan, Anil Kanduri, Amir-Mohammad Rahmani, Pasi Liljeberg, Axel Jantsch, and Hannu Tenhunen. "MapPro: Proactive Runtime Mapping for Dynamic Workloads by Quantifying Ripple Effect of Applications on Networks-on-Chip". In: *Proceedings of the International Symposium on Networks on Chip*. Vancouver, Canada, Sept. 2015

Example 1: Performance Driven Task Mapping

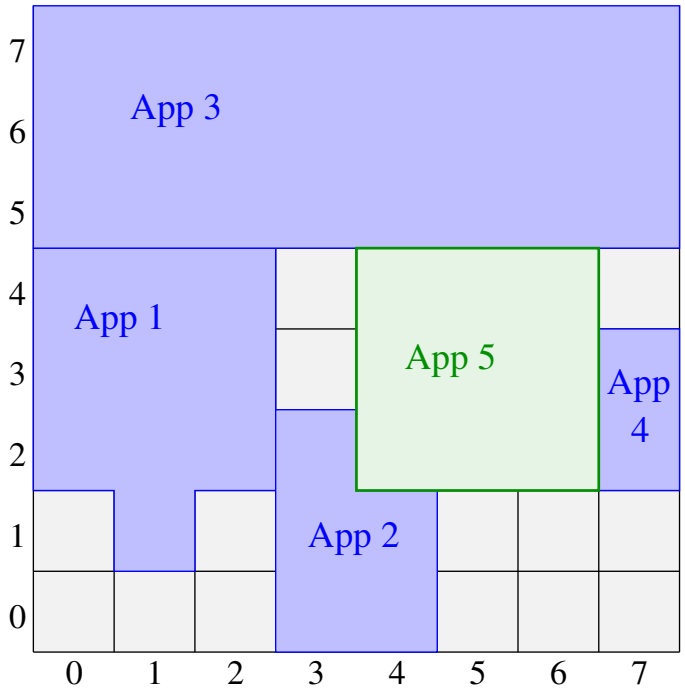


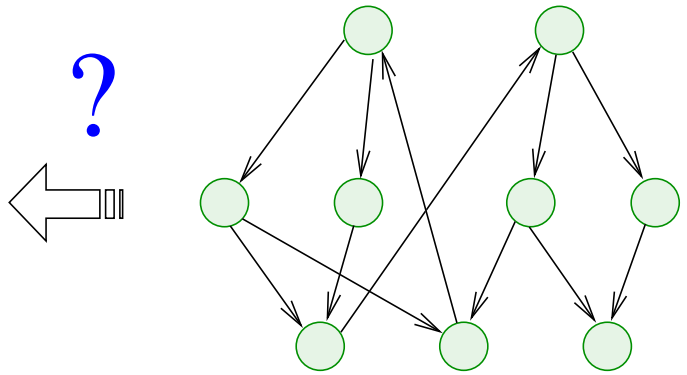
Example 1: Performance Driven Task Mapping



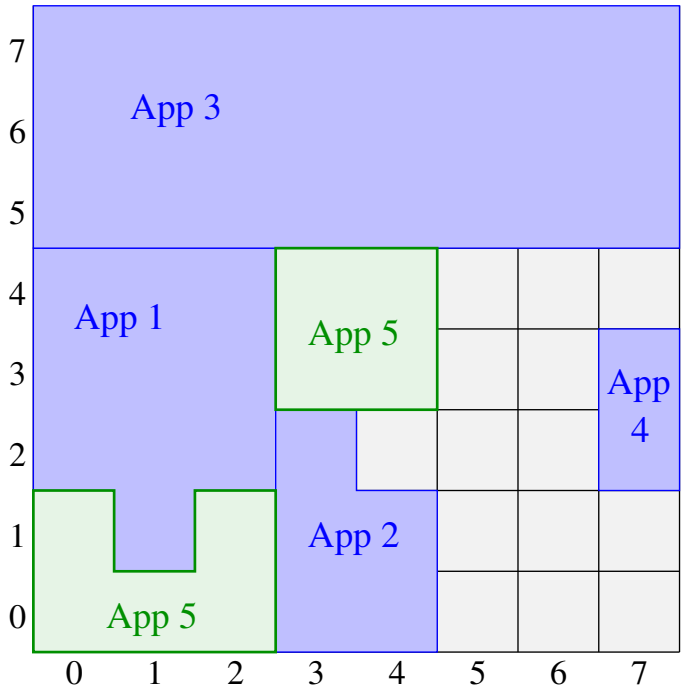


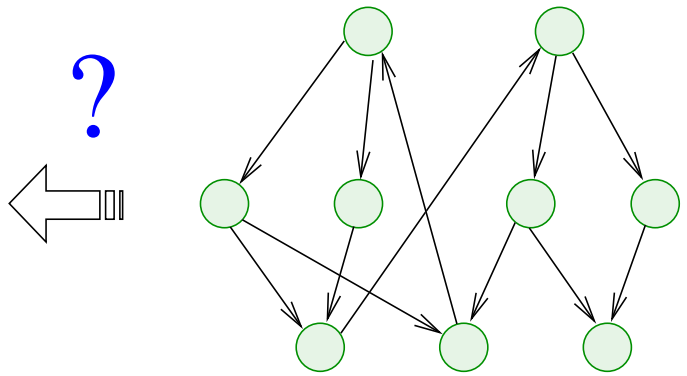
Example 1: Performance Driven Task Mapping



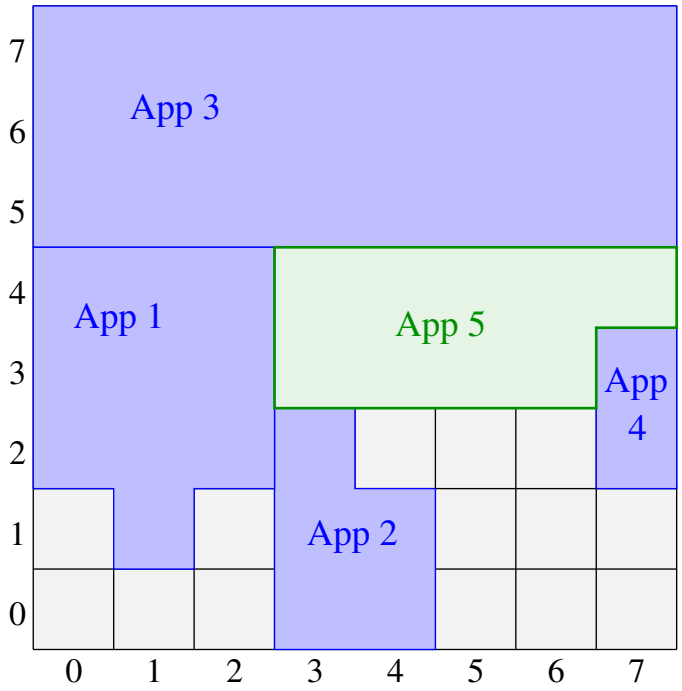


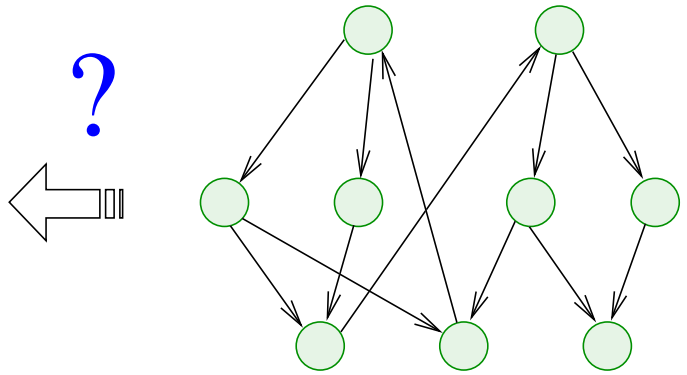
Example 1: Performance Driven Task Mapping



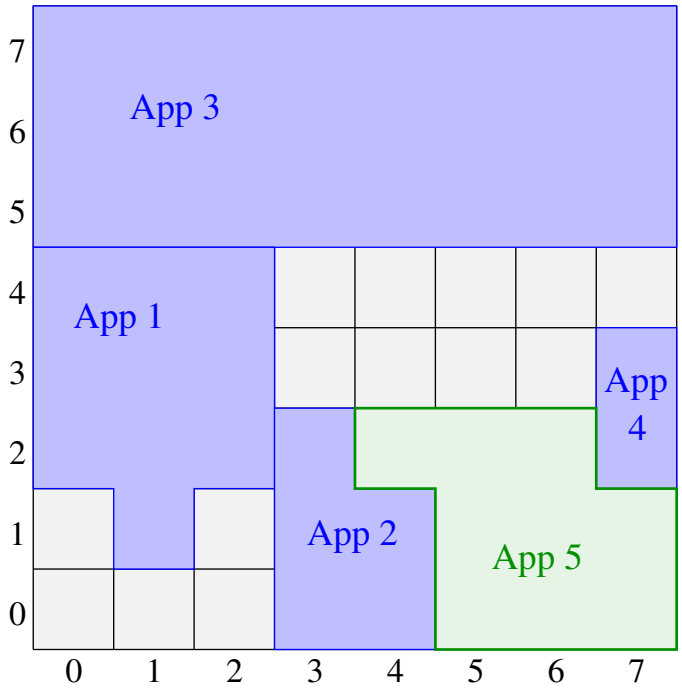


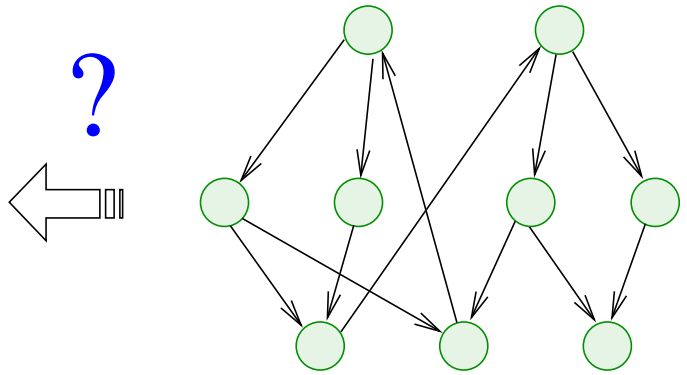
Example 1: Performance Driven Task Mapping



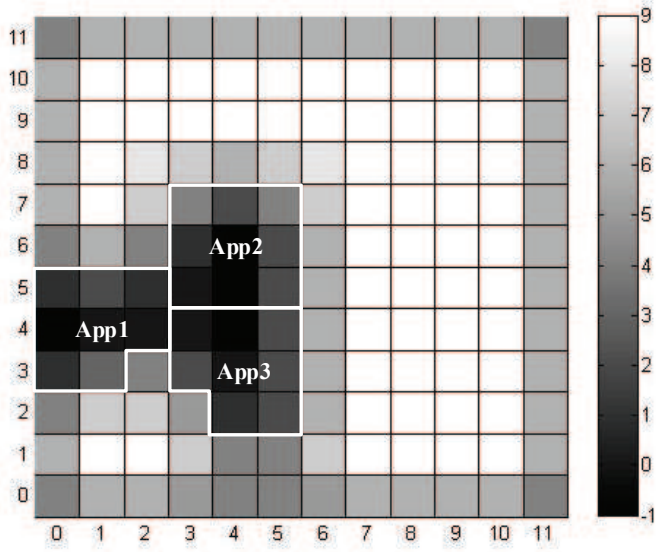


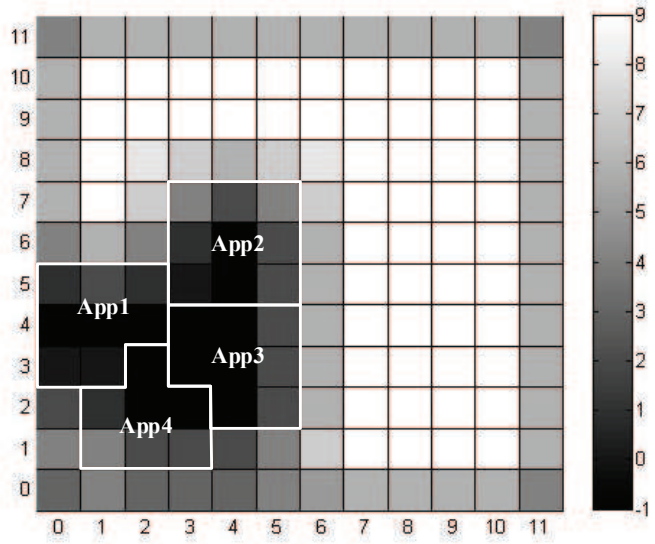
Example 1: Performance Driven Task Mapping





Example 1: Performance Driven Task Mapping

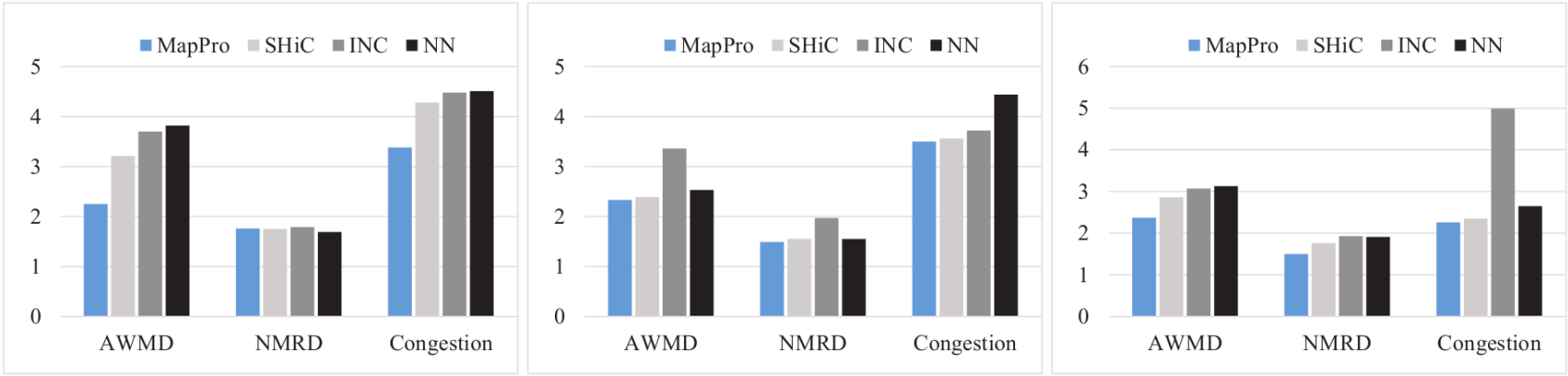




MapPro: Heuristic to minimize application internal communication delay and to minimize fragmentation.

1. First Node selection: Identifies a first node and a region for a new application;
2. Allocates specific cores around the first node;
3. Maps tasks to cores.

Example 1: Performance Driven Task Mapping



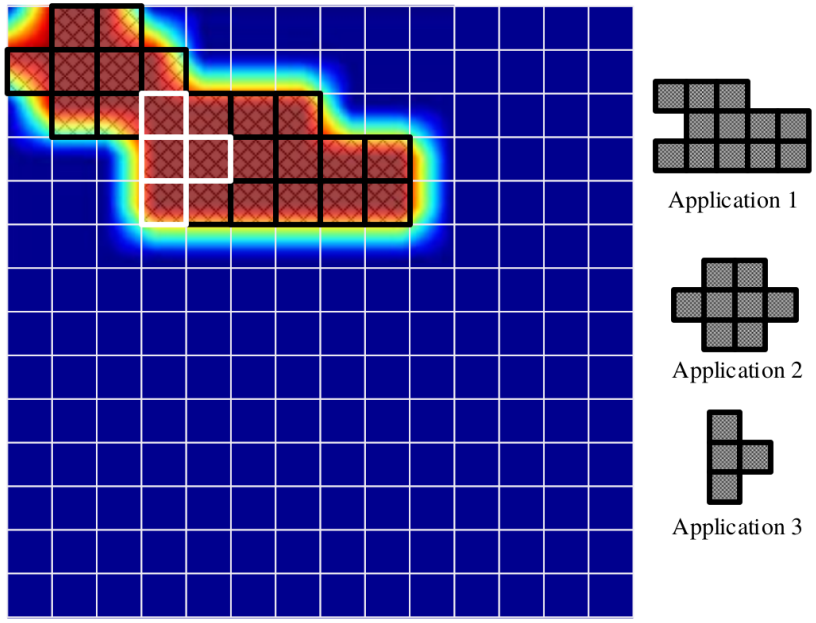
Experiments with 12x12 - 16x16 networks.

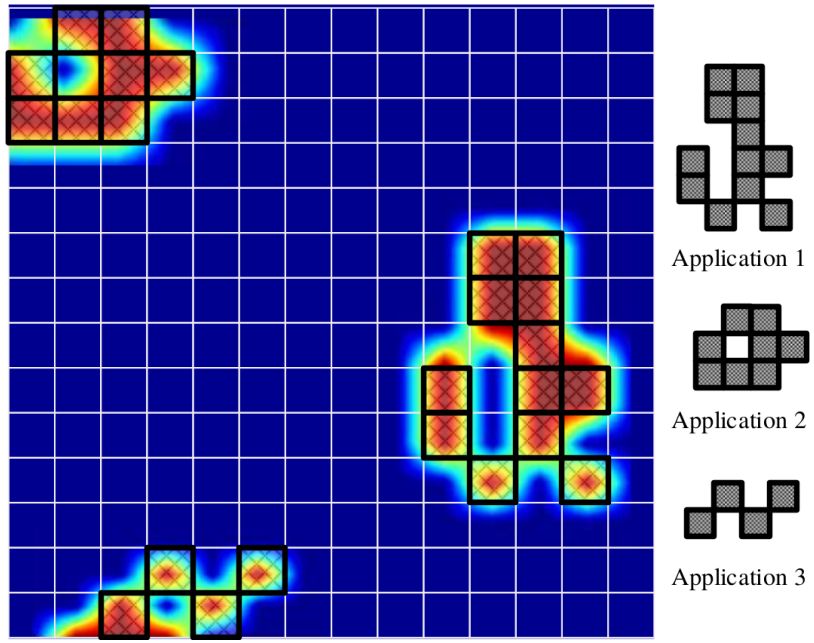
AWMD: Average Weighed Manhattan Distance: Measures the communication cost based on traffic volume.

NMRD: Normalized Mapped Region Dispersion is the normalized average of pairwise Manhattan distances of all communication nodes of a mapped application: measures the compactness of a region.

External Congestion: Number of contended packets belonging to different applications.

Example 2: Power- and Thermal Constrained Task Mapping

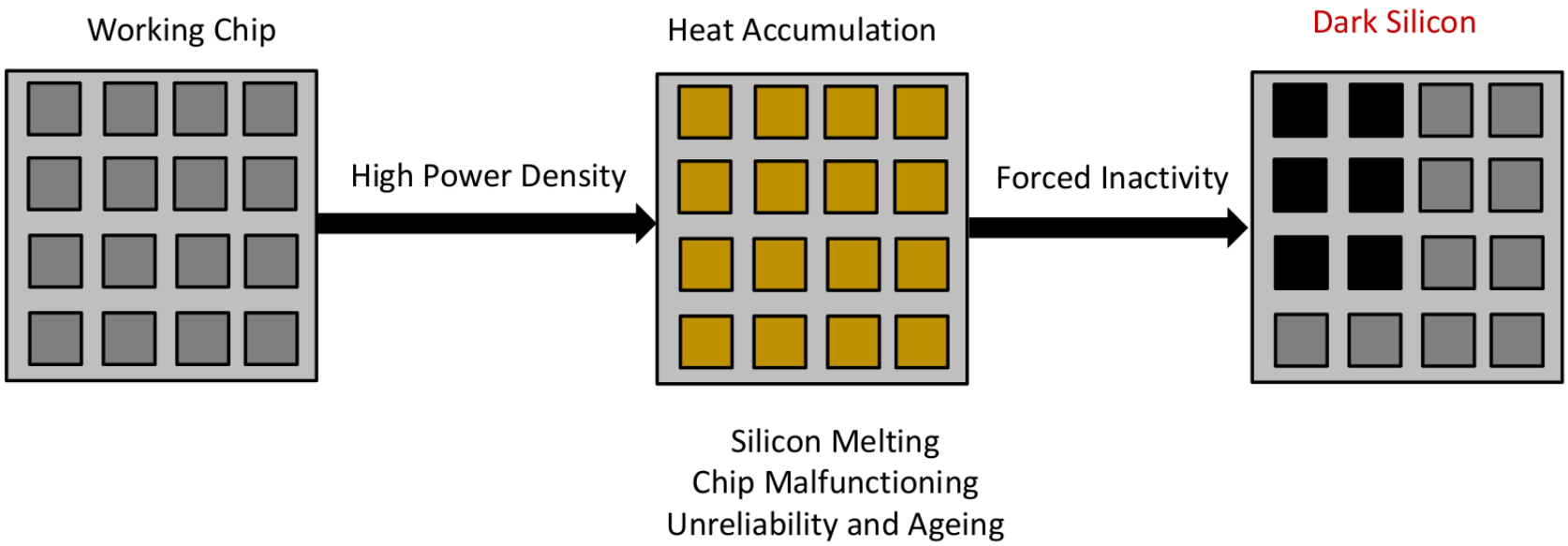




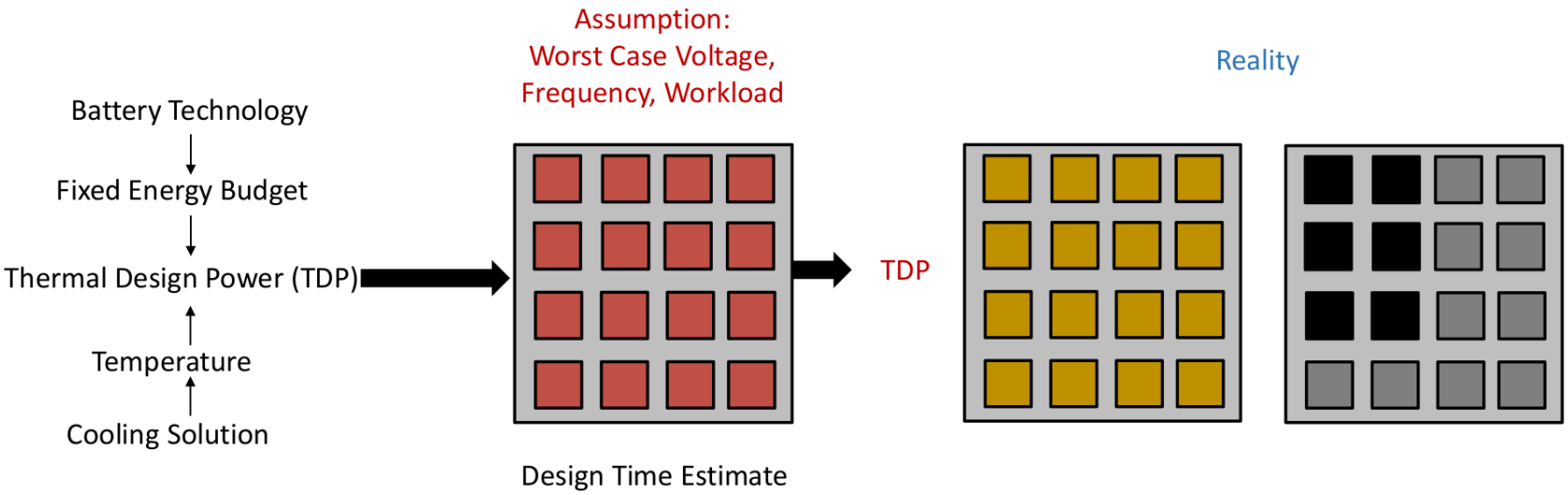
The patterning algorithm disperses mapped cores to maximize the Thermal Safe Power budget.

Anil Kanduri, Mohammad-Hashem Haghbayan, Amir-Mohammad Rahmani, Pasi Liljeberg, Axel Jantsch, and Hannu Tenhunen. “Dark Silicon Aware Runtime Mapping for Many-core Systems: A Patterning Approach”. In: *Proceedings of the International Conference on Computer Design (ICCD)*. New York City, USA, Oct. 2015, pp. 610–617

Example 2: Dark Silicon



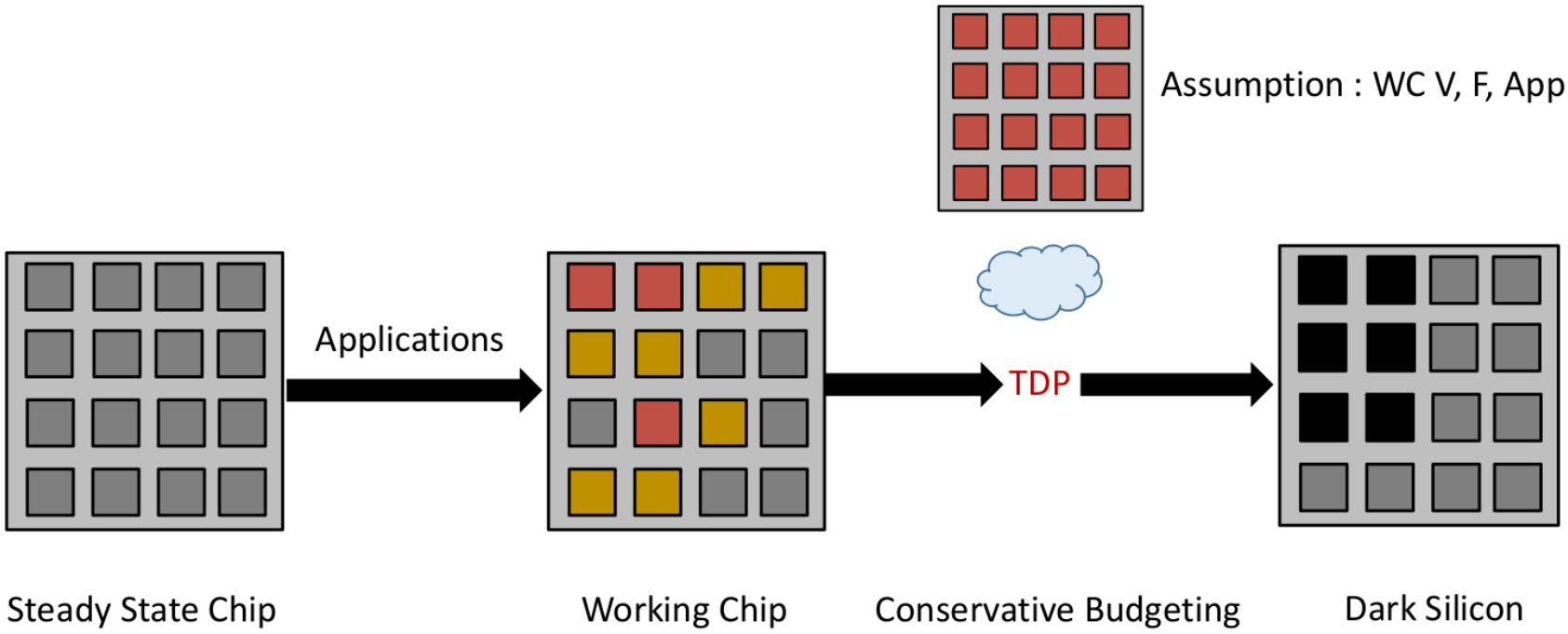
Example 2: Thermal Design Power



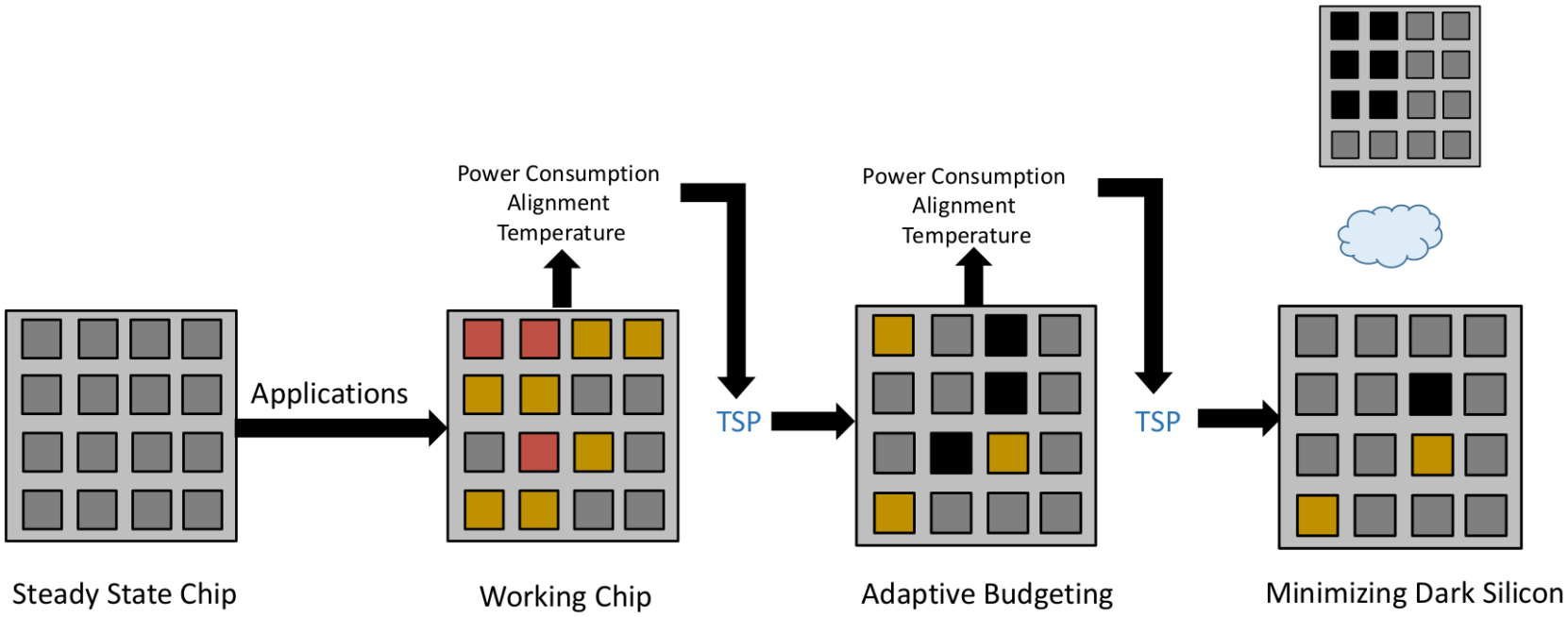
$$T(\text{core}) = f(\text{CorePower}, \text{AmbientTemp}, \text{NeighbourTemp})$$

TDP Thermal Design Power: Fixed power budget based on conservative design time estimate of the temperature.

Example 2: Fixed Power Budget



Example 2: Variable Power Budget

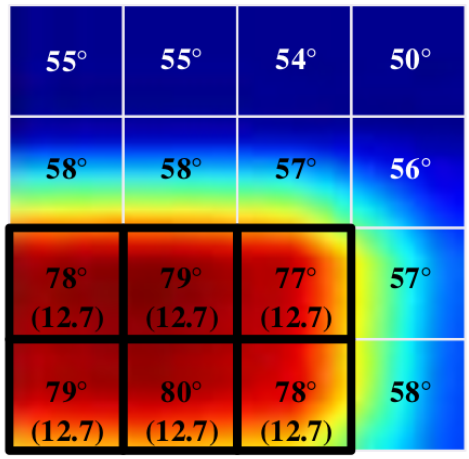


$$T(\text{core}) = f(\text{CorePower}, \text{AmbientTemp}, \text{NeighbourTemp})$$

TSP Thermal Saturation Power: Variable upper bound based on $T(\text{core})$.

Example 2: Efficient Budgeting

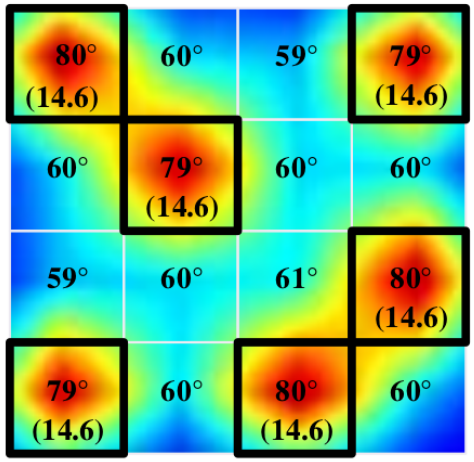
Tightly packed Cores



Neighbors accumulating temperature

Utilized Power Budget = **76.2 W**

Spreadout Cores

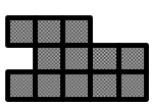


Neighbors dissipating temperature

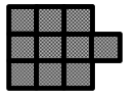
Utilized Power Budget = **87.6 W**

- ✓ 15% Better Utilization
- ✓ Activate more cores
- ✓ Reduce temperatures
- ✓ Minimize Dark Silicon

Example 2: Implications of Mapping

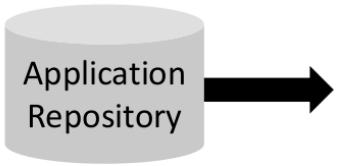


Application 1

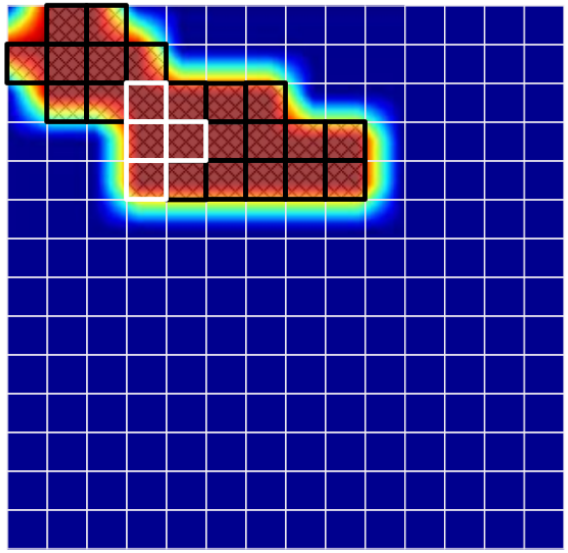


Application 2

Application 3

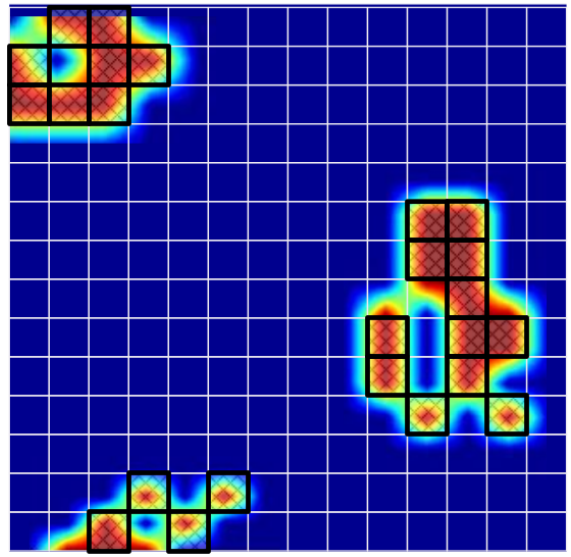


Tightly Packed – Greedy First Node



Power Budget = 66W

Spread Out – Adaptive First Node



Power Budget = 74.6W

Example 2: Power Budget Improvement

Percentage Power Budget Improvement for PAT over SC

Network Size	90% Dark		75% Dark		50% Dark	
	Avg.	Best	Avg.	Best	Avg.	Best
16x16	5.74	13.9	4.15	11.3	2.19	7.68
20x20	6.54	17.17	5.06	8.55	2.63	4.28

Percentage Power Budget Improvement for PAT over TSP-WC

Network Size	90% Dark		75% Dark		50% Dark	
	Avg.	Best	Avg.	Best	Avg.	Best
16x16	32.33	34.92	22.02	24.14	11.73	13.2
20x20	38.70	40.83	22.40	27.4	12.5	13.33

Example 2: Throughput Gain

Percentage Throughput gain for PAT over SC

Network Size	90% Dark		75% Dark		50% Dark	
	Avg.	Best	Avg.	Best	Avg.	Best
16x16	7.27	15.64	4.59	13.92	2.42	8.58
20x20	8.5	20.99	5.88	10.21	2.89	4.54

- ✓ Surplus Budget
- ➢ Added latency
- ✓ Minimal congestion
- ➢ Per Application Latency
- ✓ Per Chip Throughput

Example 3: Lifetime-Reliability-Driven Task Mapping

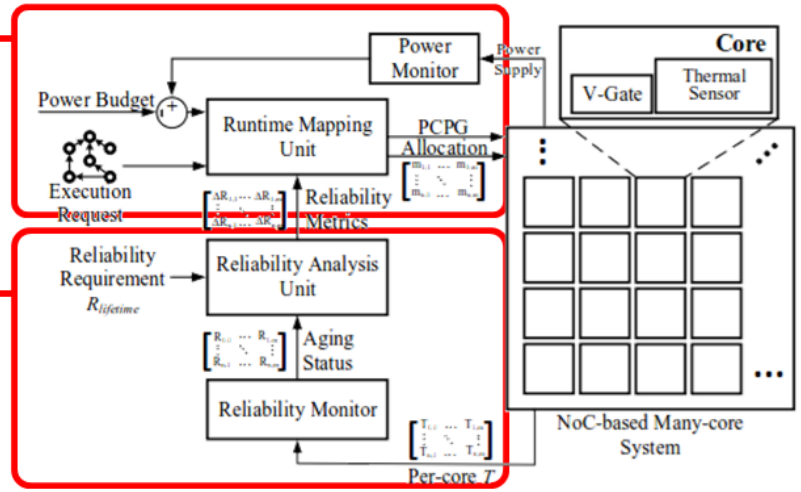
- To main limitations of many-cores:
  - Not enough power to turn on all cores (dark silicon)
  - Increased susceptibility of IC to aging and wear-out
- Goal: Introduce lifetime reliability awareness in the runtime resource management layer
  - Guarantee specified level of reliability
  - Satisfy the power budget
  - Optimize performance

M. H. Haghbayan, A. Miele, A. M. Rahmani, P. Liljeberg, and H. Tenhunen. "A lifetime-aware runtime mapping approach for many-core systems in the dark silicon era". In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2016, pp. 854–857

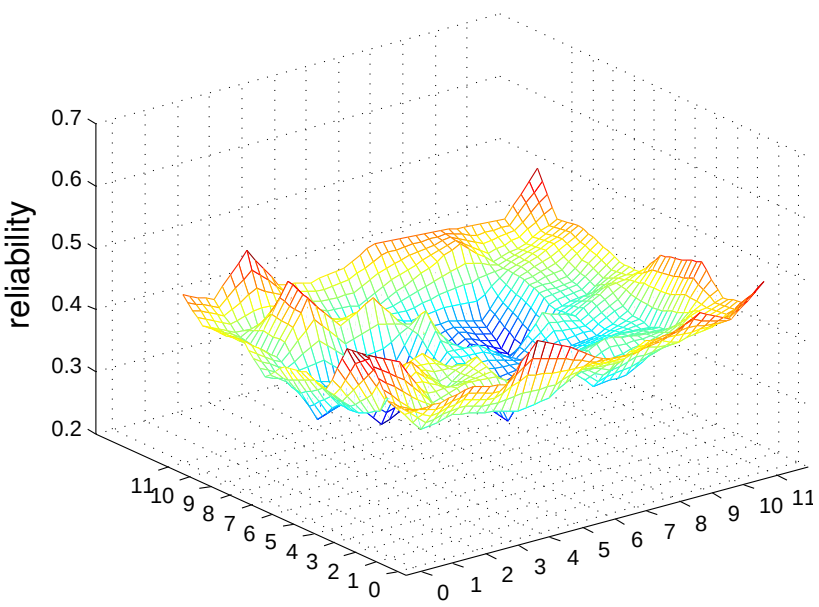
Example 3: Lifetime-Reliability-Driven Task Mapping

Proposed approach based on two feedback controllers

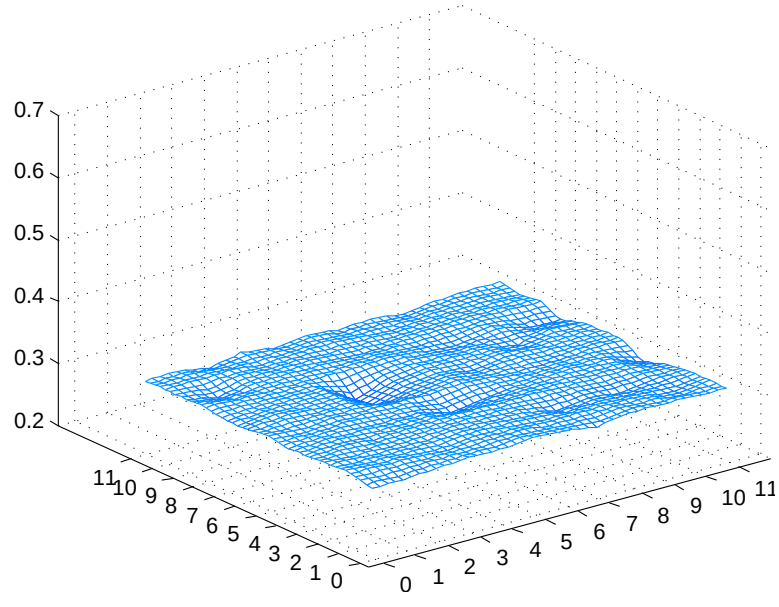
- Short-term controller
  - Application mapping
    - Select less aged cores
  - Power control
- Long-term controller
  - Reliability management
    - Compute current aging status
    - Disable highly stressed cores



Example 3: Lifetime-Reliability-Driven Task Mapping



MapPro:
lifetime=5.52 years



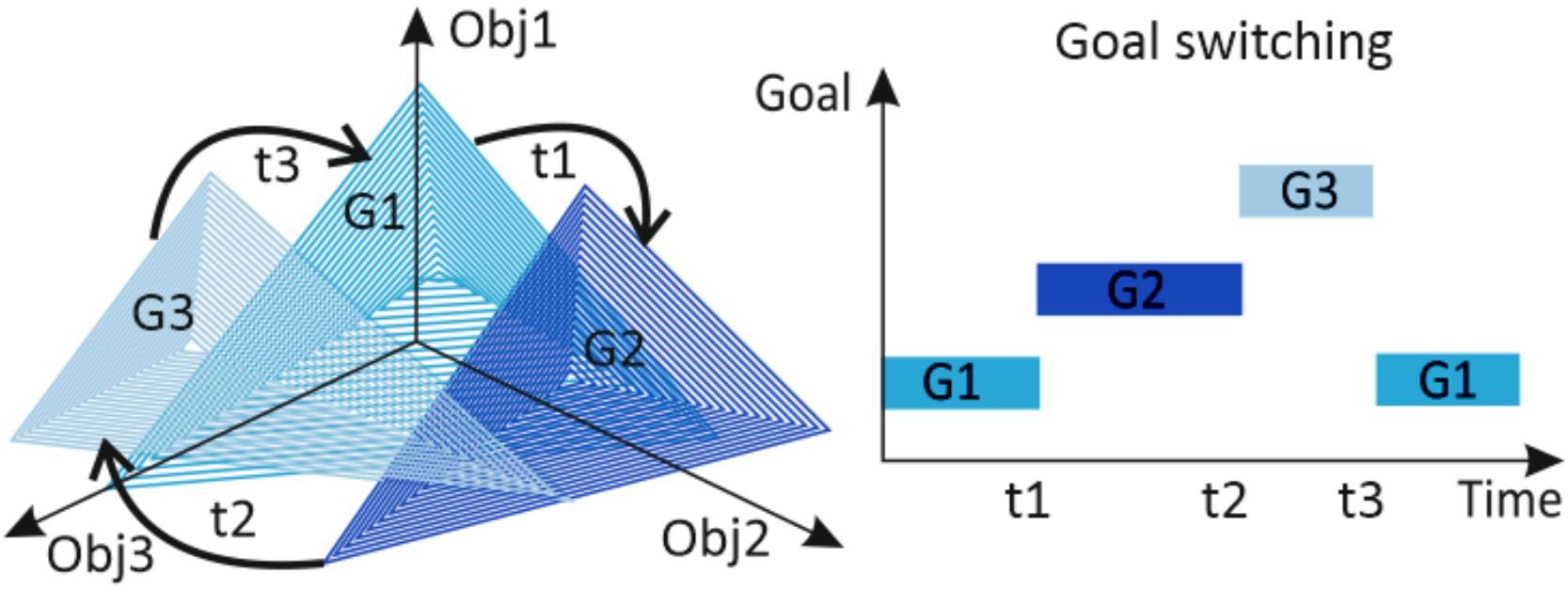
Reliability aware mapping:
lifetime=12 years

The plots show the reliability of cores at the end of the system's lifetime.
The end of the system's life is reached when the reliability of one core drops below 30%.

Challenges in Complex Many-Core SoCs

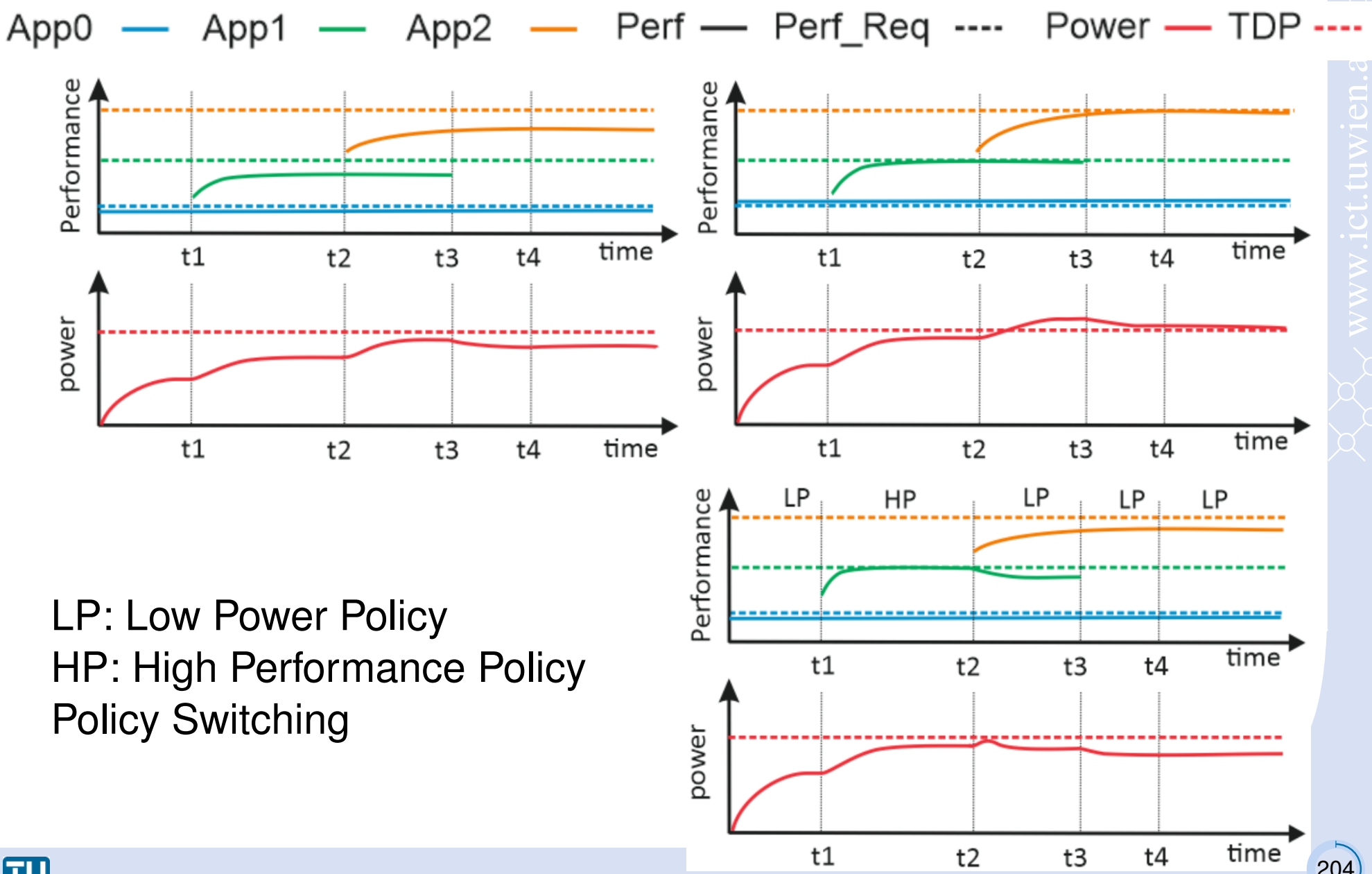
- A number and variety of objectives
  - Partially contradicting
  - At different time scales
- Objectives change over time
- The system state has to be known
- Application objectives have to be known

Goals and Objectives



Elham Shamsa, Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Axel Jantsch, and Nikil Dutt. “Goal Formulation: Abstracting Dynamic Objectives for Efficient On-chip Resource Allocation”. In: *IEEE Nordic Circuits and Systems Conference (NorCAS)*. Tallinn, Estonia, Oct. 2018

Time Varying Goals

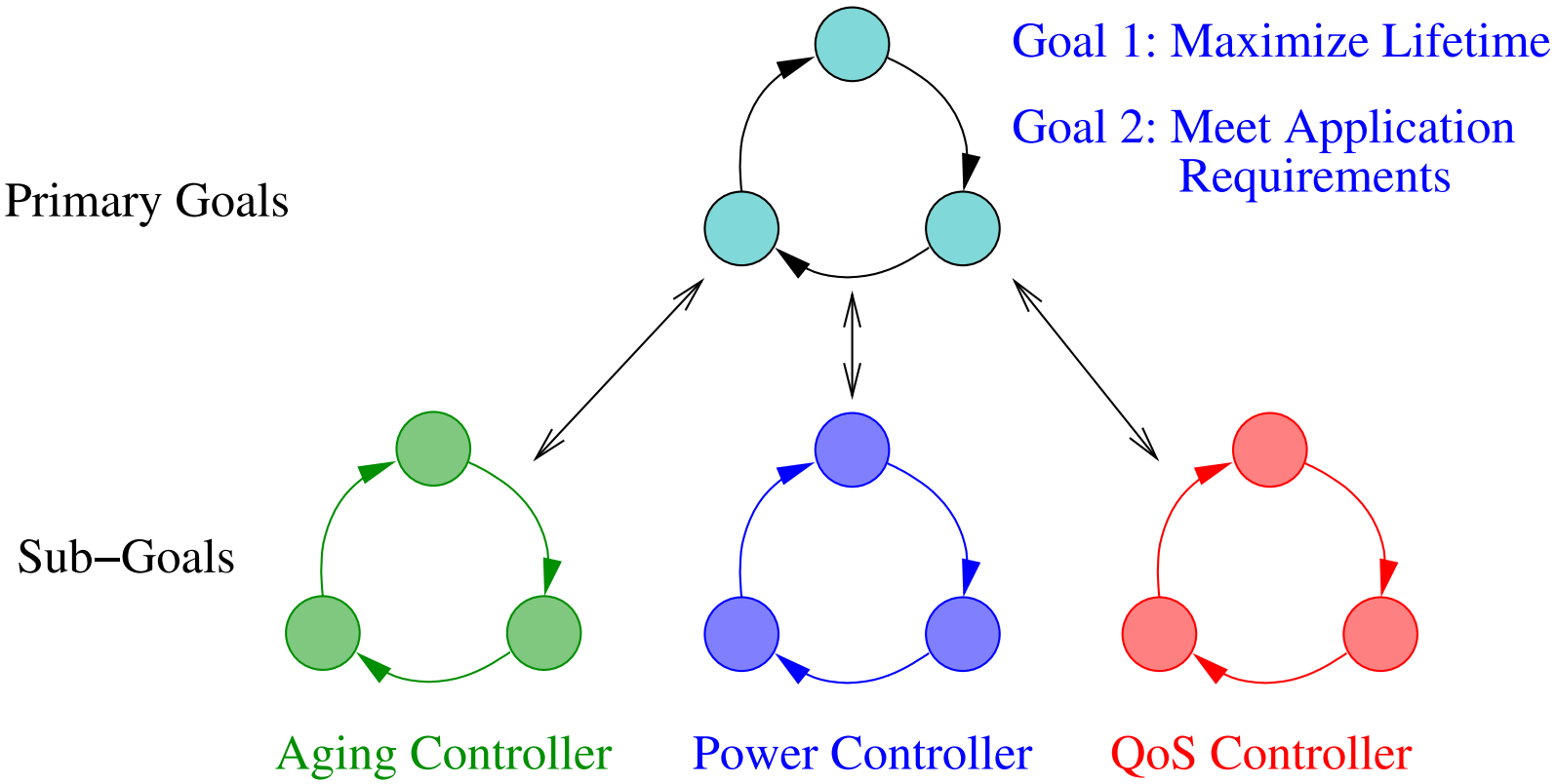


LP: Low Power Policy
HP: High Performance Policy
Policy Switching

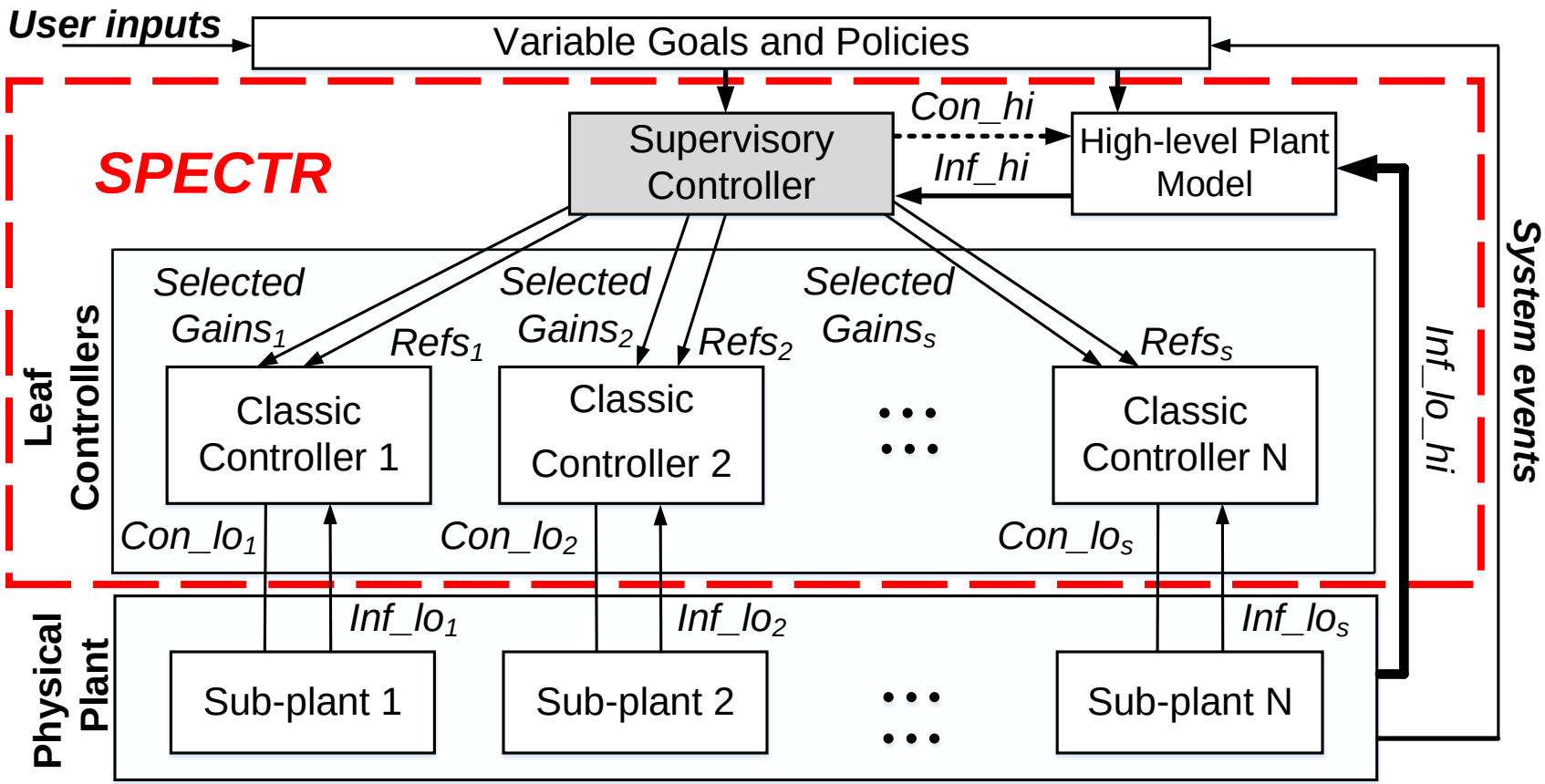
Goal Management Levels

1. Single objective; Design time;
2. Multiple objectives; Design time;
3. Multiple objectives; Run time;
4. Multiple, hierarchical objectives; Run time;

Hiararchical Goal Management

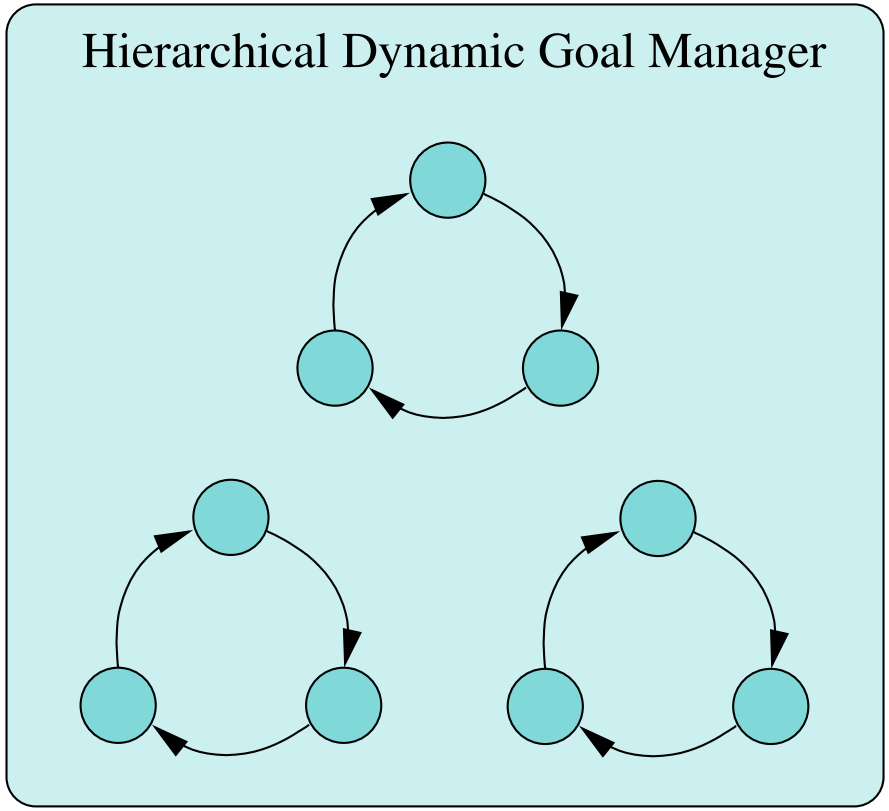


Supervisory Control

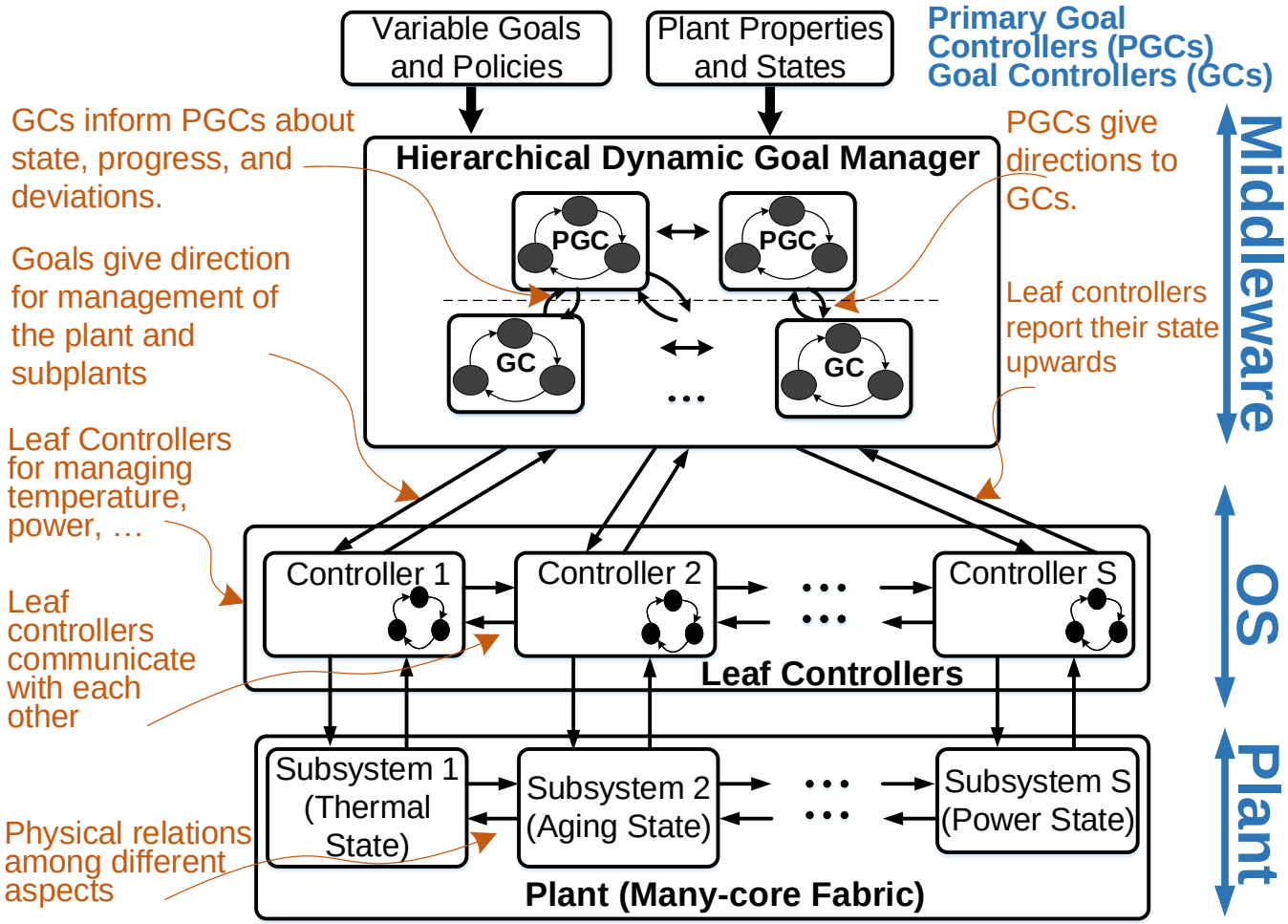


Amir M. Rahmani et al. "SPECTR - Formal Supervisory Control and Coordination for Many-core Systems Resource Management". In: *Proceedings of the 23rd ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. Williamsburg, VA, USA, Mar. 2018; T. R. Mück, B. Donyanavard, K. Moazzemi, A. M. Rahmani, A. Jantsch, and N. D. Dutt. "Design Methodology for Responsive and Robust MIMO Control of Heterogeneous Multicores". In: *IEEE Transactions on Multi-Scale Computing Systems* PP.99 (2018), pp. 1–1

Goal Management Inputs

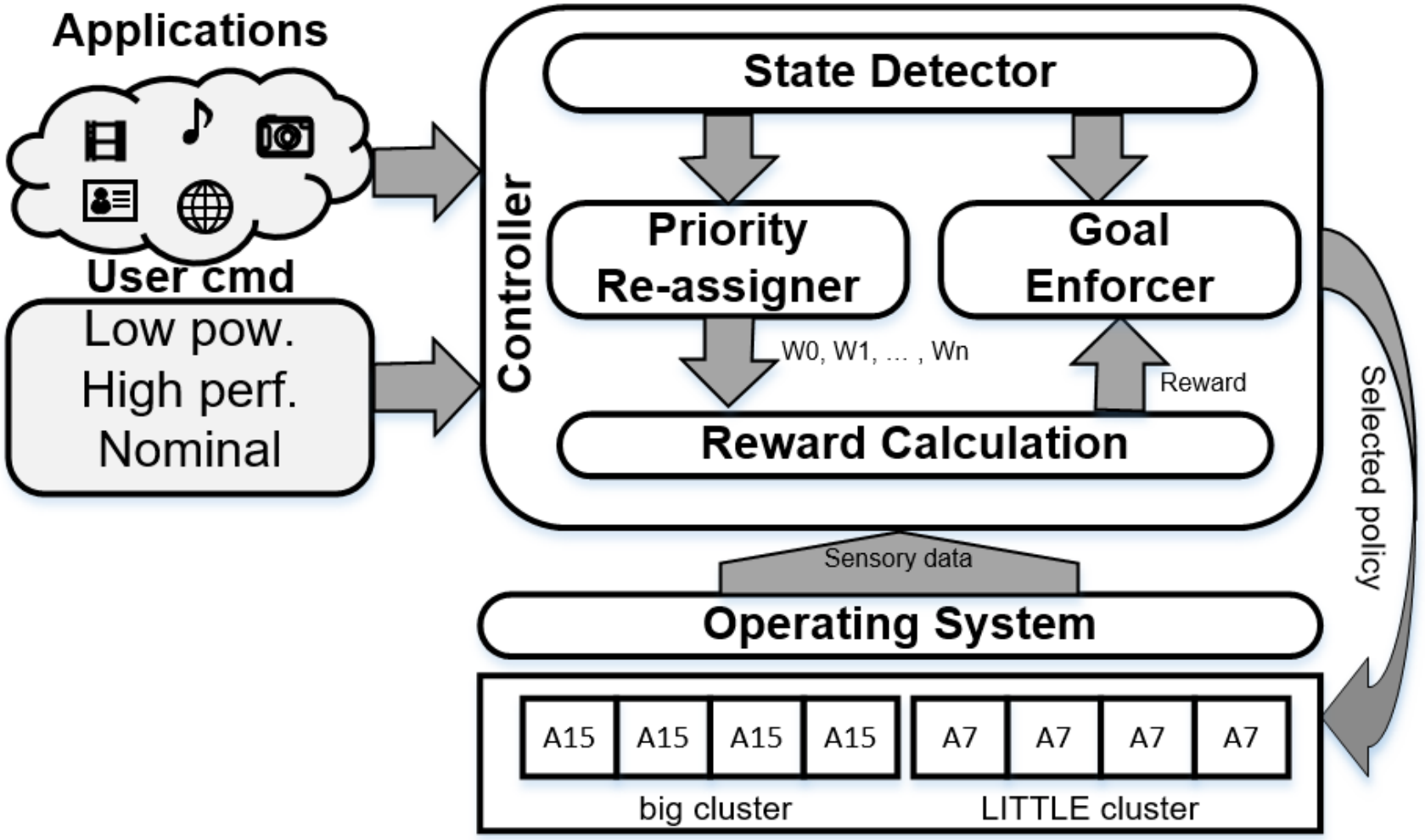


Hierarchical Goal Mangement



- The system's requirements changes over its lifetime.
- Different objectives are invoked at different time.

Goal Driven Autonomy



Elham Shamsa, Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Axel Jantsch, and Nikil Dutt. "Goal-Driven Autonomy for Efficient On-chip Resource Management: Transforming Objectives to Goals". In: *Proceedings of the Design and Test Europe Conference (DATE)*. Florence, Italy, Mar. 2019

Terminology

Agent is an actor in the system, that pursues specific objectives. $\mathcal{B} = \{B_1, B_2, B_3\}$

Application is an application running on the system. $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$.

Parameter: are entities measured and subject to control, like power consumption and application performance. E.g. $\mathcal{P}_{\text{pow}}(\text{core1})$, $\mathcal{P}_{\text{pow}}(\text{PLATFORM})$, $\mathcal{P}_{\text{perf}}(A_2)$.

Terminology

Objective function: either minimizes or maximizes a parameter or puts a constraint on a parameter. E.g.
$o_1 := \mathcal{P}_{\text{pow}}(\text{PLATFORM}) \rightarrow \min$,
$o_2 := \mathcal{P}_{\text{pow}}(\text{PLATFORM}) \leq C_1$,
$o_3 := \mathcal{P}_{\text{perf}}(A_1) \geq C_2$.

Objective is a set of objective functions, e.g. $O = \{o_1, o_2\}$.
$O(B_1)$ is the objective of agent B_1 and $O(A_1)$ is the objective of application A_1.
$O_{\mathcal{P}}(\mathcal{B}, \mathcal{A})$ is the set of objective functions of agents $\mathcal{B}$ and applications $\mathcal{A}$ relevant for parameter $\mathcal{P}$.
E.g. $O_{\text{pow}}(\{B_2, B_4\}, \{A_1, A_3\})$ is the set of objective functions of agents B_2 and B_4 and applications A_1 and A_3 relevant to power.

Terminology

Hierarchy Level: is a number assigned to actors; the higher the level, the more important is the actor.
E.g. $H(B_1) = 3$,
$H(\text{PLATFORM}) = 2$,
$H_P(\mathcal{B})$ is the highest hierarchy level of any agent in $\mathcal{B}$ which includes an objective function relevant for parameter P:

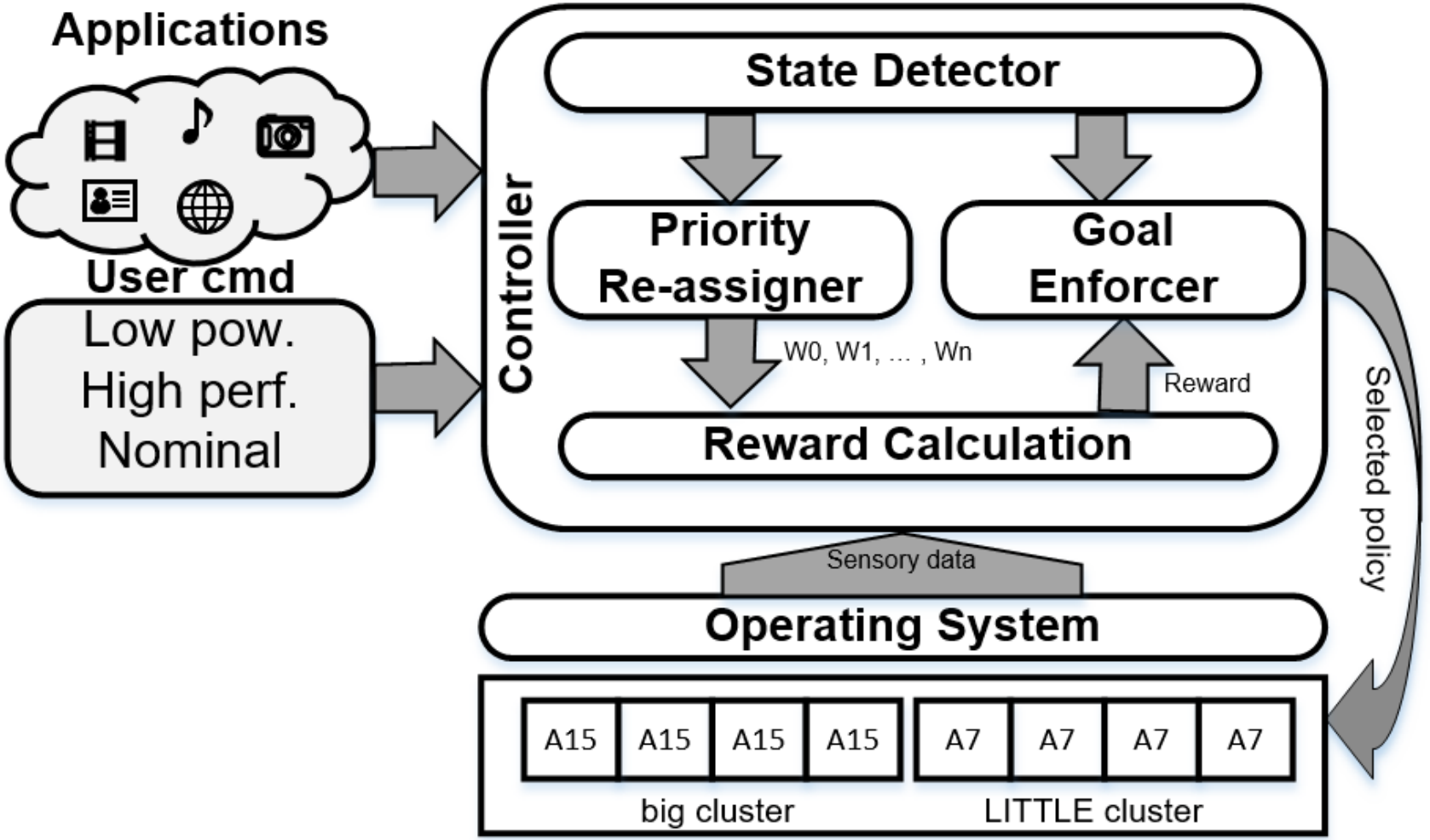
$$H_{\mathcal{P}}(\mathcal{B}) = \max_{B \in \mathcal{B}}(H(B)) \text{ for which } O_{\mathcal{P}}(\{B\}, \{\}) \neq \{\}$$

Terminology

Urgency of a parameter denotes the ratio of measured to desired value.
It is a function of all relevant objective functions for a parameter: $U_{\mathcal{P}}(O_{\mathcal{P}}) \rightarrow \mathbb{R}_{+}$.

Priority of a parameter is the product of its urgency and the highest hierarchy level of the involved agents: $\boldsymbol{P}_{\mathcal{P}} = U_{\mathcal{P}} H_{\mathcal{P}}$.

SoC Example



SoC Example

Agents: $\mathcal{B} = \{\text{USER}, \text{PLATFORM}, \text{APPLICATION}\}$

Applications: There ar n applications active:

$$\mathcal{A} = \{A_1, A_2, \ldots, A_n\}.$$

Parameters: $\mathcal{P}_{\text{pow}}, \mathcal{P}_{\text{perf}}$

Platform objectives:

$$O(\text{PLATFORM}) = \{\mathcal{P}_{\text{pow}}(\text{PLATFORM}) \leq \text{TDP}\}$$

User Objectives:

$$O(\text{USER}) = \begin{cases} \{\mathcal{P}_{\text{perf}}(\text{PLATFORM}) \rightarrow \max\} \\ \quad \text{if user command = "High Performance"} \\ \{\mathcal{P}_{\text{pow}}(\text{PLATFORM}) \rightarrow \min\} \\ \quad \text{if user command = "Low Power"} \end{cases}$$

Application Objectives: A minimum ($C_A^{\min}$) and a maximum ($C_A^{\max}$) performance constraint is given for each application A:

$$O_A = \{\mathcal{P}_{\text{perf}}(A) \leq C_A^{\max}, \mathcal{P}_{\text{perf}}(A) \geq C_A^{\min}\}$$

State Detection

State vector:

- **Power**: Violation: $\text{TDP} < p$
  Potential Violation: $0.8\ \text{TDP} \leq p \leq \text{TDP}$
  No Violation: $p \leq 0.8\ \text{TDP}$
- **User Command**: High Performance
  Low Power
- **Performance per application**: [Min run time,
  Max run time]

Goal Hierarchy

$$
\begin{aligned}
H(\text{PLATFORM}) &= \begin{cases} 5 & \text{if } \mathcal{P}_{\text{pow,cur}} > 0.9\text{TDP} \\ 2.5 & \text{if } 0.9\text{TDP} > \mathcal{P}_{\text{pow,cur}} > 0.8\,\text{TDP} \\ 1 & \text{if } \mathcal{P}_{\text{pow,cur}} < 0.8\text{TDP} \end{cases} \\
H(\text{USER}) &= 2 \\
H(\text{APPLICATION}) &= (1 + \frac{n_{viol}}{n})
\end{aligned}
$$

Priority Assignment

- Primary goals: thermal safety
- Secondary goals: User experience
- Tertiary goals: Application requirements

Priority Assignment - Urgency

Urgency is the extent of a violation of a parameter:

$$U_{\mathcal{P}_{\text{pow}}}(\{\mathcal{P}_{\text{pow}}(\text{PLATFORM}) \leq \text{TDP}, \mathcal{P}_{\text{pow}}(\text{PLATFORM}) \rightarrow \text{min}\}) = \frac{\mathcal{P}_{\text{pow,cur}}}{\text{TDP}}$$

$\mathcal{P}_{\text{pow,cur}}$	is the instantaneous power consumption;
TDP	is Thermal Design Power.

Priority Assignment - Urgency

Case 1: if User Command = “High Performance”:

$$U_{\mathcal{P}_{\text{perf}}}(A, \{\mathcal{P}_{\text{perf}}(A) \leq C_A^{\max}, \mathcal{P}_{\text{perf}}(A) \geq C_A^{\min}, \mathcal{P}_{\text{perf}}(\text{PLATFORM}) \rightarrow \max\})$$
$$= \frac{C_A^{\max} - \mathcal{P}_{\text{perf,cur}}(A)}{C_A^{\max} - C_A^{\text{ref}^1}}$$
$$C_A^{\text{ref}^1} = \frac{C_A^{\max} + C_A^{min}}{2}$$

Case 2: if User Command = “Low Power”:

$$U_{\mathcal{P}_{\text{perf}}}(A, \{\mathcal{P}_{\text{perf}}(A) \leq C_A^{\max}, \mathcal{P}_{\text{perf}}(A) \geq C_A^{\min}\})$$
$$= \frac{C_A^{\max} - \mathcal{P}_{\text{perf,cur}}(A)}{C_A^{\max} - C_A^{\text{ref}^2}}$$
$$C_A^{\text{ref}^2} = C_A^{\min}$$

Priority

$$\boldsymbol{P}_{\mathcal{P}_{\text{pow}}} = U_{\mathcal{P}_{\text{pow}}} H_{\mathcal{P}_{\text{pow}}} = \frac{\mathcal{P}_{\text{pow,cur}}}{\text{TDP}} \cdot \max(H(\text{USER}), H(\text{PLATFORM}))$$

$$\boldsymbol{P}_{\mathcal{P}_{\text{perf}}}(A) = U_{\mathcal{P}_{\text{perf}}}(A) H_{\mathcal{P}_{\text{perf}}}$$

$$= \begin{cases} \frac{C_A^{\max} - \mathcal{P}_{\text{perf,cur}}(A)}{C_A^{\max} - C_A^{\text{ref}1}} \cdot \max(H(\text{USER}), H(\text{APPLICATION})) \\ \qquad \text{(if User Command = ""High Performance")} \\ \frac{C_A^{\max} - \mathcal{P}_{\text{perf,cur}}(A)}{C_A^{\max} - C_A^{\text{ref}2}} \cdot H(\text{APPLICATION})) \\ \qquad \text{(if User Command = "Low Power")} \end{cases}$$

Goal Enforcement

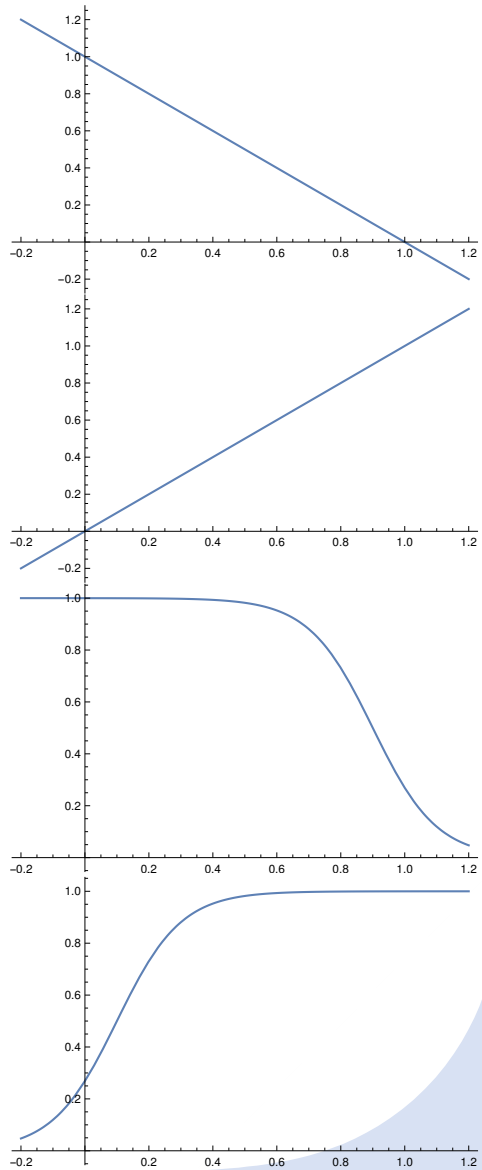
- Selects action that most likely will satisfy the highest priority goal;
- Action = Resource allocation policy;
- Initial action is randomly selected;
- Actions are assessed in a reinforcement learning loop;
- Reinforcement learning is based on a reward function.

Rewards

- A function is assigned to every objective function.
- A $[\min, \max]$ interval is assumed for each reward function.
- The reward function is normalized to $[0, 1] \rightarrow [0.1]$.
- For minimizing and maximizing a linear reward function is used.
- For bounds a variant of the generalized logistic function is used:
  - $R(x) = \frac{1}{(1+e^{B(x-A)})^C}$
  - For lower bound constraints: $A = 0.1, B = -10, C = 1$.
  - For upper bound constraints: $A = 0.9, B = 10, C = 1$.

Reward Functions

Objective function	Reward function
$y = f(x) \rightarrow \min$	$R_{\min}(y') = -y'$
$y = f(x) \rightarrow \max$	$R_{\max}(y') = y'$
$y = f(x) \leq C_{\max}$	$R_{\text{ub}}(y') = \frac{1}{(1+e^{10(y'-0.9)})^1}$
$y = f(x) \geq C_{\min}$	$R_{\text{lb}}(y') = \frac{1}{(1+e^{-10(y'-0.1)})^1}$



y' is y normalized to $[0, 1]$.

Reward Calculation

$$Reward = W_0 R_0 + W_1 R_1 + W_2 R_2 + ... + W_n R_n$$

With the objective functions for power and performance:

$$R = W_{\mathcal{P}_{\text{pow}}} \cdot R_{\mathcal{P}_{\text{pow}}} + \sum_{A \in \mathcal{A}} W_{\mathcal{P}_{\text{perf}}}(A) \cdot R_{\mathcal{P}_{\text{perf}}}(A)$$

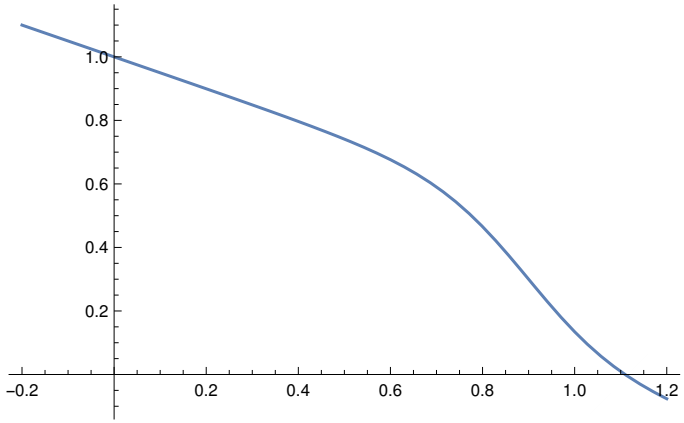
Reward Calculation for Power

Objective Functions for power with user command “Low Power”:

$$
\begin{aligned}
O_{\mathcal{P}_{\text{pow}}}(\mathcal{B}, \mathcal{A}) &= O(\text{PLATFORM}) \cup O(\text{USER}) \\
&= \{\mathcal{P}_{\text{pow}}(\text{PLATFORM}) \leq \text{TDP}, \\
&\quad \mathcal{P}_{\text{pow}}(\text{PLATFORM}) \rightarrow \min\}
\end{aligned}
$$

Reward function:

$$
\begin{aligned}
R_{\mathcal{P}_{\text{pow}}} &= \frac{1}{2}(R_{\text{min}}(y') + R_{\text{ub}}(y')) \\
&= \frac{1}{2}(-y + \frac{1}{1 + e^{10((y'-0.9)}})
\end{aligned}
$$



where y' is the normalized $\mathcal{P}_{\text{pow,cur}}$.

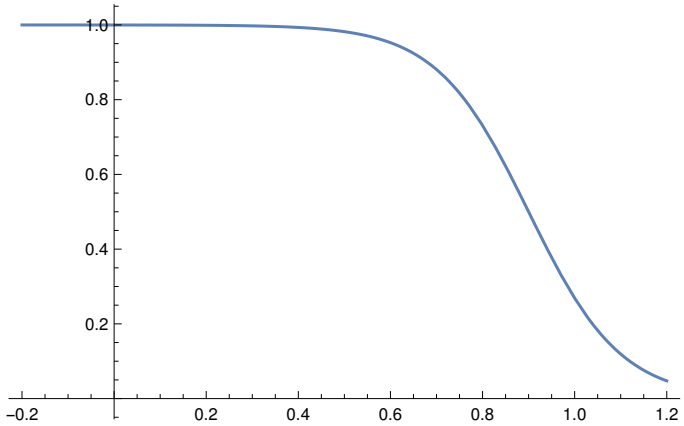
Reward Calculation for Power

Objective Functions for power with user command “High Performance”:

$$\begin{aligned} O_{\mathcal{P}_{\text{pow}}}(\mathcal{B}, \mathcal{A}) &= O(\text{PLATFORM}) \cup O(\text{USER}) \\ &= \{\mathcal{P}_{\text{pow}}(\text{PLATFORM}) \leq \text{TDP}\} \end{aligned}$$

Reward function:

$$\begin{aligned} R_{\mathcal{P}_{\text{pow}}} &= R_{\text{ub}}(y')) \\ &= \frac{1}{1 + e^{10((y'-0.9)}} \end{aligned}$$



where y' is the normalized $\mathcal{P}_{\text{pow,cur}}$.

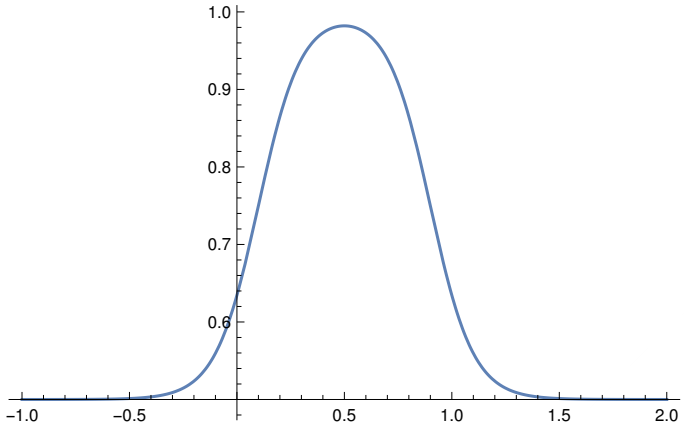
Reward Calculation for Performance

Objective Functions for performance with user command “Low Power”:

$$\begin{aligned} O_{\mathcal{P}_{\text{perf}}}(\mathcal{B}, \{A\}) &= O_{\mathcal{P}_{\text{perf}}}(\text{USER}) \cup O_{\mathcal{P}_{\text{perf}}}(\{A\}) \\ &= \{\mathcal{P}_{\text{perf}}(A) \leq C_A^{\max}, \mathcal{P}_{\text{perf}}(A) \geq C_A^{\min}\} \end{aligned}$$

Reward function for A:

$$\begin{aligned} R_{\mathcal{P}_{\text{perf}}}(A) &= \frac{1}{2}(R_{\text{lb}}(y') + R_{\text{ub}}(y')) \\ &= \frac{1}{2}\left(\frac{1}{1 + e^{10((y'-0.9)}} + \frac{1}{1 + e^{-10((y'-0.1)}}\right) \end{aligned}$$



where y' is the normalized $\mathcal{P}_{\text{perf,cur}}(A)$.

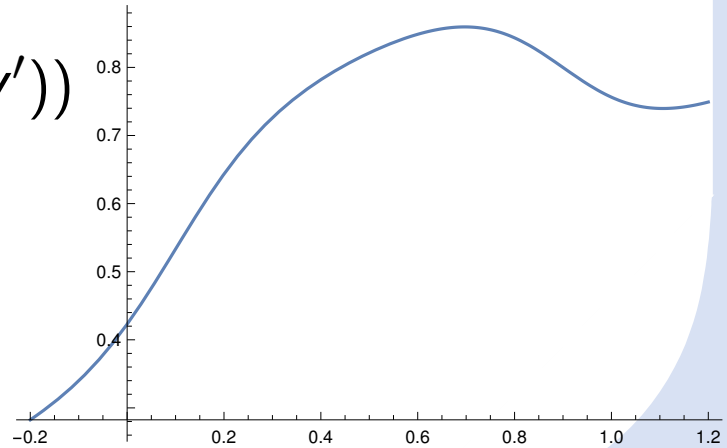
Reward Calculation for Performance

Objective Functions for performance with user command “High Performance”:

$$
\begin{aligned}
O_{\mathcal{P}_{\text{perf}}}(\mathcal{B}, \{A\}) &= O_{\mathcal{P}_{\text{perf}}}(\text{USER}) \cup O_{\mathcal{P}_{\text{perf}}}(\{A\}) \\
&= \{\mathcal{P}_{\text{perf}}(A) \rightarrow \max, \\
&\quad \mathcal{P}_{\text{perf}}(A) \leq C_A^{\max}, \mathcal{P}_{\text{perf}}(A) \geq C_A^{\min}\}
\end{aligned}
$$

Reward function for A:

$$
\begin{aligned}
R_{\mathcal{P}_{\text{perf}}}(A) &= \frac{1}{3}(R_{\max}(y') + R_{\text{lb}}(y') + R_{\text{ub}}(y')) \\
&= \frac{1}{3}(y' + \frac{1}{1 + e^{10((y'-0.9)}} \\
&\quad + \frac{1}{1 + e^{-10((y'-0.1)}})
\end{aligned}
$$



where y' is the normalized $\mathcal{P}_{\text{perf,cur}}(A)$.

Reward Calculation

$$\begin{aligned} R &= W_0 \times R_0 + W_1 \times R_1 + W_2 \times R_2 + \ldots + W_n \times R_n \\ &= W_{\mathcal{P}_{\text{pow}}} \cdot R_{\mathcal{P}_{\text{pow}}} + \sum_{A \in \mathcal{A}} W_{\mathcal{P}_{\text{perf}}}(A) \cdot R_{\mathcal{P}_{\text{perf}}}(A) \end{aligned}$$

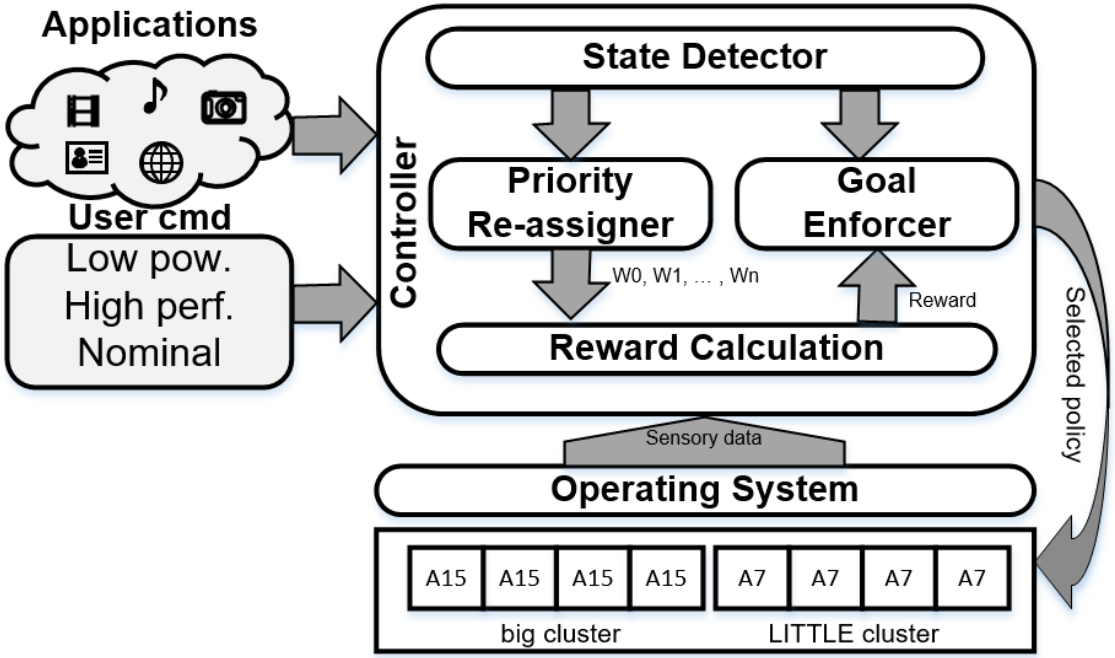
For the weights we use priorities:

$$\begin{aligned} \boldsymbol{P}_{\mathcal{P}_{\text{pow}}} &= U_{\mathcal{P}_{\text{pow}}} H_{\mathcal{P}_{\text{pow}}} = \frac{\mathcal{P}_{\text{pow,cur}}}{\text{TDP}} \cdot \max(H(\text{USER}), H(\text{PLATFORM})) \\ \boldsymbol{P}_{\mathcal{P}_{\text{perf}}}(A) &= U_{\mathcal{P}_{\text{perf}}}(A) H_{\mathcal{P}_{\text{perf}}} \end{aligned}$$

Thus:

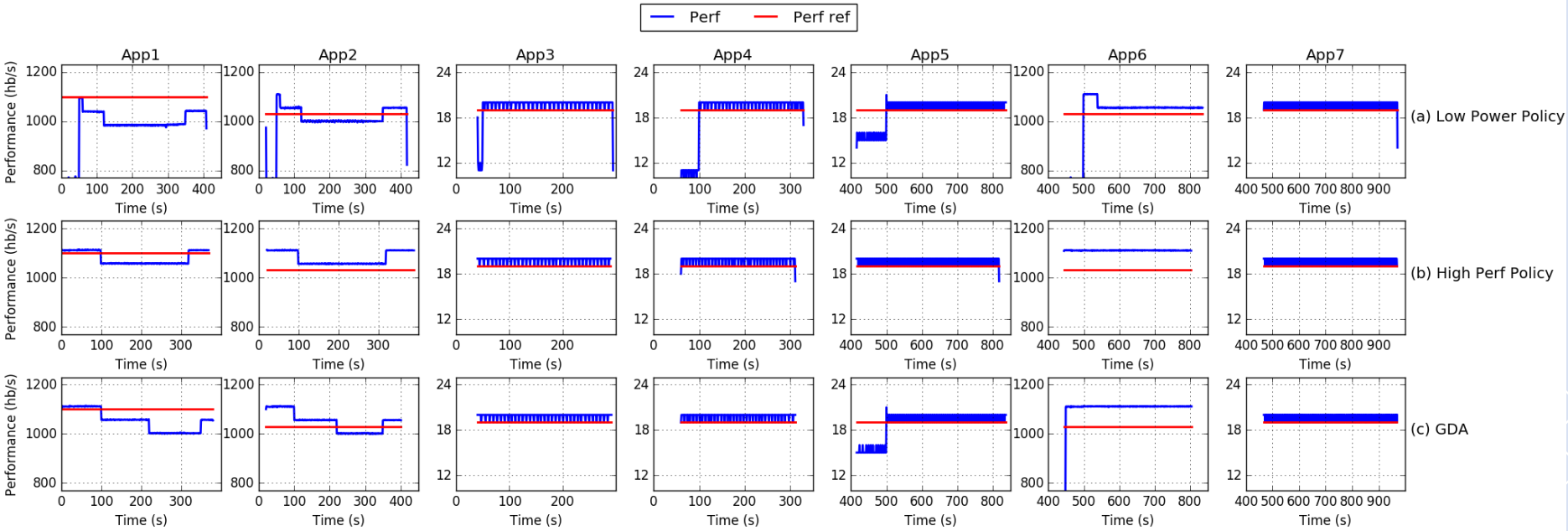
$$R = \boldsymbol{P}_{\mathcal{P}_{\text{pow}}} \cdot R_{\mathcal{P}_{\text{pow}}} + \sum_{A \in \mathcal{A}} \boldsymbol{P}_{\mathcal{P}_{\text{perf}}}(A) \cdot R_{\mathcal{P}_{\text{perf}}}(A)$$

Q-Learning



- Actions are task migration, cluster DVFS;
- Rewards are updated;
- Actions with highest rewards are executed;
- Initially, actions are selected randomly.

Experiments

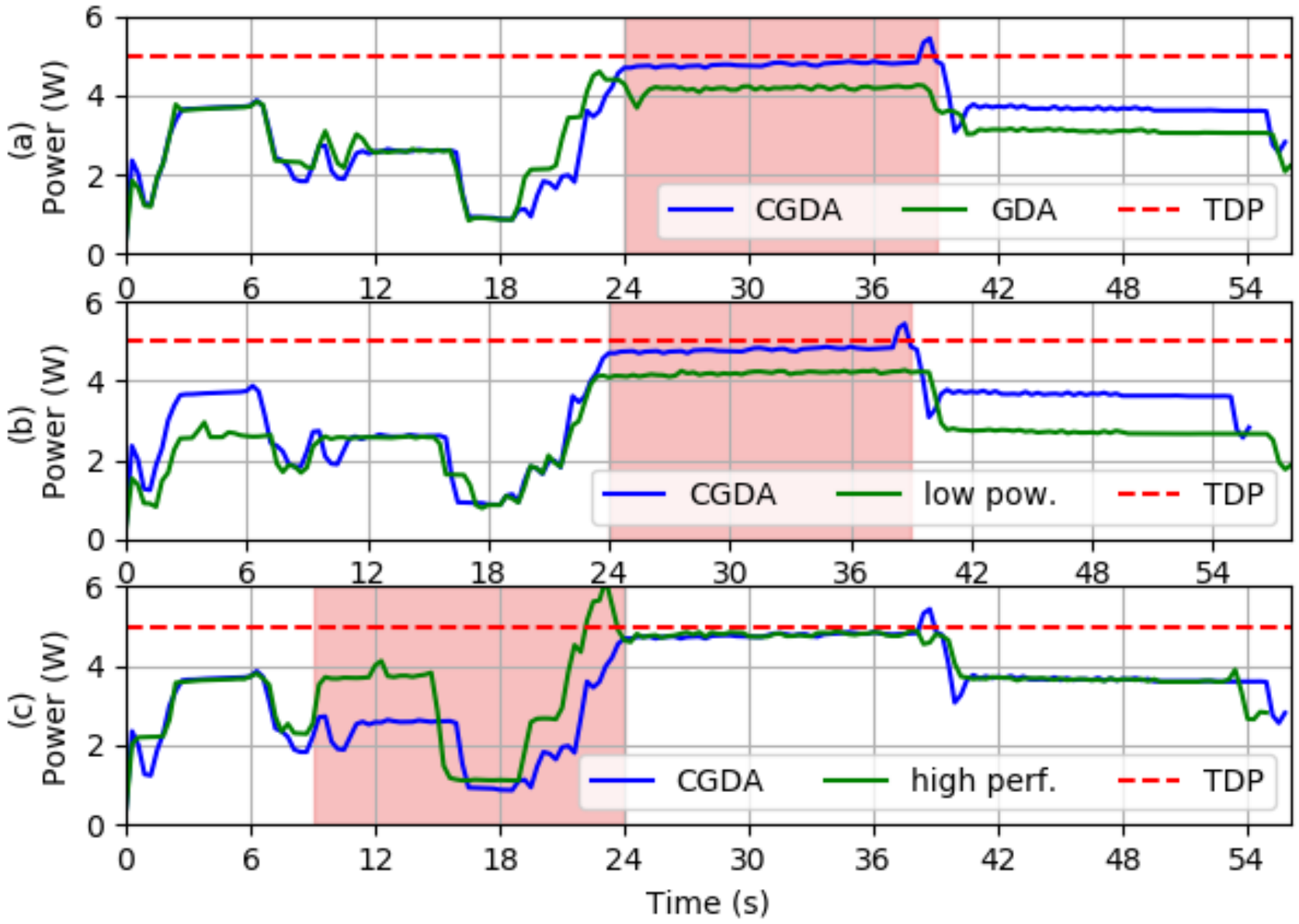


Experiments with a set of microkernel benchmarks;
Hardkernel Odroid XU3 board,
with two clusters (4 big (A15) and 4 little (A7) CPU cores);
Performance in heartbeats/sec.

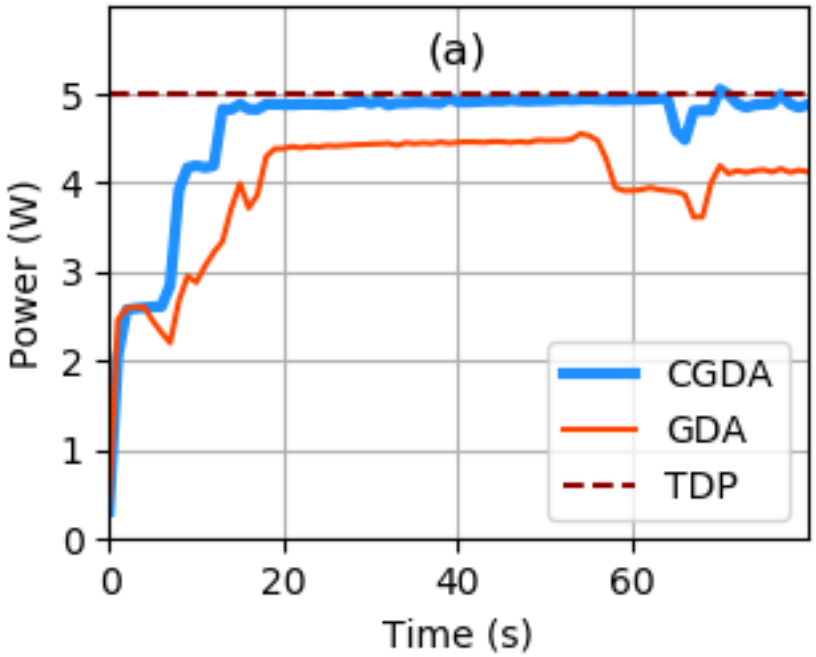
Comparison

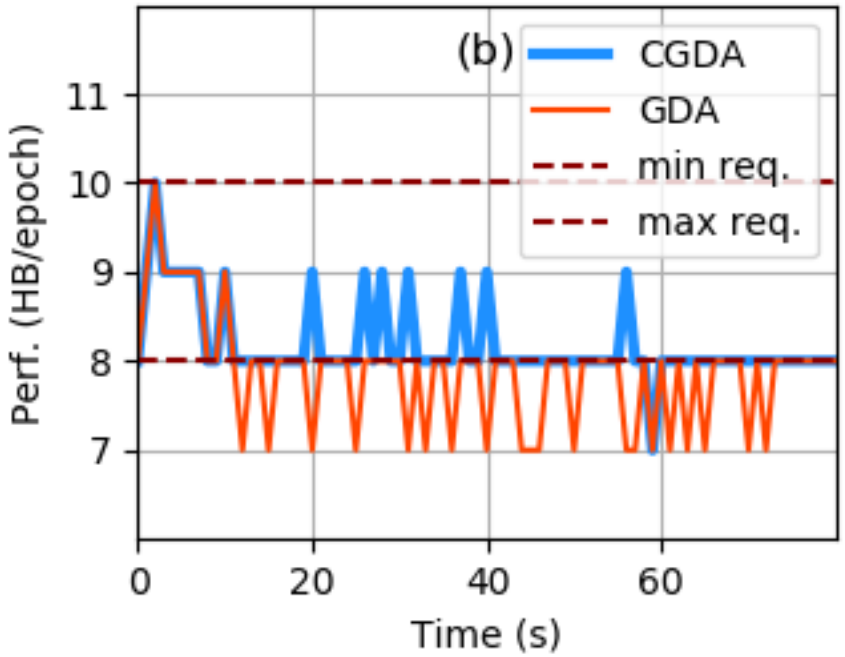
Tech.	Obj	Cmd	Pwr viol.	Perf. viol.	Avg. pwr (W)
LP	Power	X	0%	27%	2.86
HP	Perf.	X	3%	0%	3.7
GDA	Dynamic	✓	0%	14%	3.1
CGDA	Dynamic	✓	1%	2%	3.4

Experiments - Power Evaluation



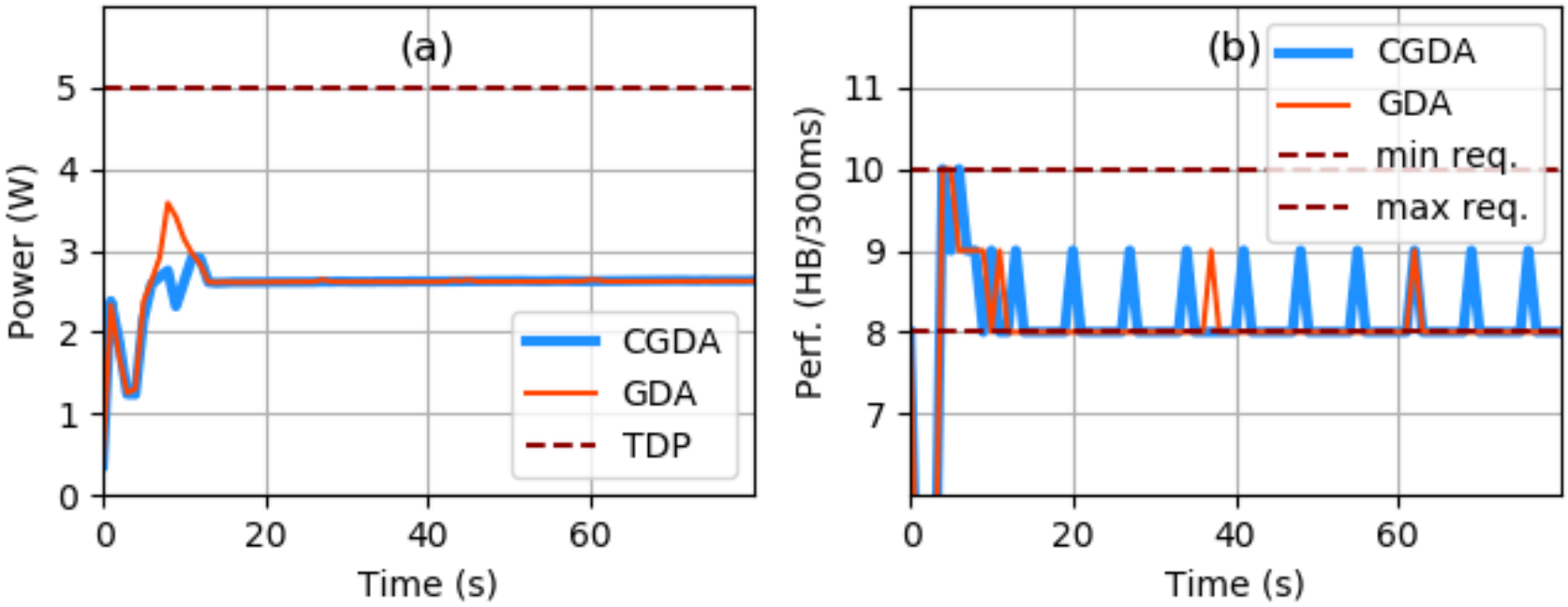
Experiments - Load Evaluation I





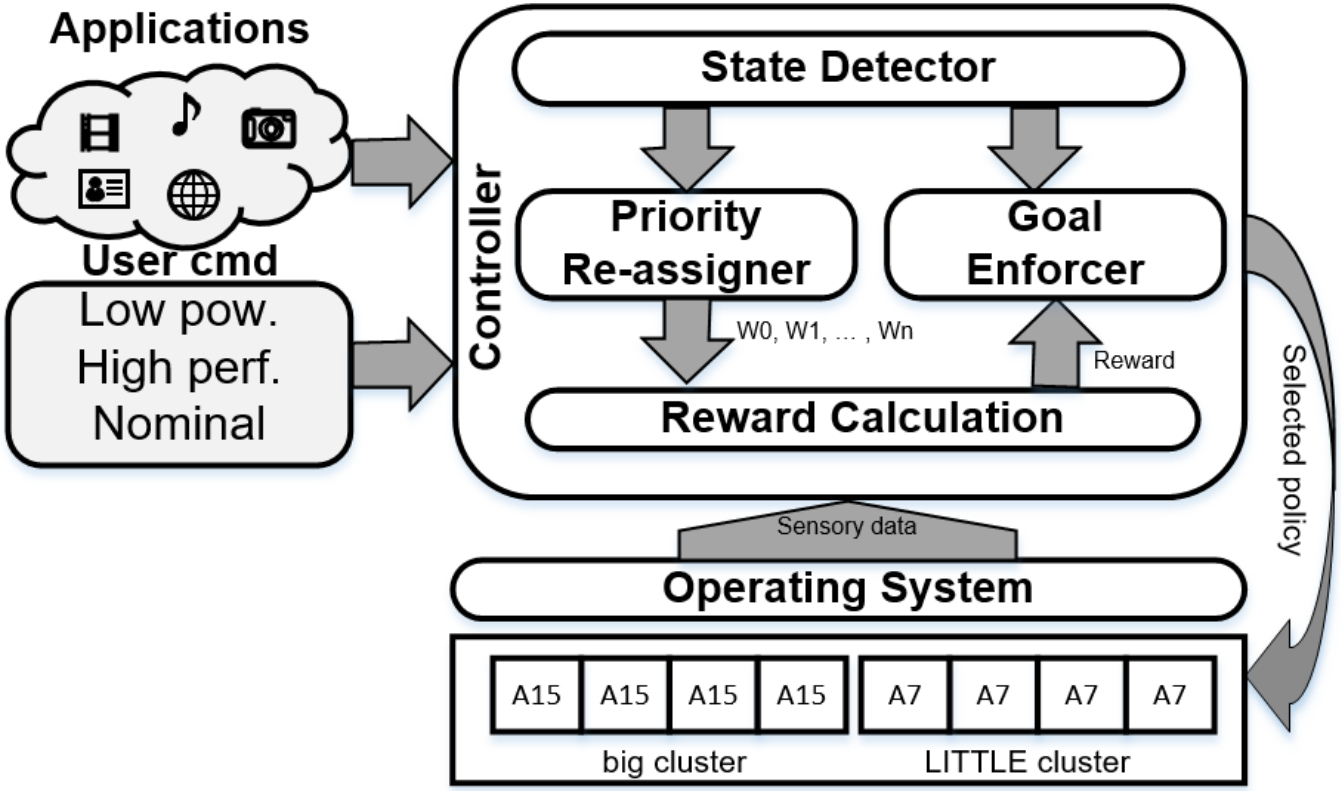
High Load Scenario

Experiments - Load Evaluation II



Low Load Scenario

Goal Driven Autonomy



Elham Shamsa, Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Axel Jantsch, and Nikil Dutt. "Goal-Driven Autonomy for Efficient On-chip Resource Management: Transforming Objectives to Goals". In: *Proceedings of the Design and Test Europe Conference (DATE)*. Florence, Italy, Mar. 2019

Axel Jantsch et al. "Hierarchical Dynamic Goal Management for IoT Systems". In: *Proceedings of the IEEE International Symposium on Quality Electronic Design (ISQED 2018)*. USA, Mar. 2018

Amir M. Rahmani, Axel Jantsch, and Nikil Dutt. "HDGM: Hierarchical Dynamic Goal Management for Many-Core Resource Allocation". In: *IEEE Embedded Systems letters* 10.3 (Sept. 2018)

Goal Driven Autonomy

Summary

- Framework for managing various different goals and objectives;
- Goals can dynamically change;
- Actions are improved during operation based on reinforcement learning.

Examples

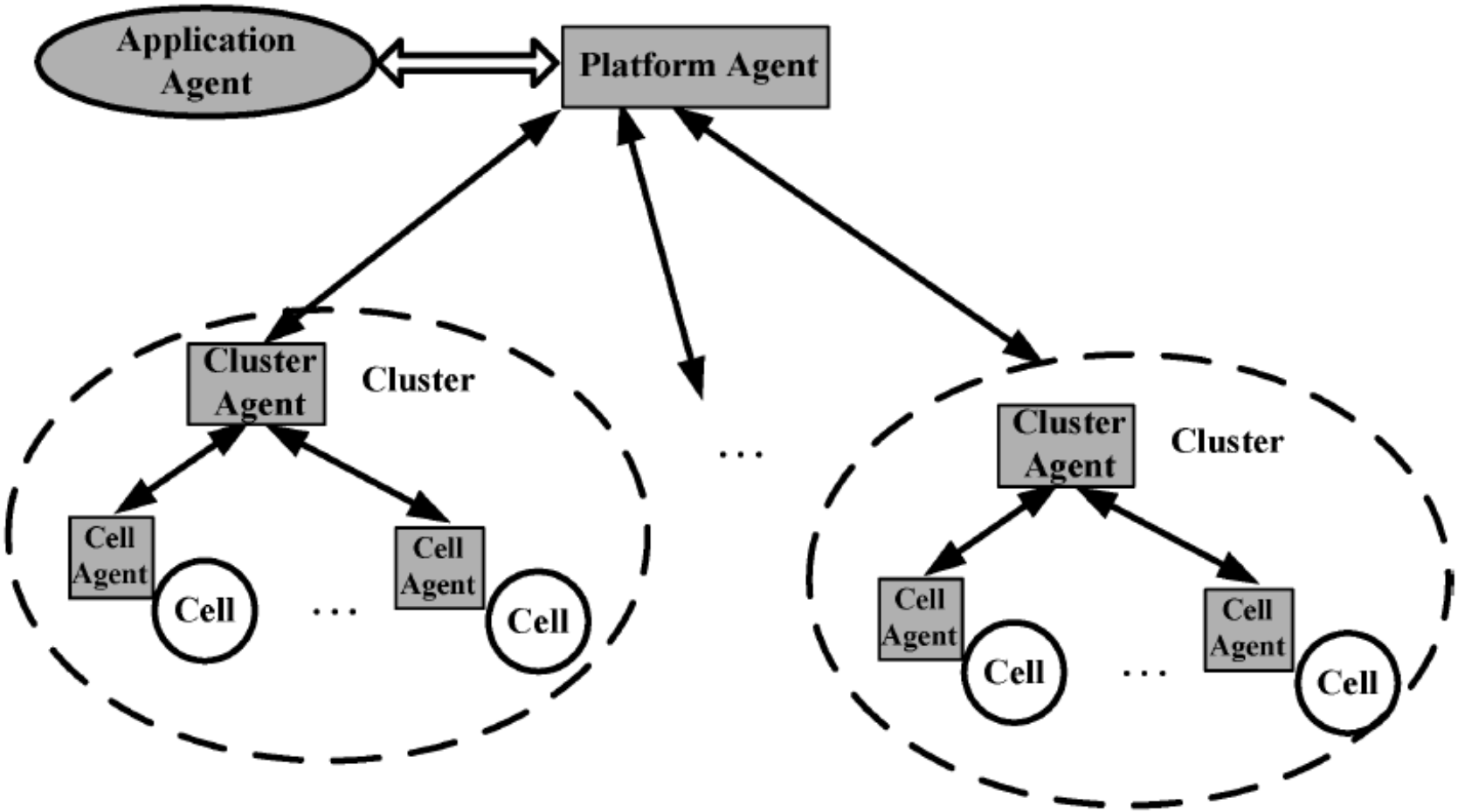
HAMSoC - A Hierarchical Agent Monitored System on Chip

- Self-monitoring design platform for multi-core SoCs
- Three levels of agents: cell, cluster, platform
- Dedicated design layer for self-awareness and adaptivity
- Application: Power management in NoC based multi-core SoC

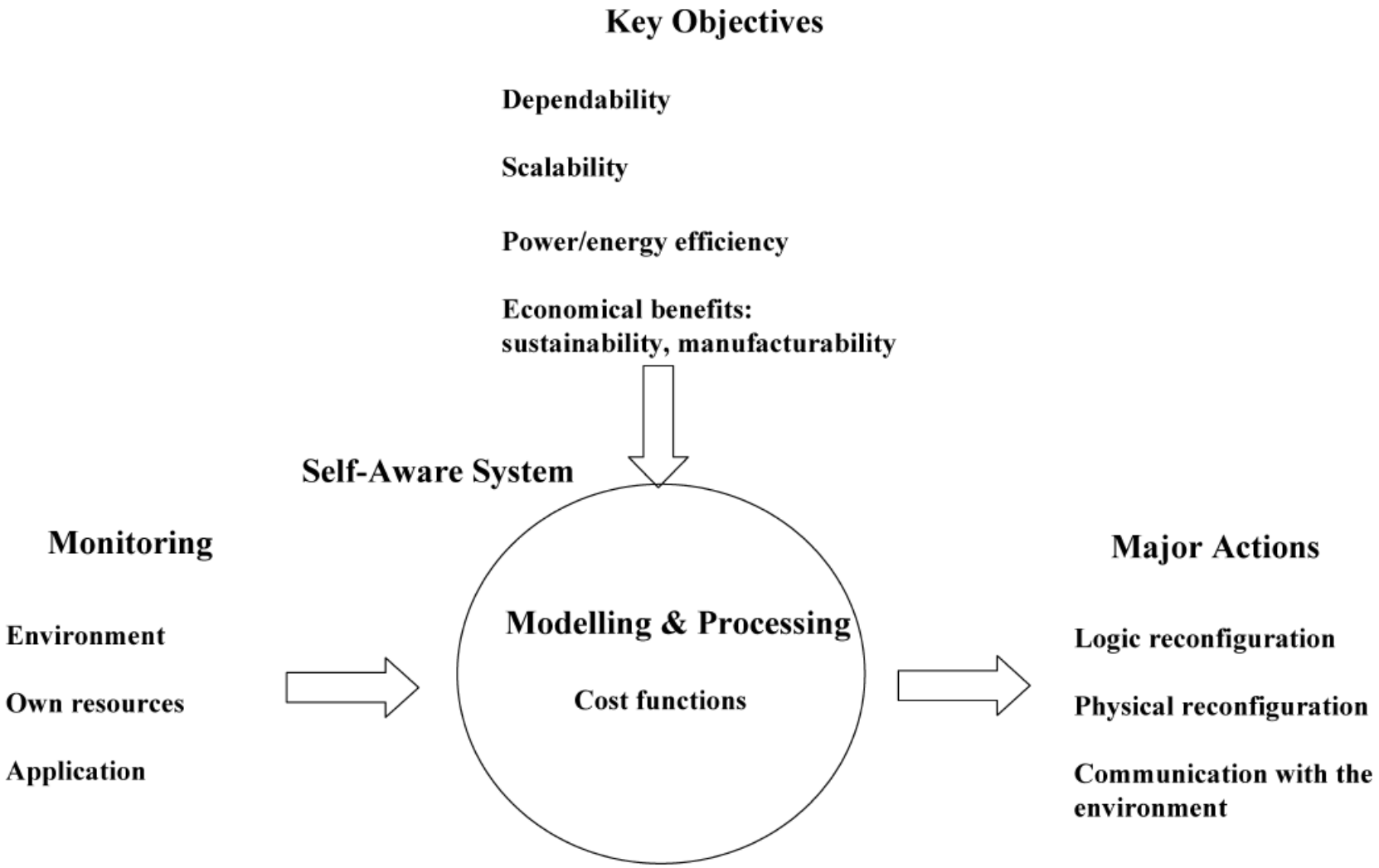
Liang Guang, Ethiopia Nigussie, Pekka Rantala, Jouni Isoaho, and Hannu Tenhunen. “Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip”. In: *ACM Trans. Embed. Comput. Syst.* 9.3 (2010), pp. 1–24

Liang Guang. “Hierarchical Agent-based Adaptation for Self-Aware Embedded Computing Systems”. PhD thesis. Turku, Finland: University of Turku, 2012

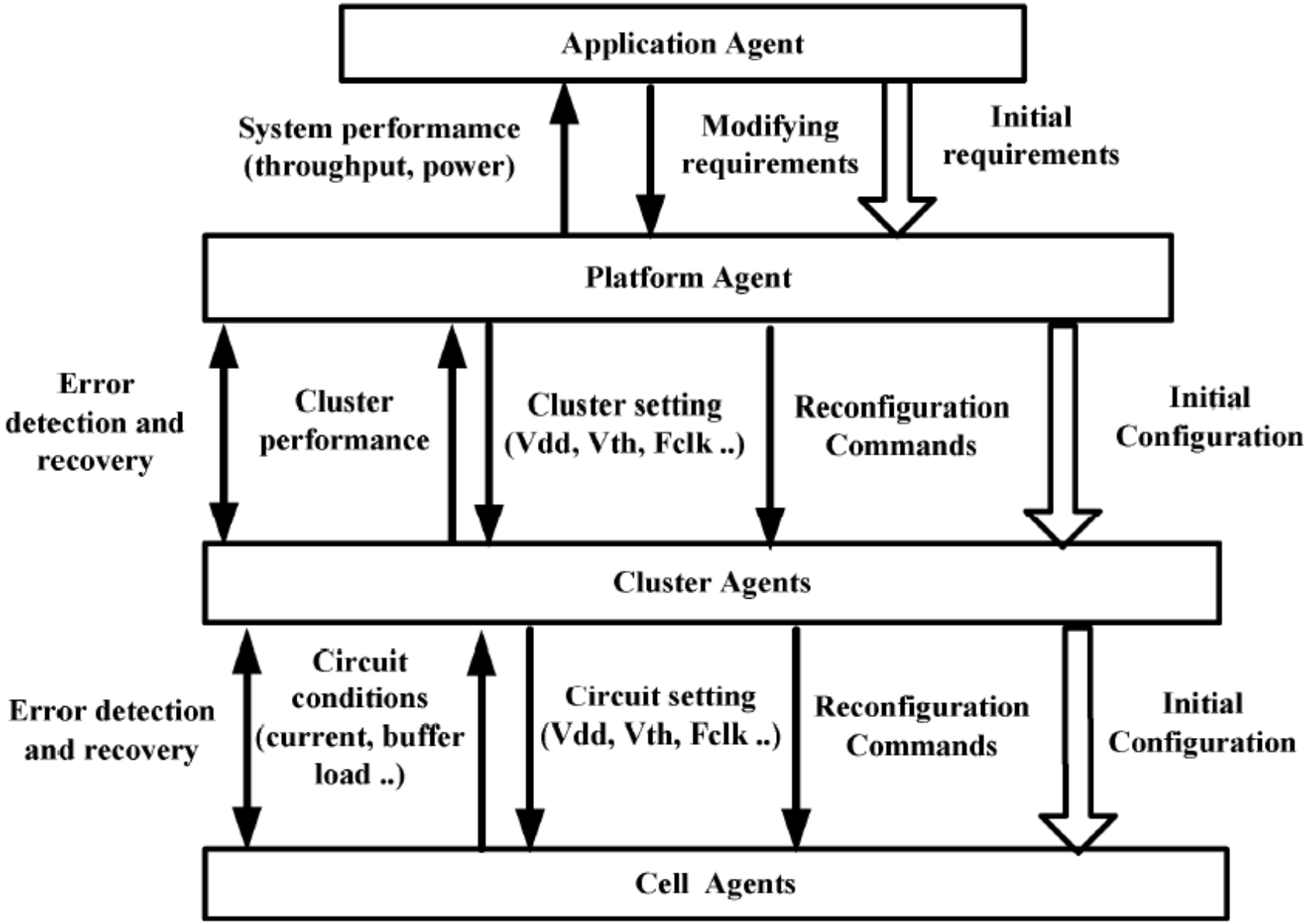
HAMSoC - A Hierarchical Agent Monitored SoC I



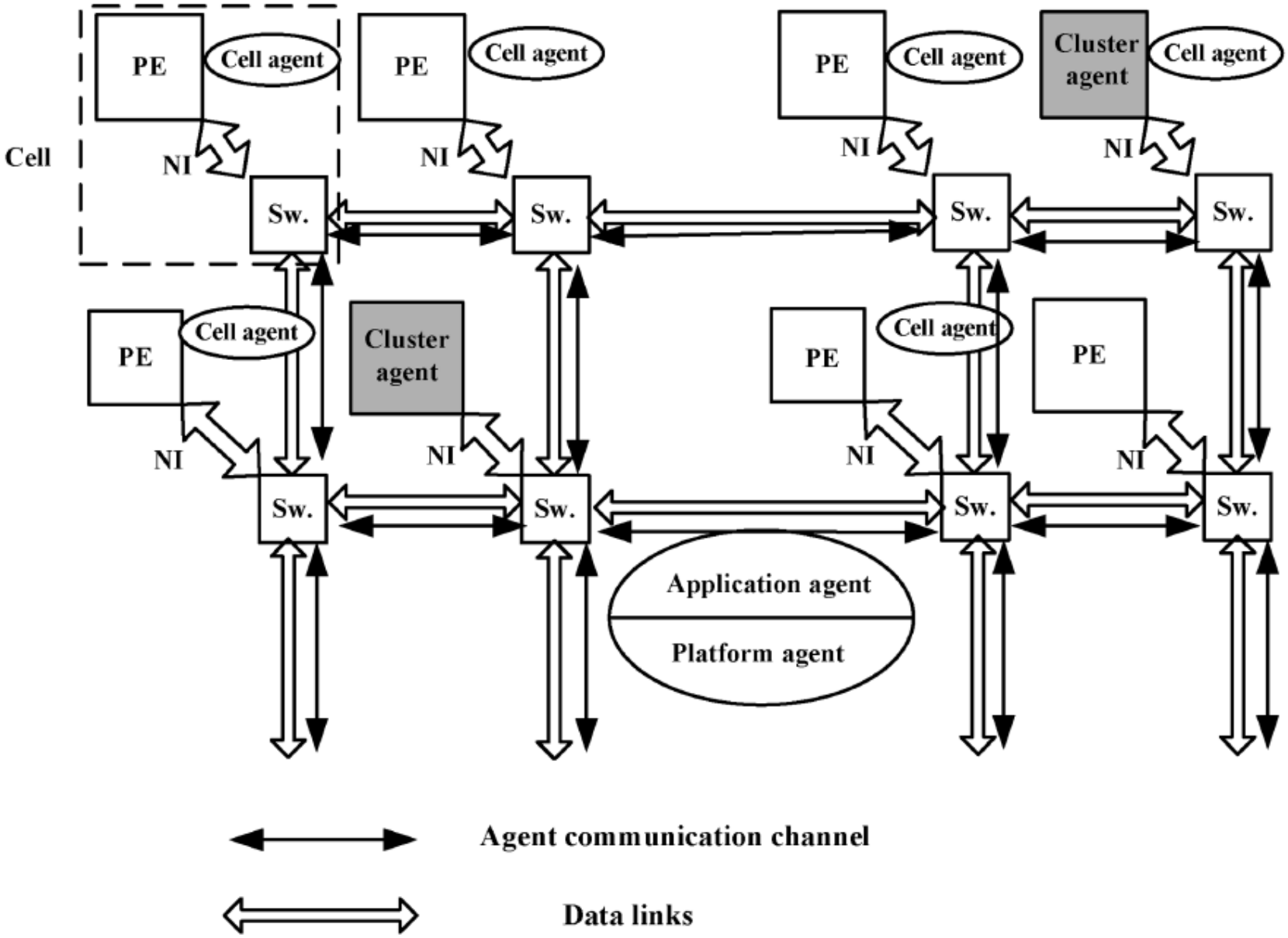
HAMSoC - A Hierarchical Agent Monitored SoC II



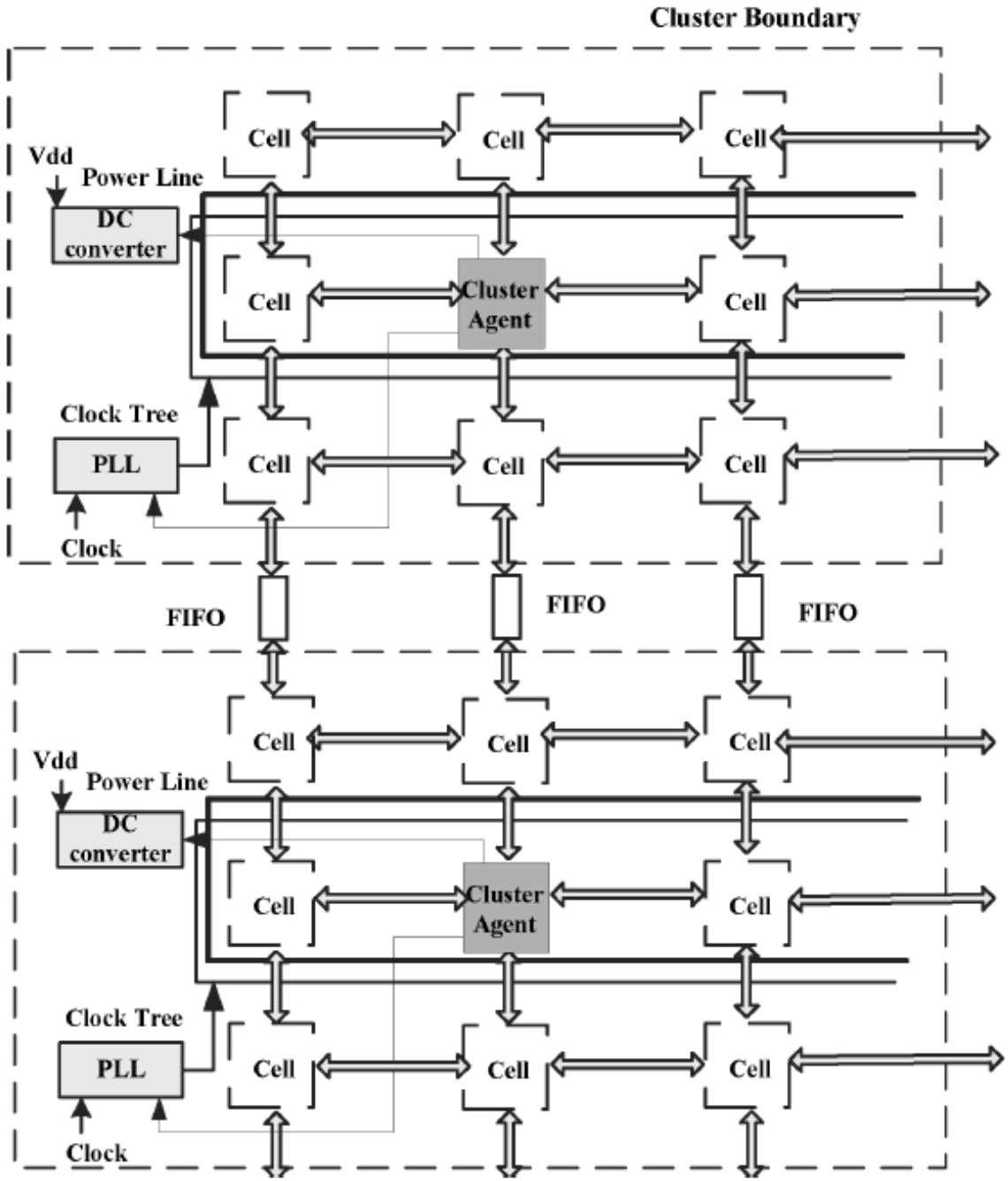
HAMSoC - A Hierarchical Agent Monitored SoC III



HAMSoC - A Hierarchical Agent Monitored SoC IV



HAMSoC - A Hierarchical Agent Monitored SoC V



Normalized Communication Energy of Three Energy-Efficient Architectures

Traffic Pattern	Cluster-based DVFS	Single-domain DVFS	Static Voltage Island
1	80.90%	106.29%	1
2	79.36%	101.98%	1
3	96.21%	100.41%	1
4	90.18%	106.52%	1

Single-domain DVFS: only one DVFS domain, no clusters.
Static voltage island: Lowest statically voltage set for each cluster based on given load.

Normalized Communication Latencies of Three Energy-Efficient Architectures

Traffic Pattern	Cluster-based DVFS	Single-domain DVFS	Static Voltage Island
1	165.34%	131.63%	1
2	144.37%	142.44%	1
3	123.59%	108.44%	1
4	124.00%	121.38%	1

Area Overhead of Three Energy-Efficient Architectures (in mm^2)

Architecture	Links	Switches	DC Regulators & PLLs	Total	% of a Chip Size
Cluster-based DVFS	23.35	12.88	10.63	46.86	17.04%
Single-domain DVFS	23.35	12.88	0.38	36.61	13.31%
Static voltage island	22.63	12.88	0	35.51	12.91%

Summary

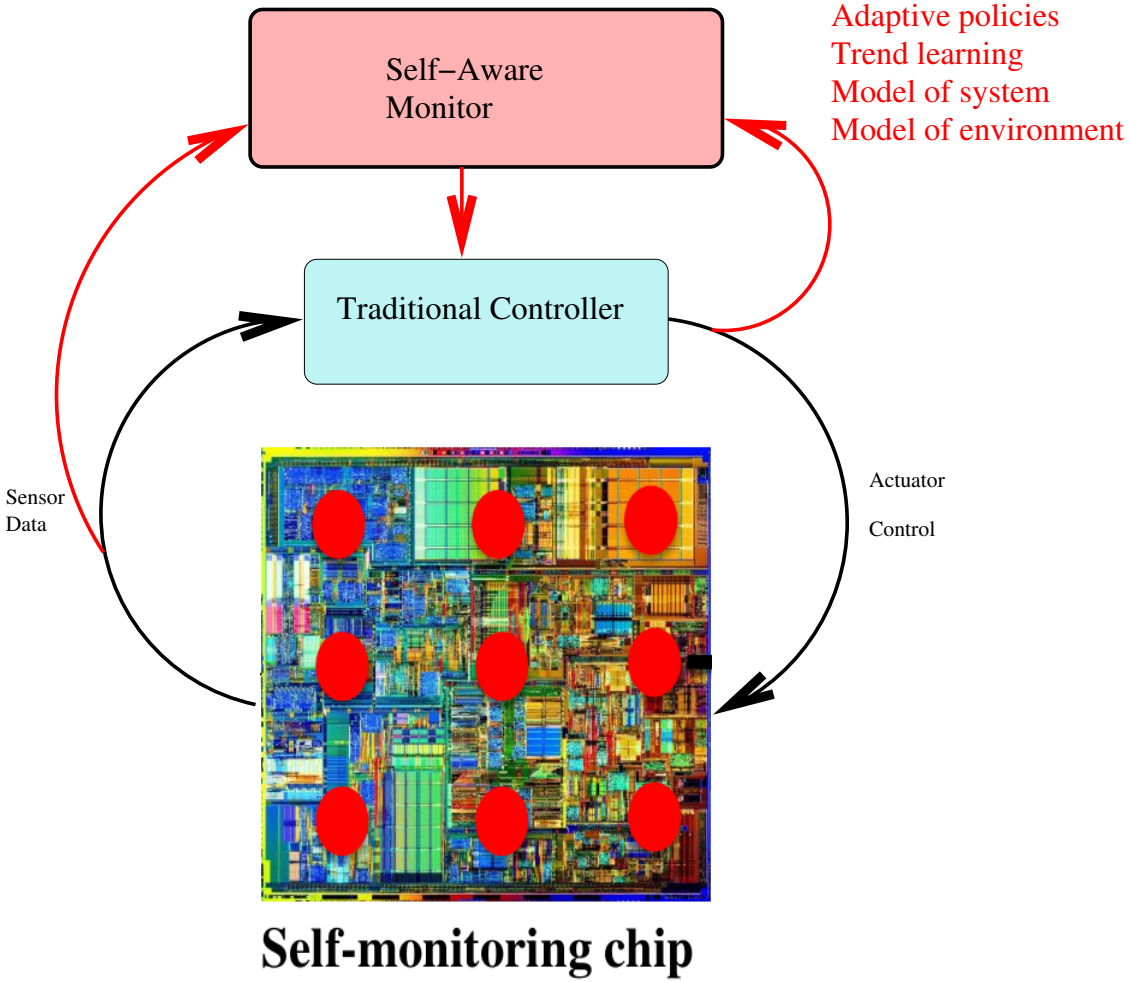
- A hierarchical monitoring and control architecture;
- Applied to
  - adaptive power management, and
  - fault tolerance through reconfiguration.

CPSoC - A Sensor Rich SoC Platform

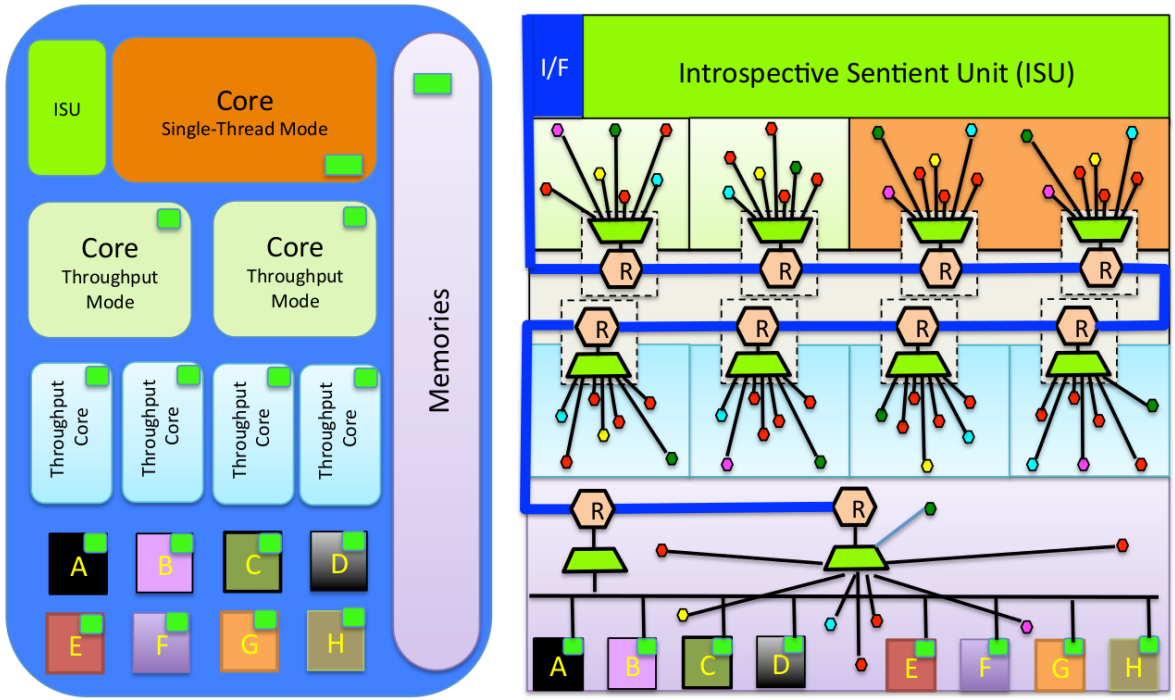
- Sensors and actuators at five layers:
  - Device/ circuit architecture
  - Hardware architecture
  - Network/Bus communication architecture
  - Operating system
  - Application
- Observe-decide-act paradigm
- Codesign of control, communication and computing

Santanu Sarma, Nikil Dutt, N. Venkatasubramaniana, A. Nicolau, and P. Gupta. *CyberPhysical-System-On-Chip (CPSoC): Sensor-Actuator Rich Self-Aware Computational Platform*. Tech. rep. CECS Technical Report No: CECS TR–13–06. Irvine, CA 92697-2620, USA: Center for Embedded Computer Systems University of California, Irvine, May 2013

Cyber-Physical SoC

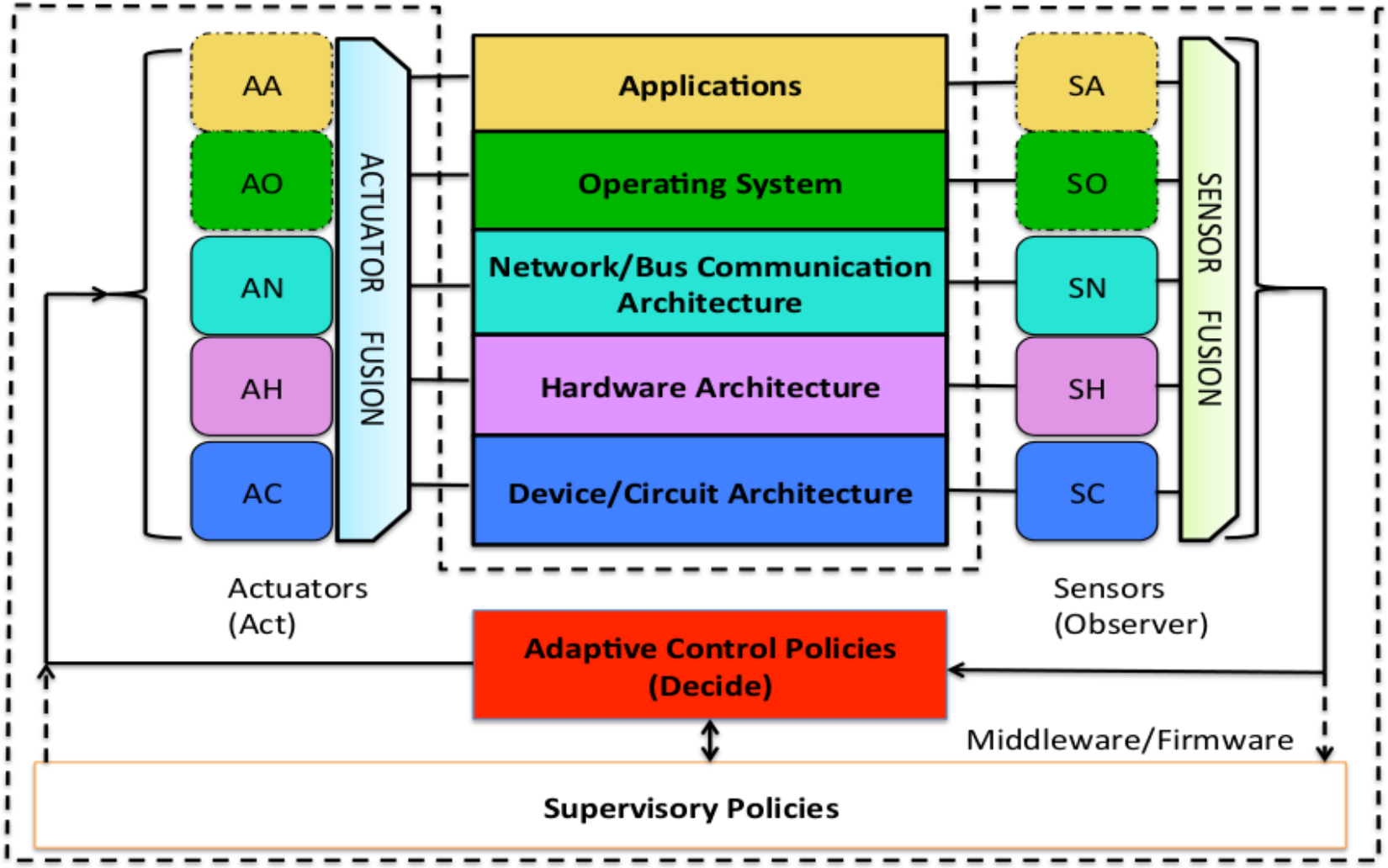


CPSoC - A Sensor Rich SoC Platform



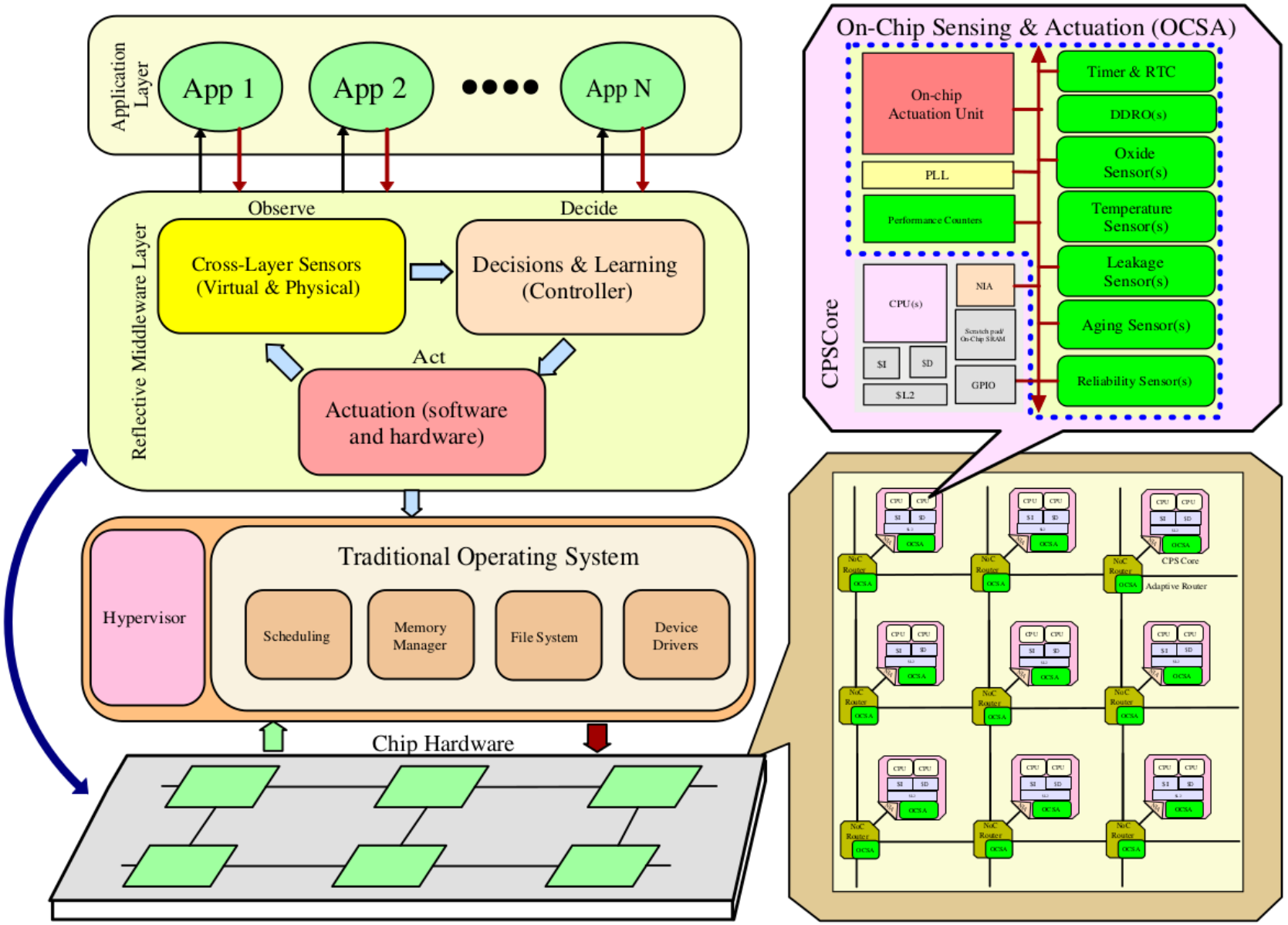
Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. "CyberPhysical-System-On-Chip (CPSoC): A Self-Aware MPSoC Paradigm with Cross-Layer Virtual Sensing and Actuation". In: *Proceedngs of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*. Grenoble, France, Mar. 2015

CPSoC - A Sensor Rich SoC Platform

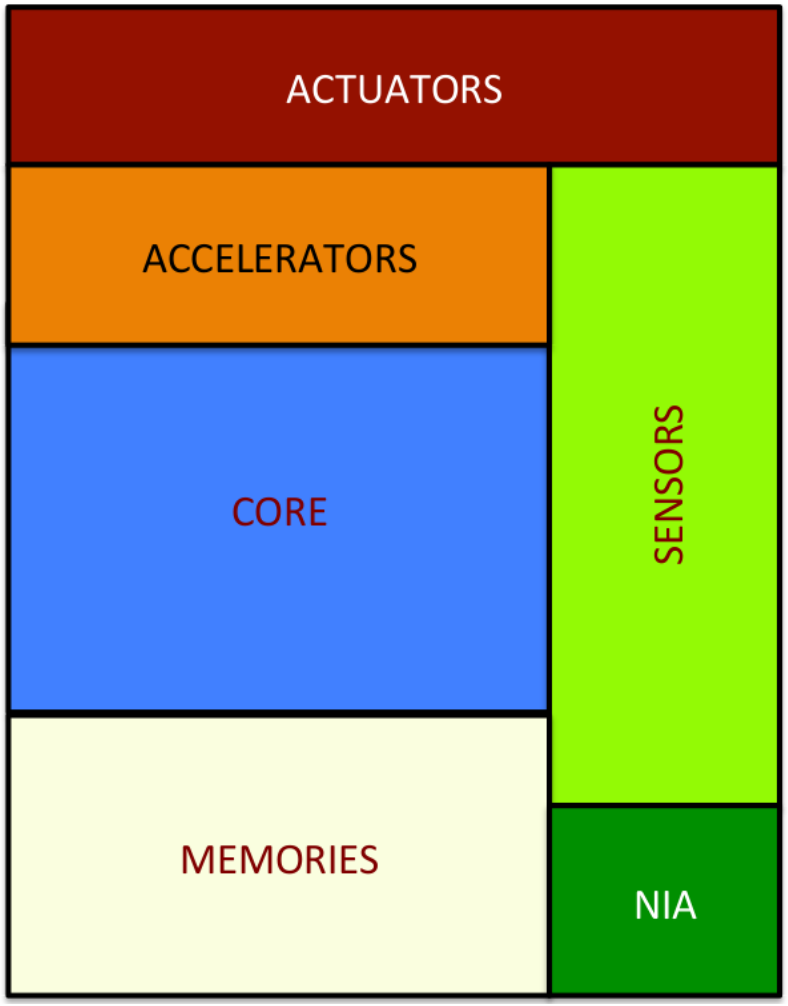


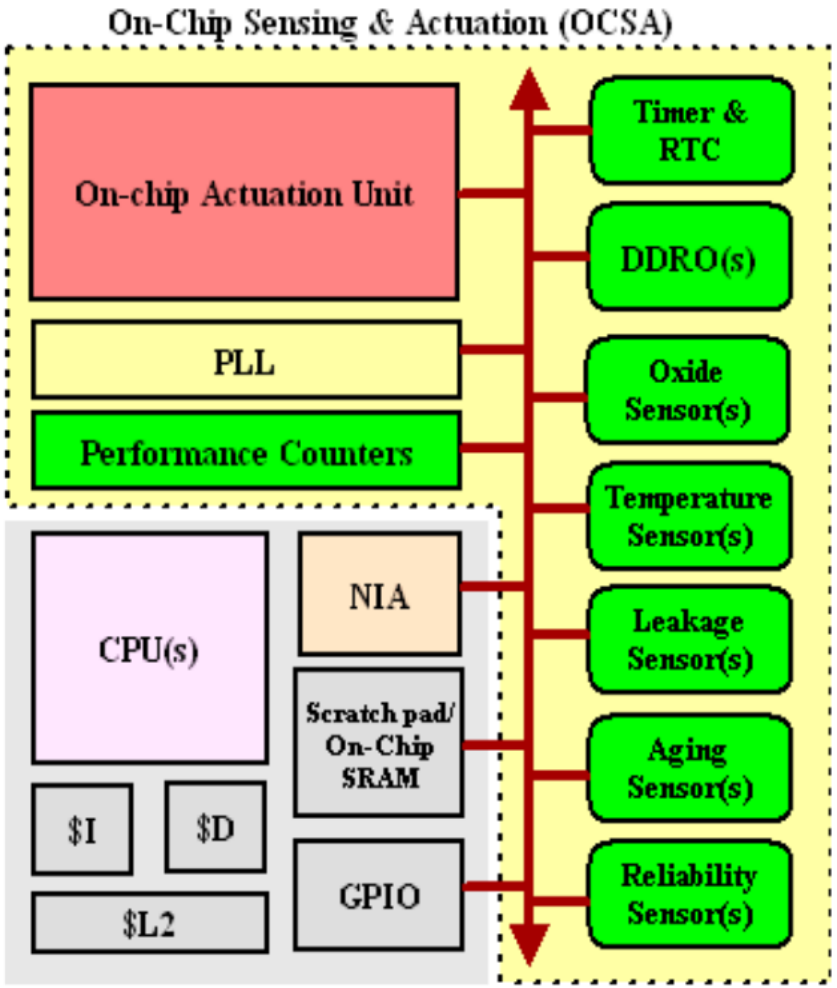
Nikil Dutt, Axel Jantsch, and Santanu Sarma. "Self-Aware Cyber-Physical Systems-on-Chip". In: *Proceedings of the International Conference for Computer Aided Design*. invited. Austin, Texas, USA, Nov. 2015

CPSoC - A Sensor Rich SoC Platform



CPSoC - A Sensor Rich SoC Platform



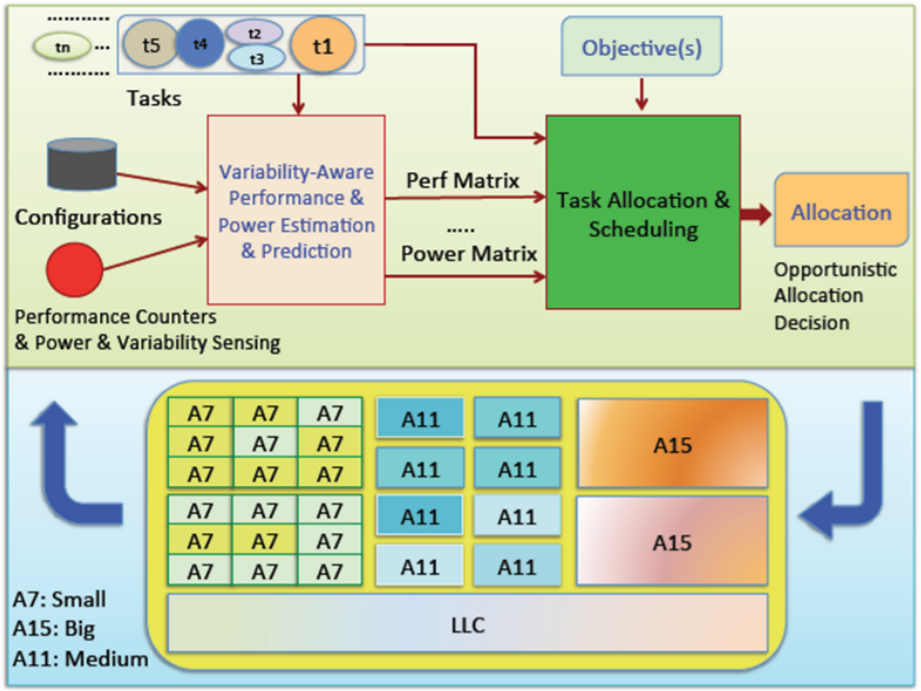


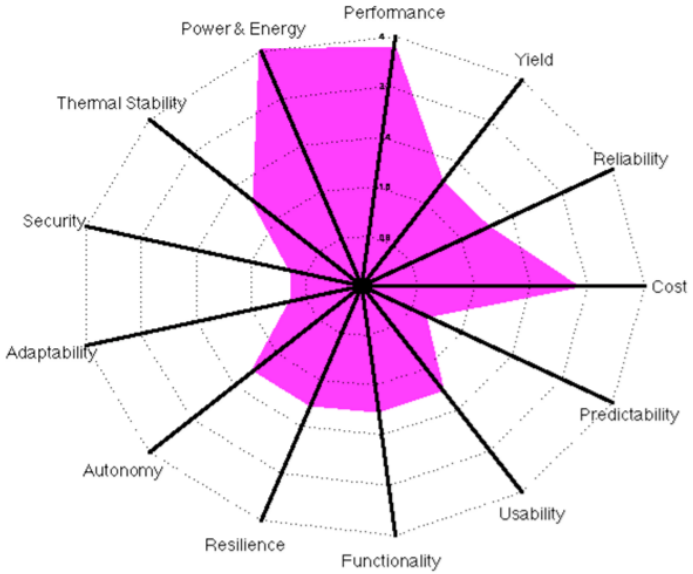
Sensing and Actuating at All Layers

Layers	Virtual/Physical Sensors	Virtual/Physical Actuators
Application	Workload, Power, Energy, Execution Time	Approximation, Algorithmic choice, Transformations
Operating System	System utilization, Peripheral states	Task allocation, Partitioning, Scheduling, Migration, Duty cycle
Network/Bus	Bandwidth, Packet/flit status, Channel status, Congestion	Adaptive routing, Dynamic BW allocation, Channel allocation, Flow control
Hardware Architecture	Cache miss rate, Access rate, IPC, Throughput, Resource utilization	Cache sizing, Issue width sizing, Reconfiguration, Resource provisioning
Circuit/Device	Circuit delay, Aging effects, Leakage, Temperature, Device faults	DVFS, Clock gating, Power gating

Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. "CyberPhysical-System-On-Chip (CPSoC): A Self-Aware MPSoC Paradigm with Cross-Layer Virtual Sensing and Actuation". In: *Proceedngs of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*. Grenoble, France, Mar. 2015

Improvement of Energy Efficiency



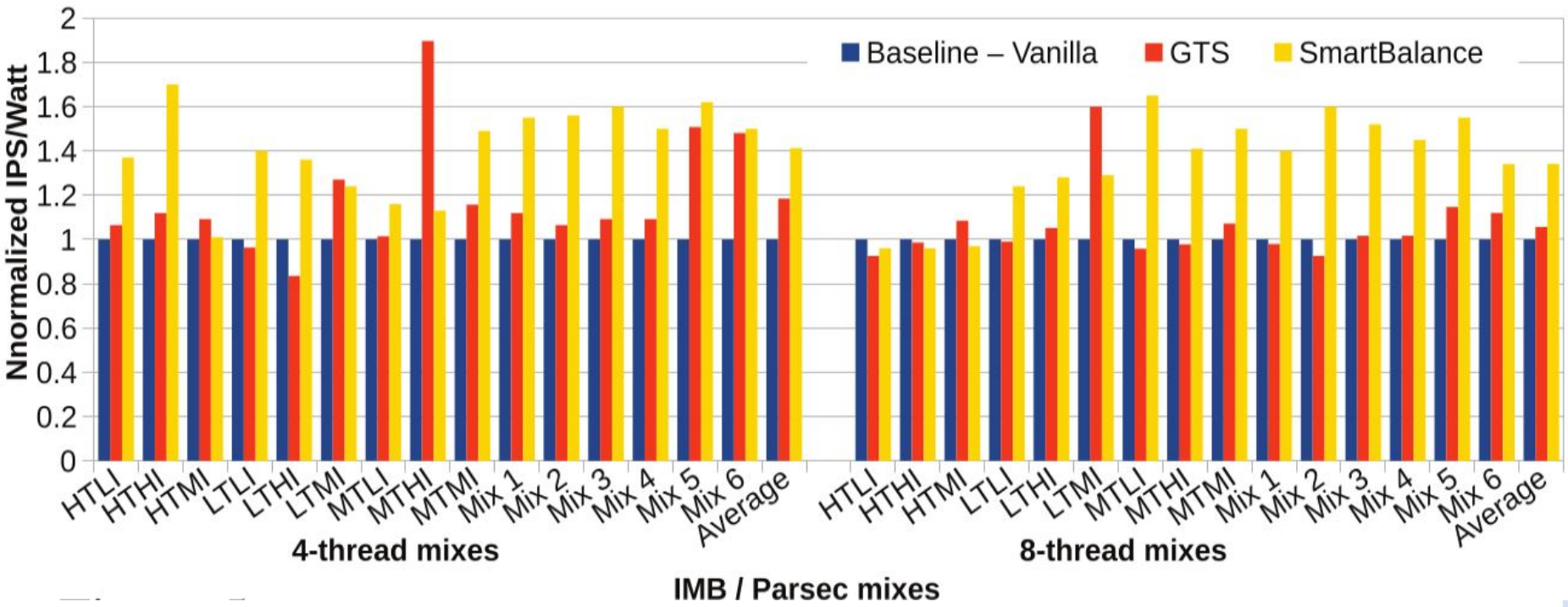


Goal:

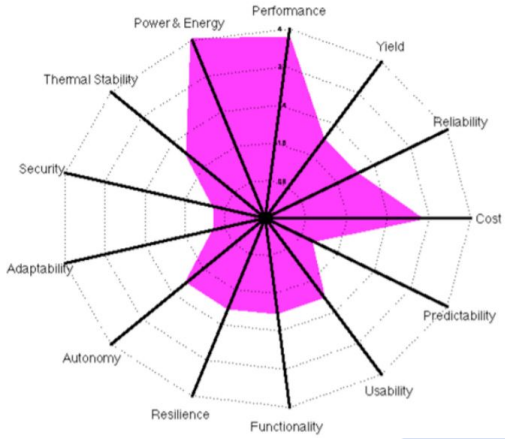
- **Energy Efficiency**

Santanu Sarma and Nikil Dutt. "Cross-Layer Exploration of Heterogeneous Multicore Processor Configurations". In: *VLSI Design Conference*. 2015

Improvement of Energy Efficiency

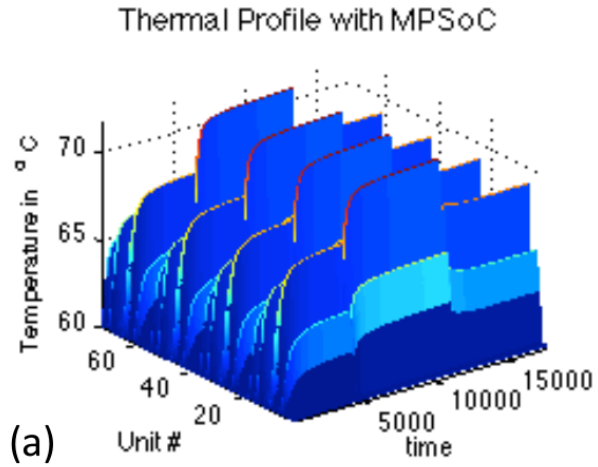


The benefit comes from actually measuring energy efficiency.

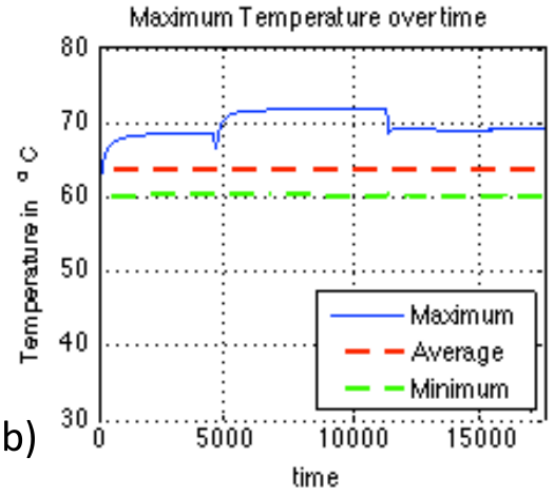


Santanu Sarma, T. Muck, L. A.D. Bathen, N. Dutt, and A. Nicolau. "SmartBalance: A Sensing-Driven Linux Load Balancer for Energy Efficiency of Heterogeneous MPSoCs". In: *Proceedings of the Design Automation Conference*. July 2015

Thermal-Aware Performance

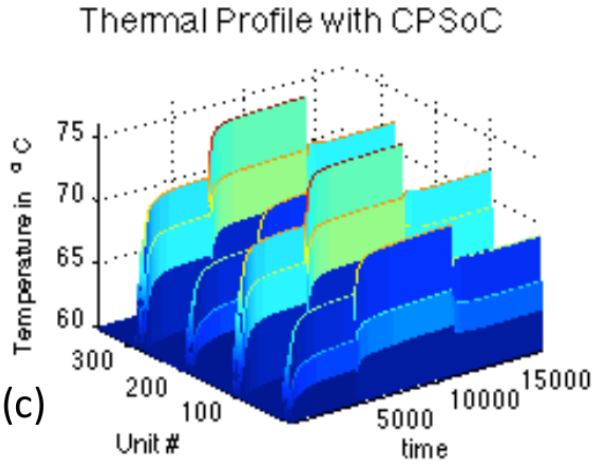


(a)

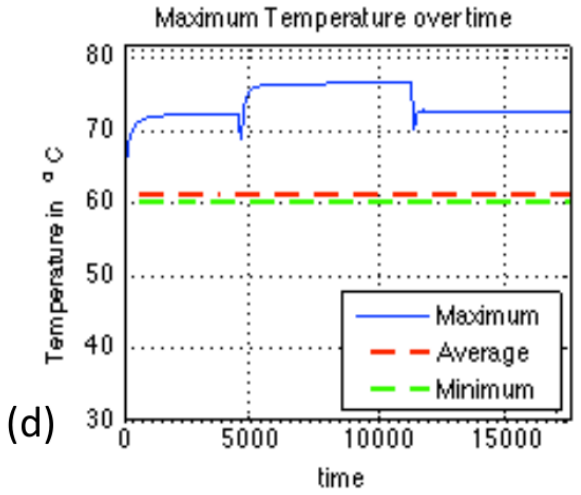


(b)

Throughput improvement by 70%-300% for same power and temperature.



(c)

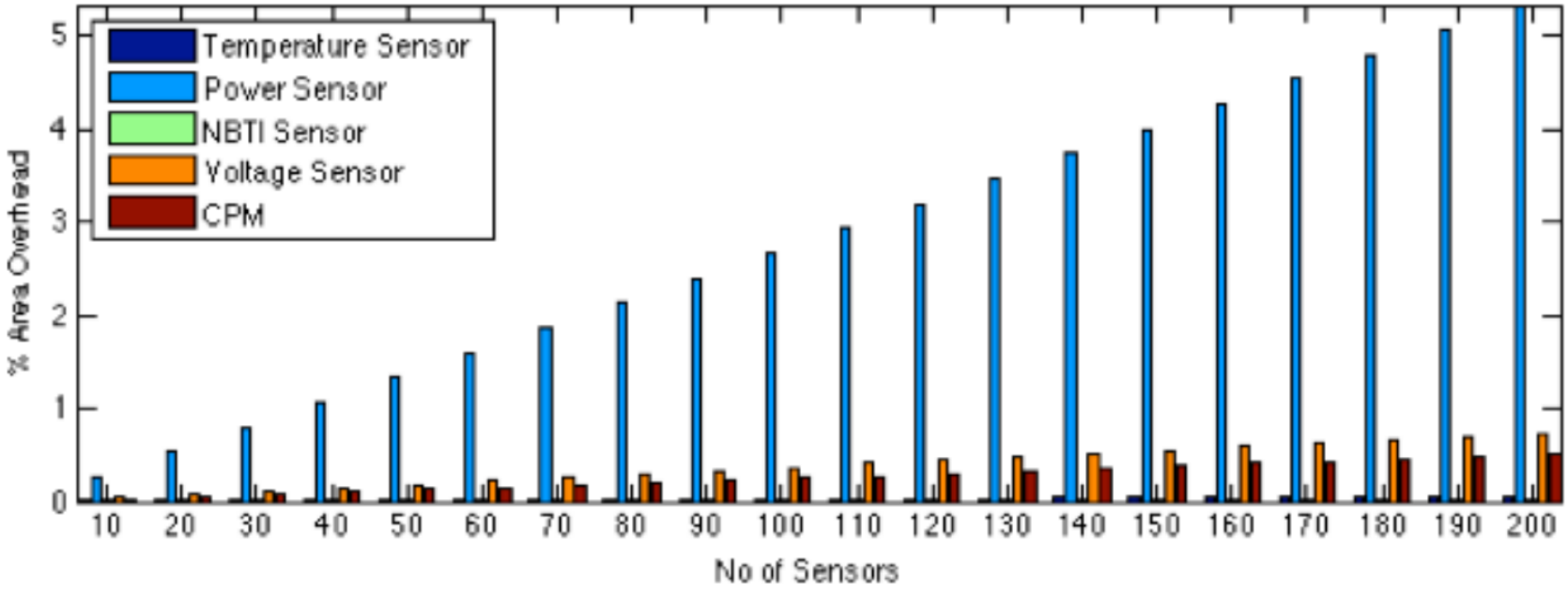


(d)

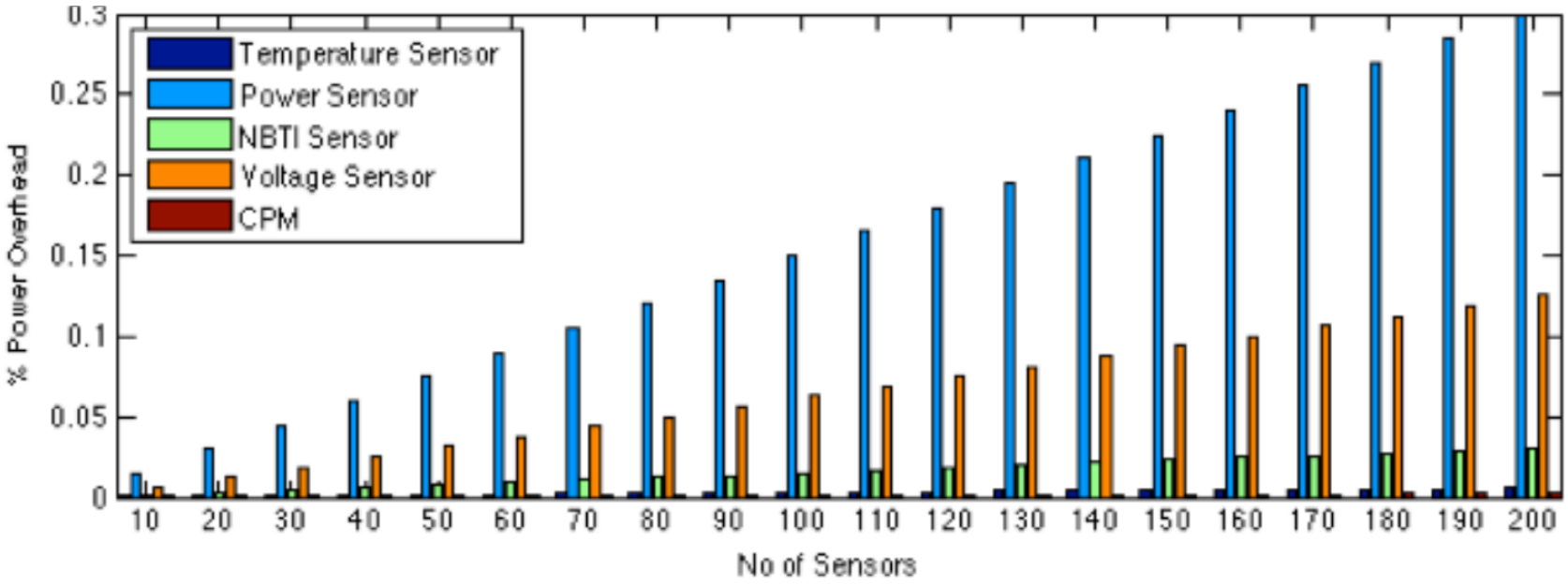
Benefit is due to accurate and fine-grain measurement and tight tracking.

Santanu Sarma, Nikil Dutt, N. Venkatasubramaniana, A. Nicolau, and P. Gupta. *CyberPhysical-System-On-Chip (CPSoC): Sensor-Actuator Rich Self-Aware Computational Platform*. Tech. rep. CECS Technical Report No: CECS TR–13–06. Irvine, CA 92697-2620, USA: Center for Embedded Computer Systems University of California, Irvine, May 2013

CPSoC - A Sensor Rich SoC Platform



Virtual sensing reduces the area overhead for 1000 sensors from 7.3% to 0.6%.



Virtual sensing reduces the power overhead for 1000 sensors from 1.7% to 0.3%.

Summary - Self-Awareness Features

Semantic interpretation	Implicit
Desirability scale	Explicit
Semantic interpretation	Implicit
History	Rudimentary and implicit
Goals	Explicit, hard coded
Expectations on environment	Rudimentary, implicit and hard coded
Expectations on itself	Rudimentary, implicit and hard coded
Self Inspection	Rudimentary

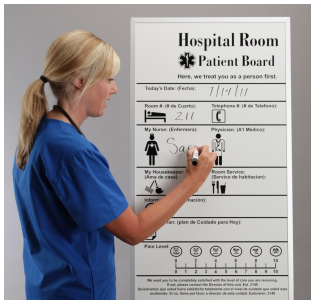
Summary

- Layered sensor, actuator and control architecture;
- Fine grained sensing can have significant benefits;
- Virtual sensing can significantly reduce overhead;
- Rudimentary self-awareness with many SA features being implicit.

Early Warning Score

Score	3	2	1	0	1	2	3
Heart rate[1]	<40	40–51	51–60	60–100	100–110	110–129	>129
Systolic BP[2]	<70	70–81	81–101	101–149	149–169	169–179	>179
Breath rate[3]		<9		9–14	14–20	20–29	>29
SPO_2 (%)	<85	85–90	90–95	>95			
Body temp.[4]	<28	28–32	32–35	35–38		38–39.5	>39.5

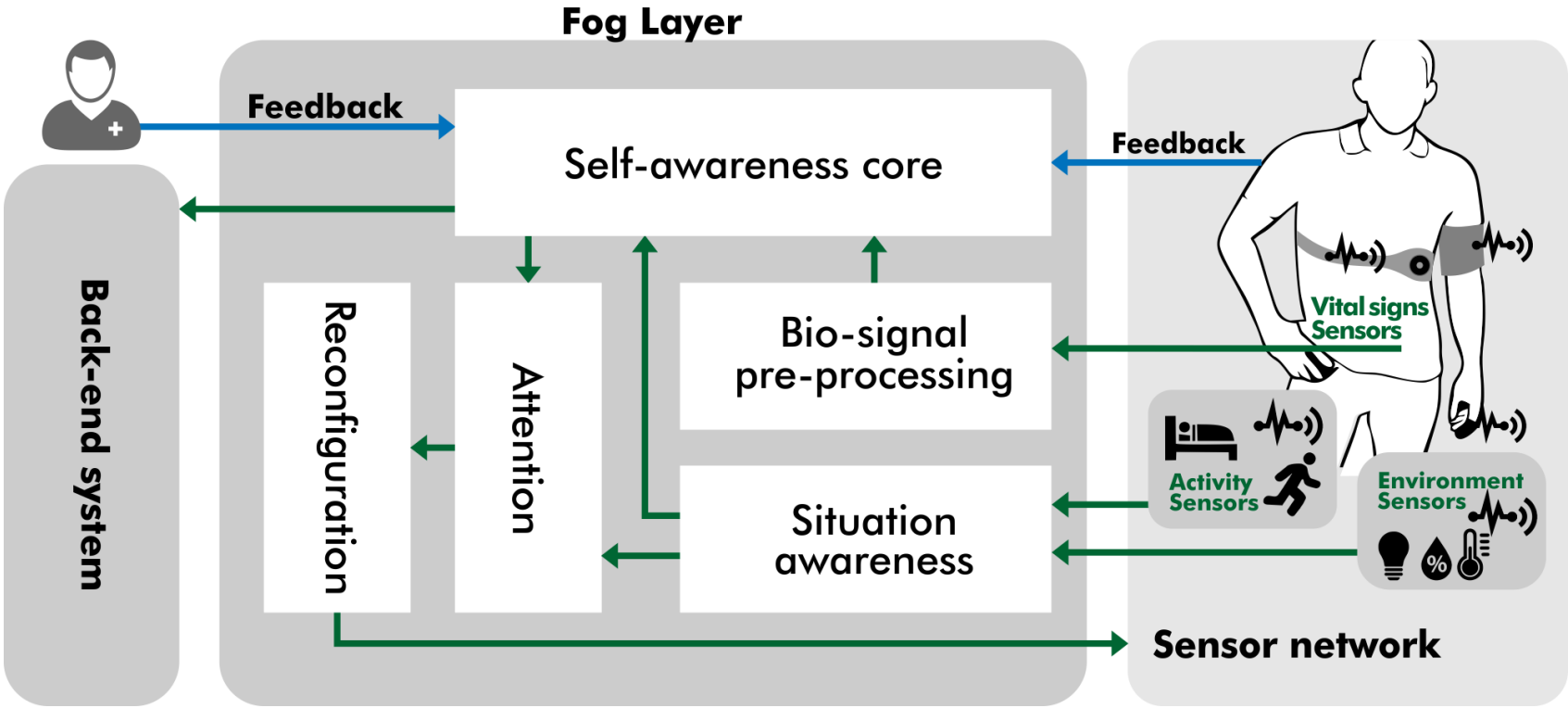
[1]beats per minute, [2]mmHg, [3]breaths per minute, [4] °C



EWS Improvement

- Data reliability:
  - Values in reasonable scope
  - Changes in reasonable scope
  - Consistency between sensors
- Situation awareness
- Power efficiency

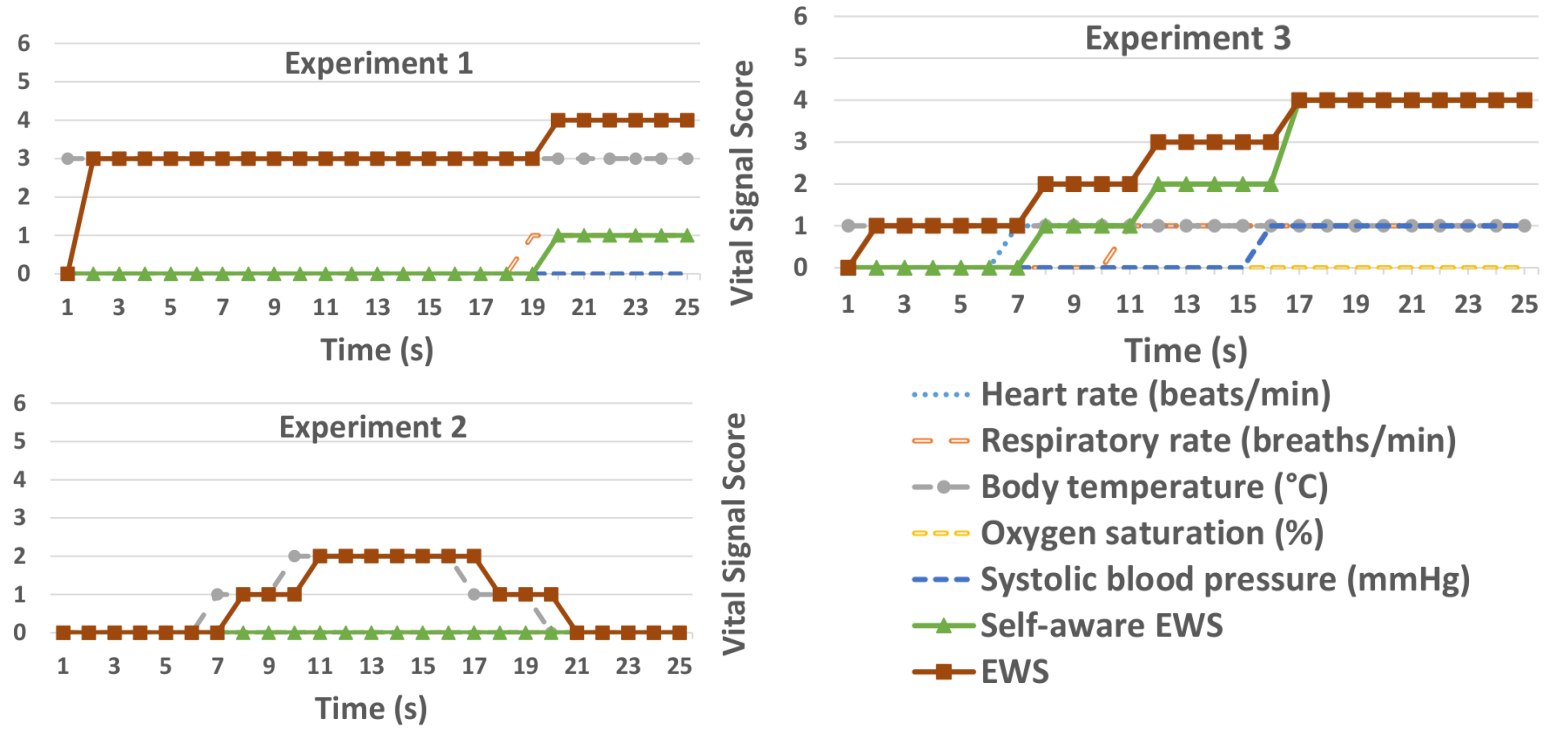
Enhanced Early Warning Score



Arman Anzanpour et al. “Self-Awareness in Remote Health Monitoring Systems using Wearable Electronics”. In: *Proceedings of Design and Test Europe Conference (DATE)*. Lausanne, Switzerland, Mar. 2017

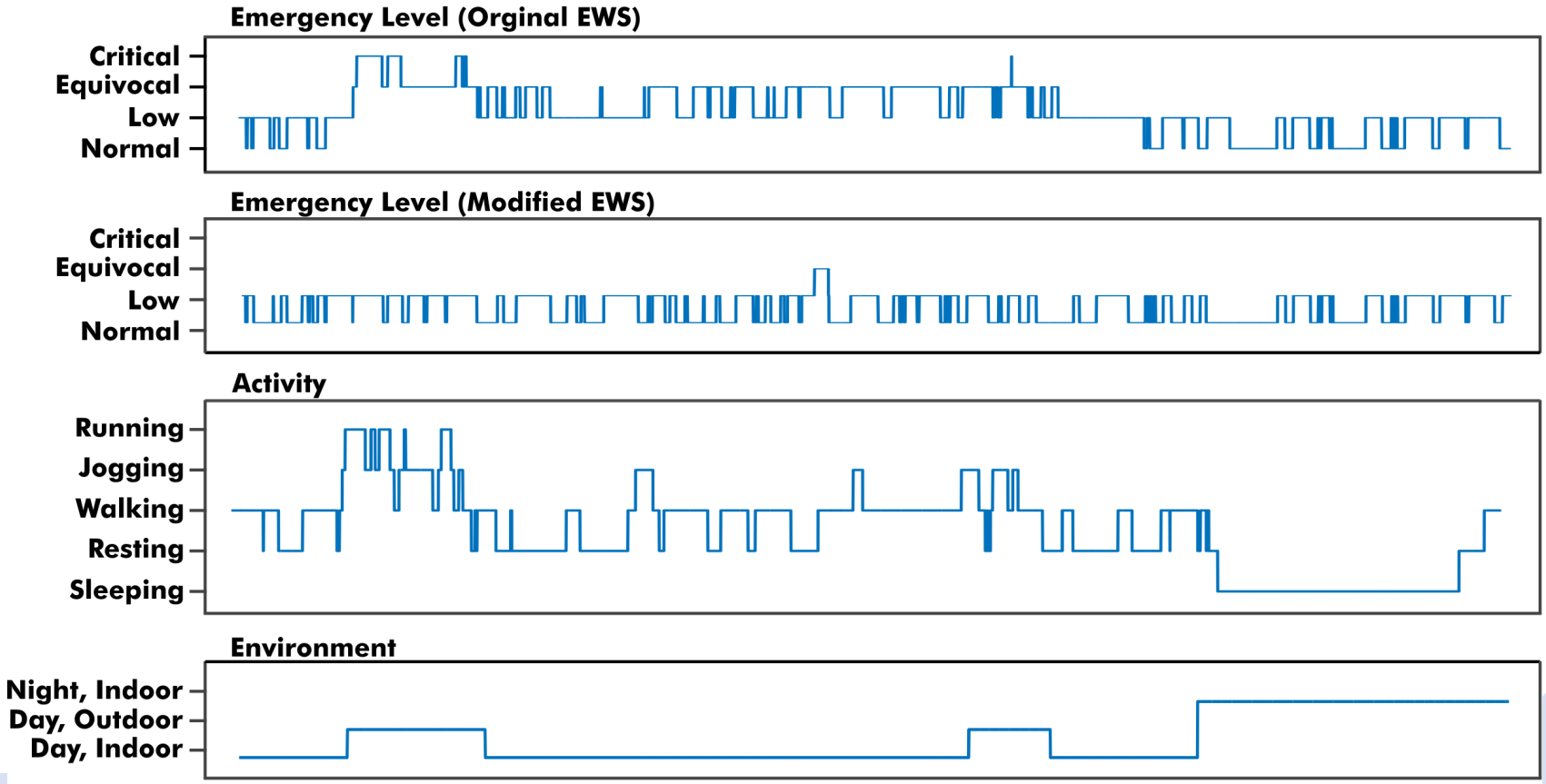
Enhanced Early Warning Score - Data Reliability

1. Check on the reliability of sensed values
2. Check on the reliability of value changes
3. Check on consistency between sensor data



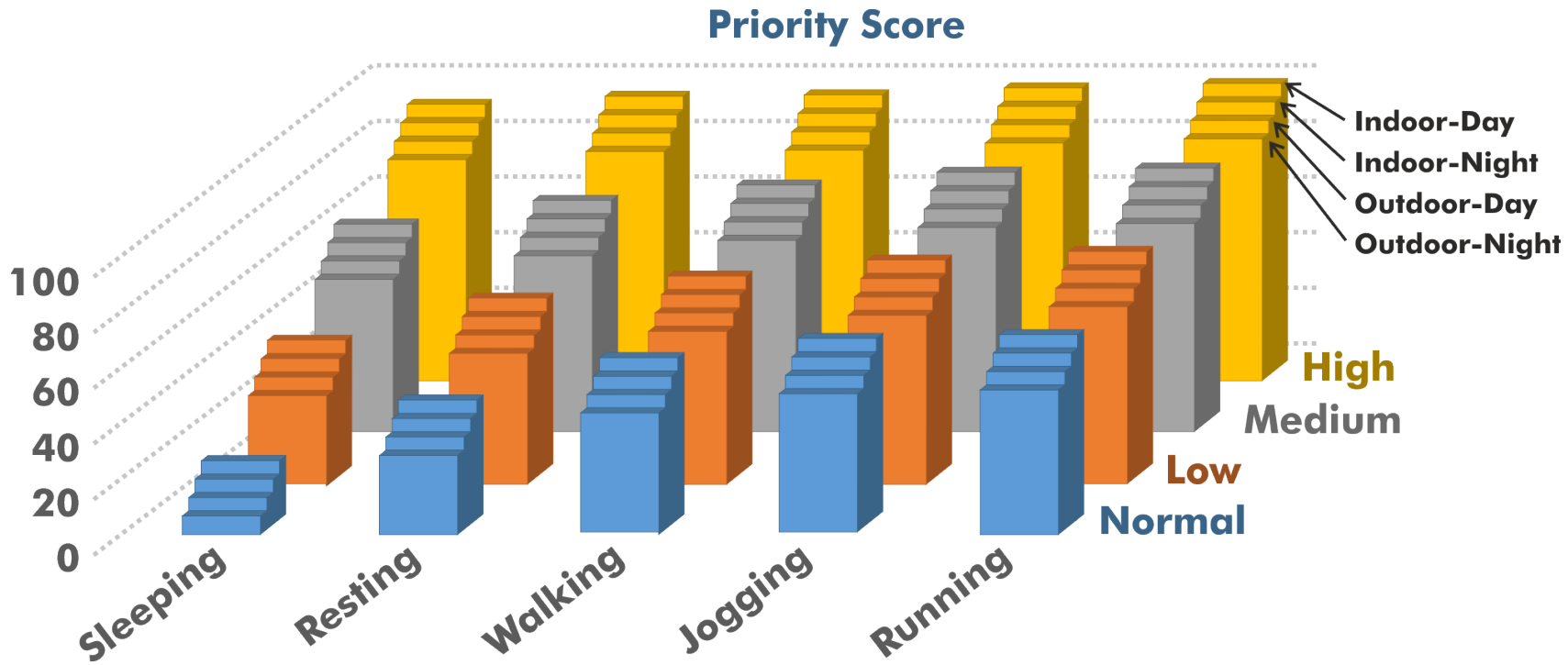
Enhanced Early Warning Score - Situation Awareness

1. Consider the activity mode of person
2. Consider time of day
3. Consider location



Enhanced Early Warning Score - Power Efficiency

1. Prioritize different situations



Enhanced Early Warning Score - Power Efficiency

1. Prioritize different situations
2. Distinguish different modes of urgency

Emergency Level:	Score:0 Normal				Score:1-3 Low				Score:4-6 Medium				Score>6 High			
	Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor	
	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night
Sleeping	E	E	E	E	C	D	D	D	B	C	C	C	A	A	B	B
Resting	D	D	D	D	C	C	C	C	B	B	B	B	A	A	B	B
Walking	C	C	C	C	B	C	C	C	B	B	B	B	A	A	A	B
Jogging	C	C	C	C	B	B	B	C	B	B	B	B	A	A	A	B
Running	C	C	C	C	B	B	B	B	B	B	B	B	A	A	A	A

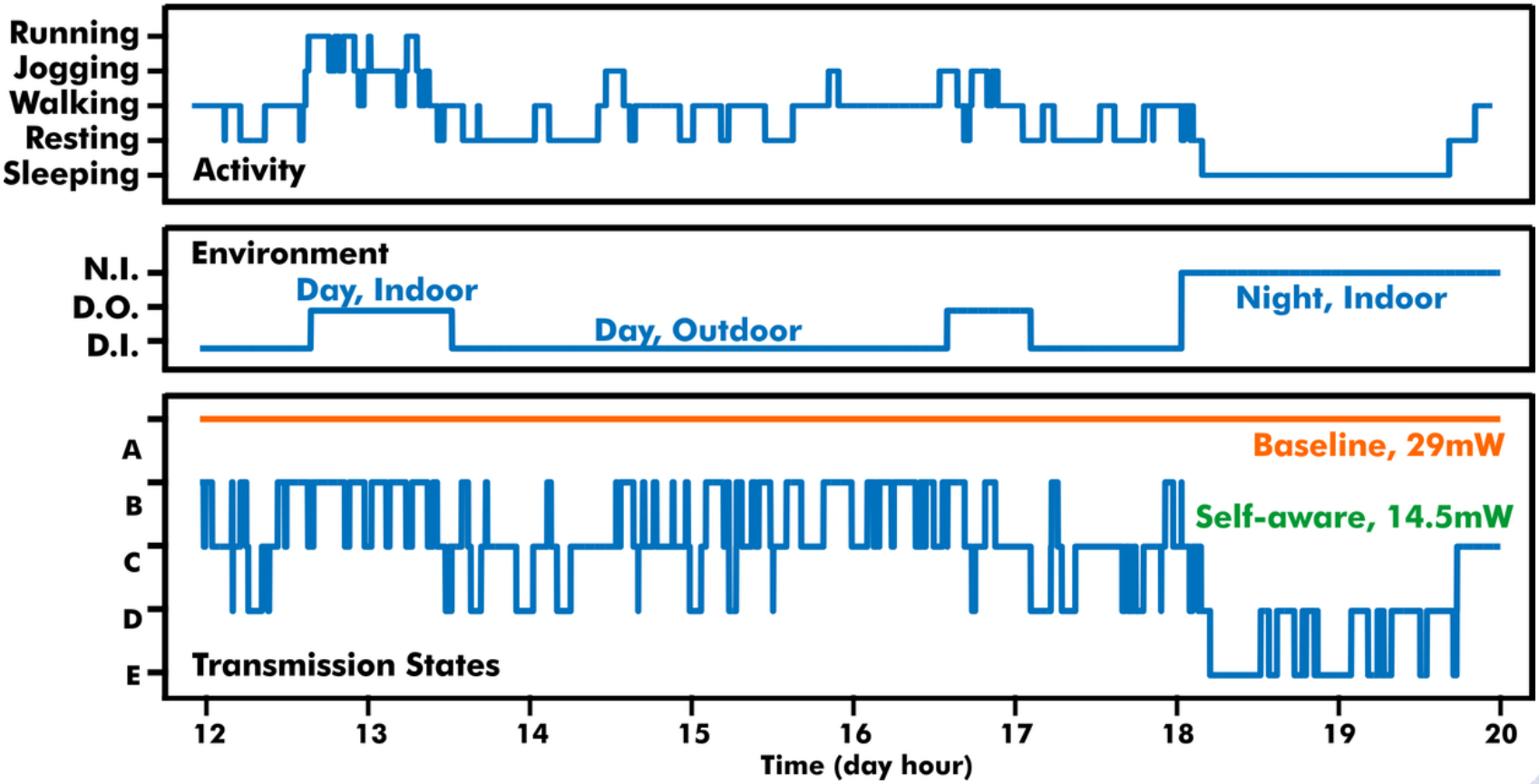
Enhanced Early Warning Score - Power Efficiency

1. Prioritize different situations
2. Distinguish different modes of urgency
3. Define sensing activity for each mode

State	Respiration Rate Activity	Blood Pressure	Heart Rate, SpO2, and Body Temp.	Transmission Power Consumption
A	Continuous	Every hour in day Disabled in night	Every sec.	29 mW
B	2 min continuous 8 min OFF	Every hour in day Disabled in night	Every sec.	26.8 mW
C	2 min continuous 3 min OFF	Every 3 hours in day Disabled in night	Every min.	12.5 mW
D	2 min continuous 8 min OFF	Every 3 hours in day Disabled in night	Every min.	7 mW
E	2 min continuous 18 min OFF	Disabled	Every min.	4.3 mW

Enhanced Early Warning Score - Power Efficiency

Over a day half the energy can be saved.



Enhanced Early Warning Score

Summary

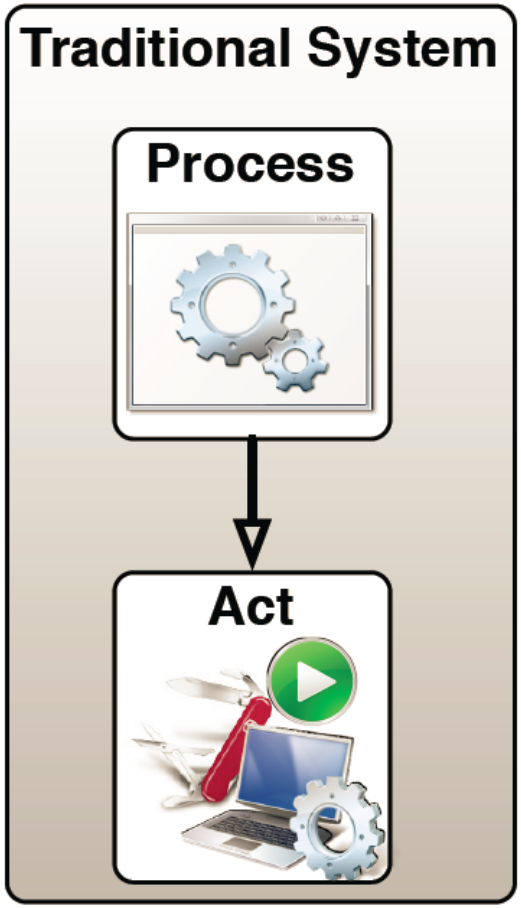
- Considering data reliability improves quality of observation;
- Considering sitation improves quality of observation;
- Collecting needed data only improves efficiency.

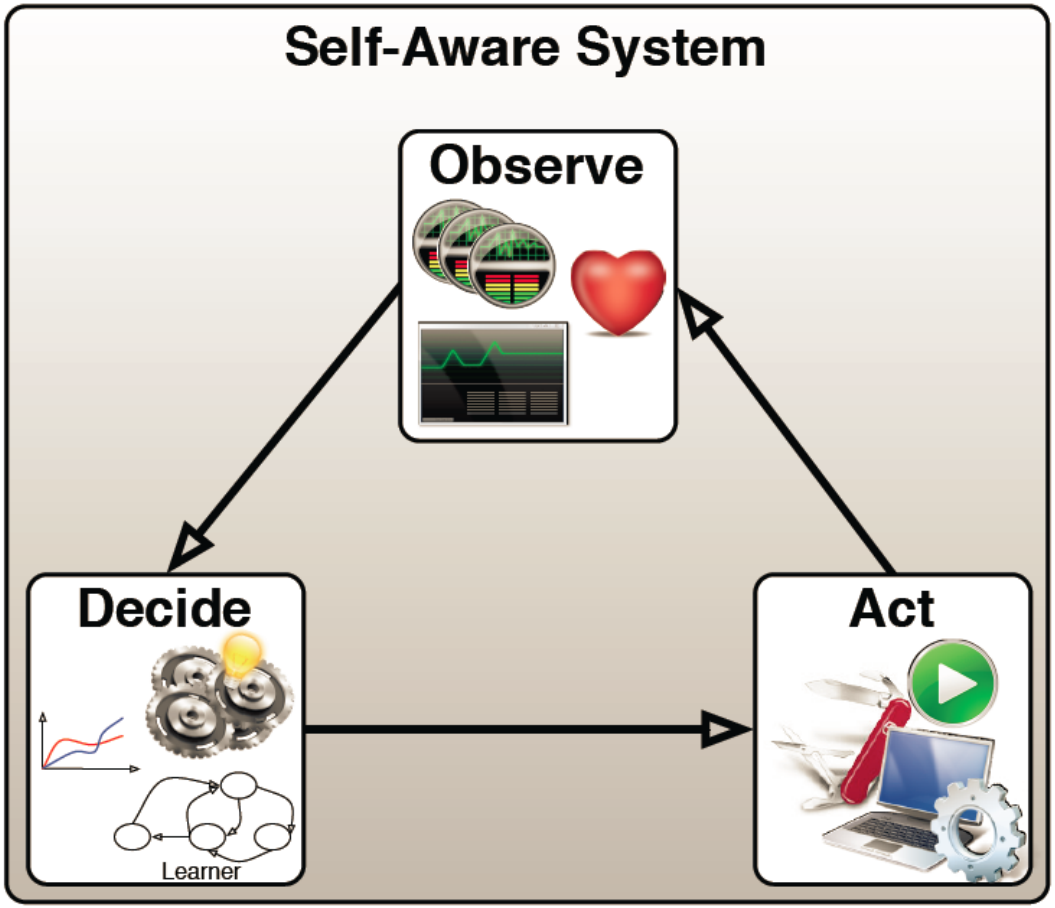
SEEC - A Framework for Self-Aware Computing

- The applications specify goals
- The platform provides possible actions
- SEEC monitors the application and decides upon actions
- Observe - Decide - Act based control loop

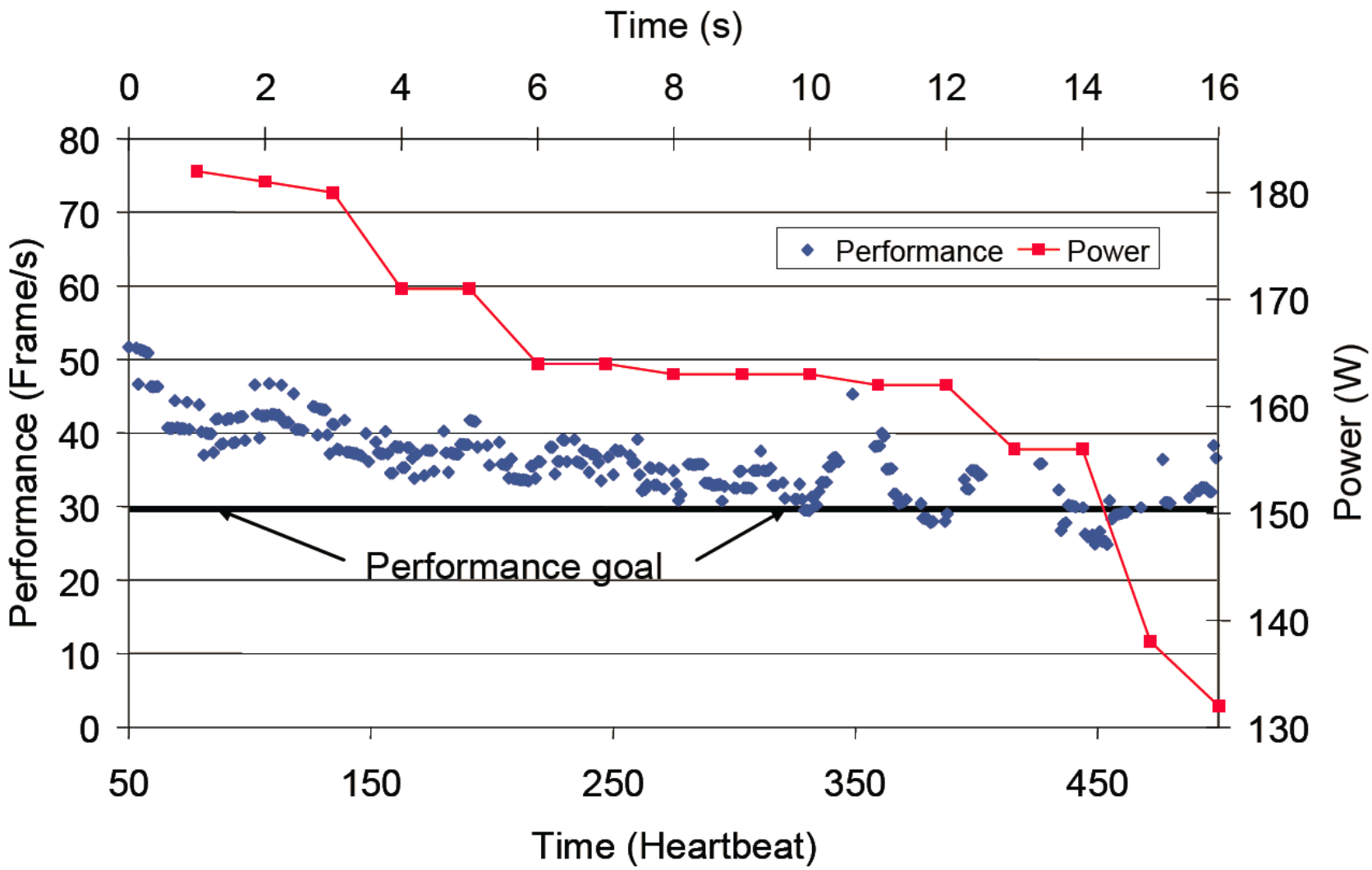
Henry Hoffmann, Martina Maggio, Marco D Santambrogio, Alberto Leva, and Anant Agarwal. *Seec: A framework for self-aware computing*. Tech. rep. MIT-CSAIL-TR-2010-049. Cambrige, Massachusetts: MIT, Oct. 2010

SEEC - A Framework for Self-Aware Computing





SEEC - A Framework for Self-Aware Computing



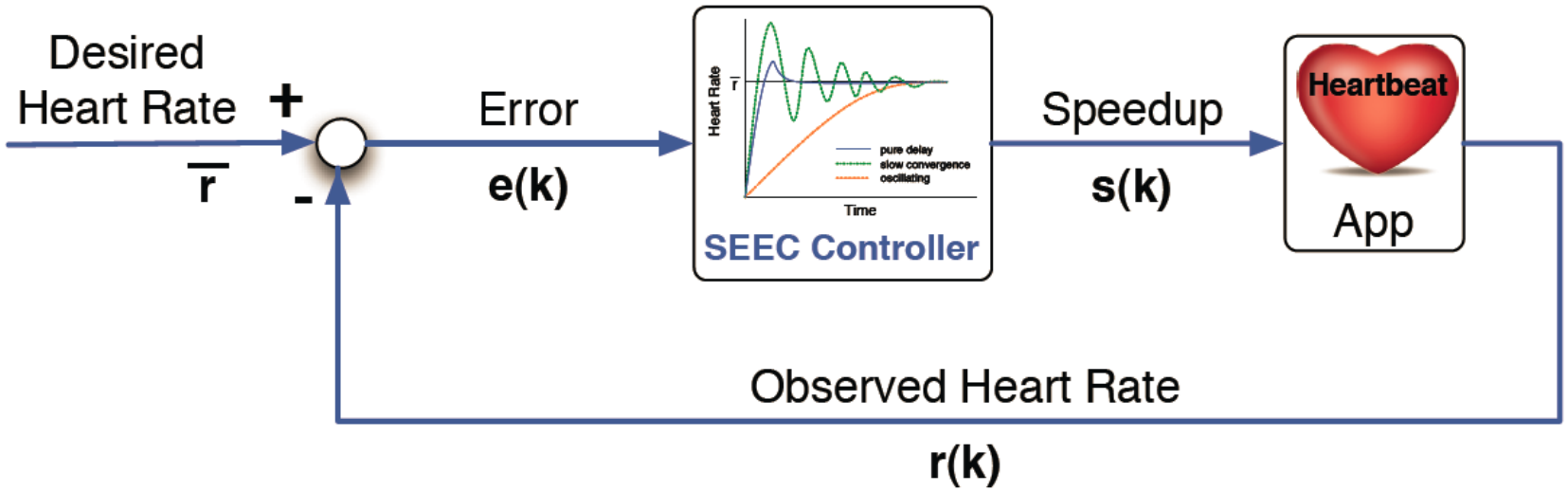
x264 encoder with 30 frames/sec performance goal.

SEEC - A Framework for Self-Aware Computing

Phase	Applications Developer	Systems Developer	SEEC Framework
Observation	Specify application goals and performance	-	Read goals and performance
Decision	-	-	Determine how much to speed up the application
Action	-	Specify actions and a function that performs actions	Initiate actions based on result of decision phase

Roles in the SEEC development framework.

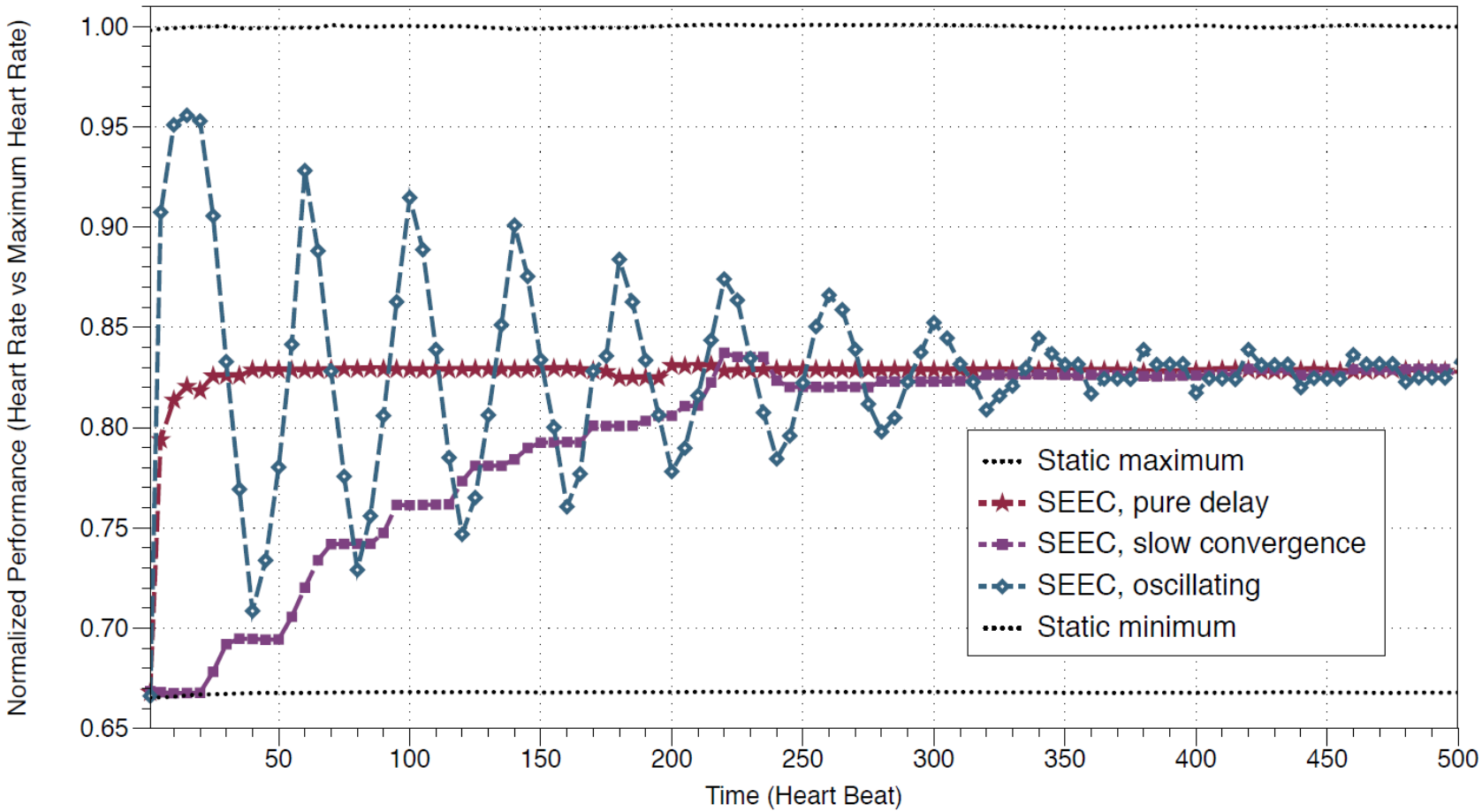
SEEC - A Framework for Self-Aware Computing



SEEC - A Framework for Self-Aware Computing

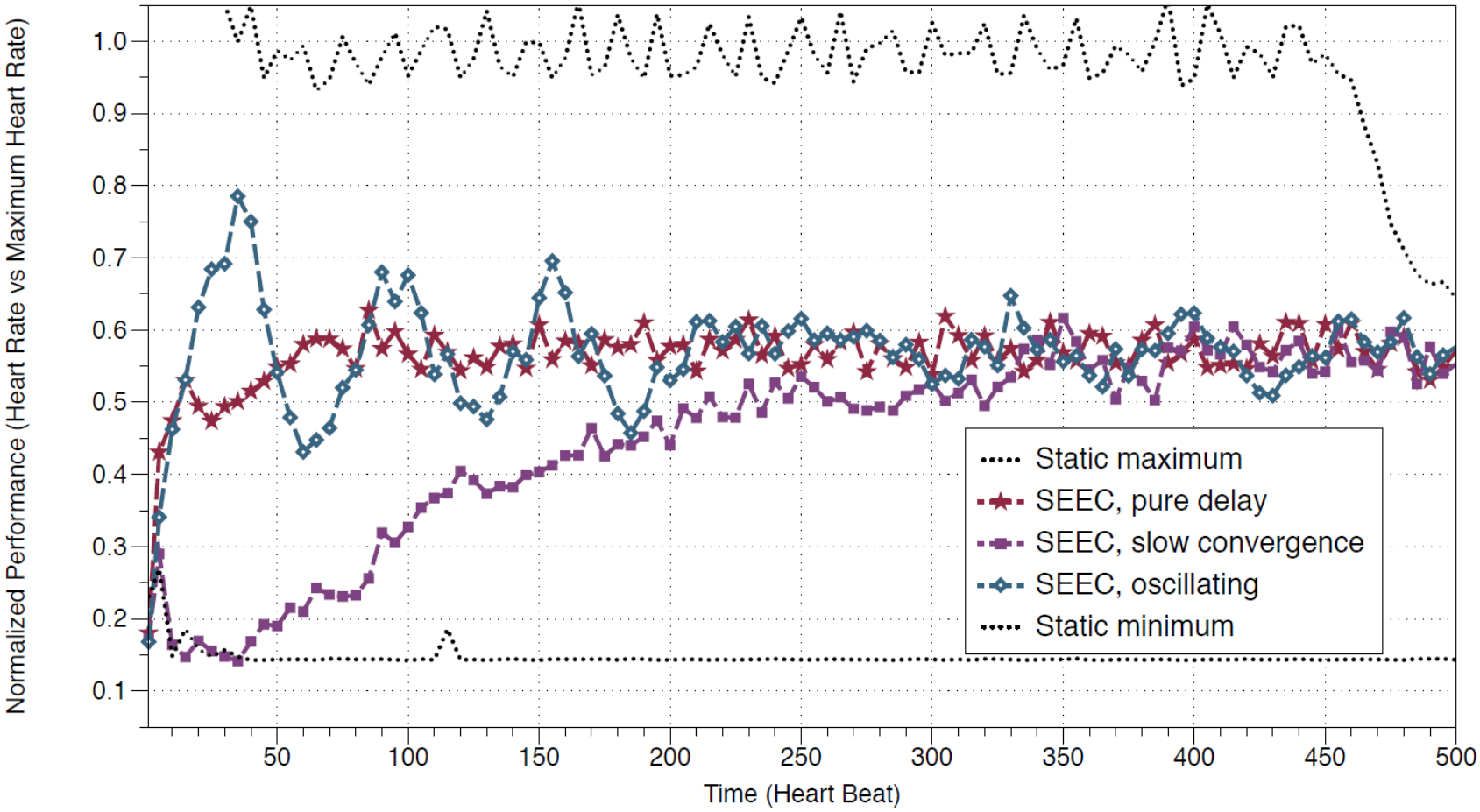
Controller	Action Set	Actuation Function	Tradeoffs
Frequency Scaler	CPU Speeds	Change CPU speed	Power vs Speed
Core Allocator	Number of available cores	Change affinity masks	Power vs Speed
DRAM Allocator	Number of available DRAM controllers	Change NUMA page allocation	Power vs Speed
Power Manager	CPU speed and in-use cores	Change CPU speed and affinity masks	Power vs Speed
Adaptive Video Encoder	Encoding Parameters and Algorithm	Change parameters, use different algorithms	Video Quality vs Speed

Application examples



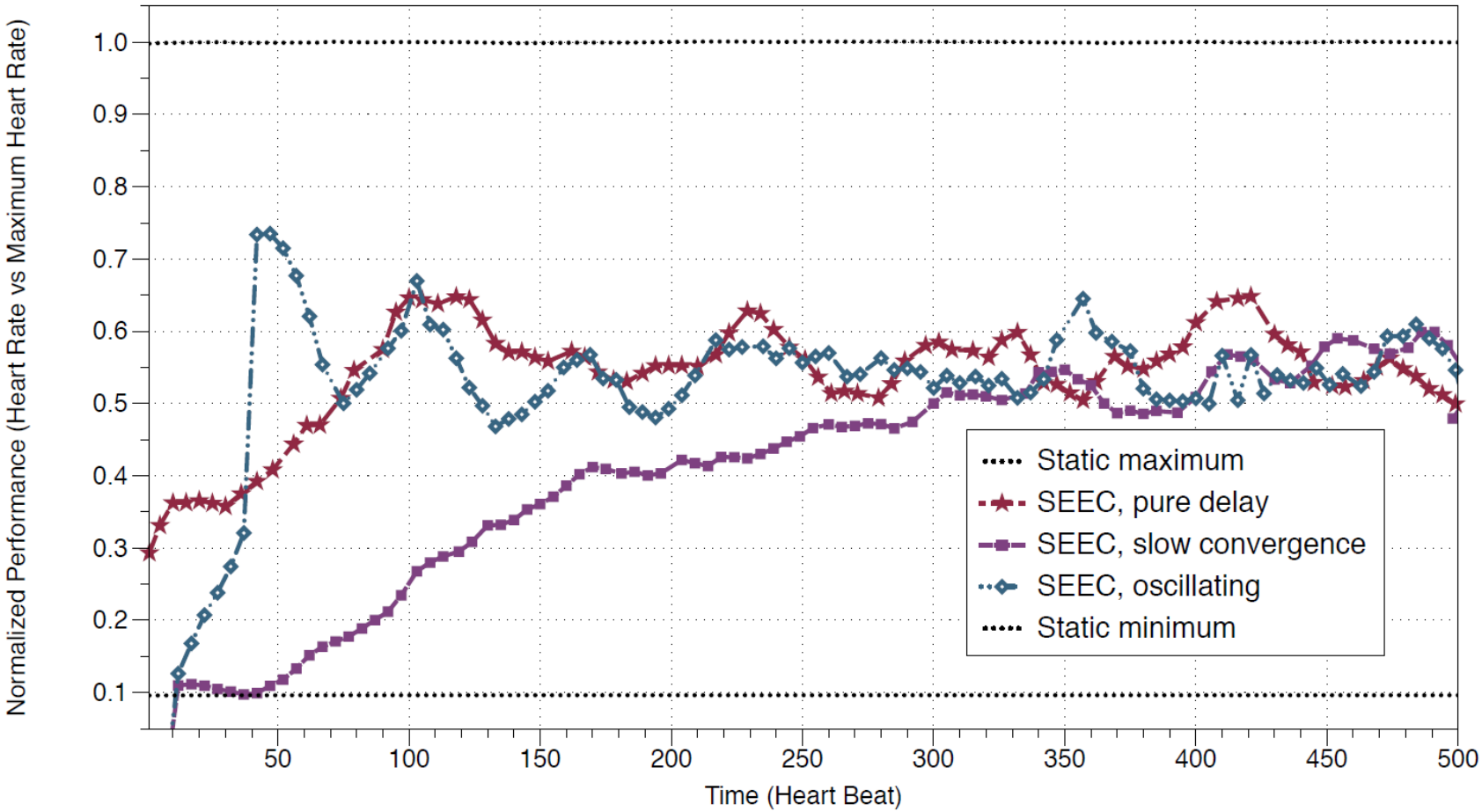
Frequency scaling for the swaptions application (PARSEC benchmark)

SEEC - A Framework for Self-Aware Computing

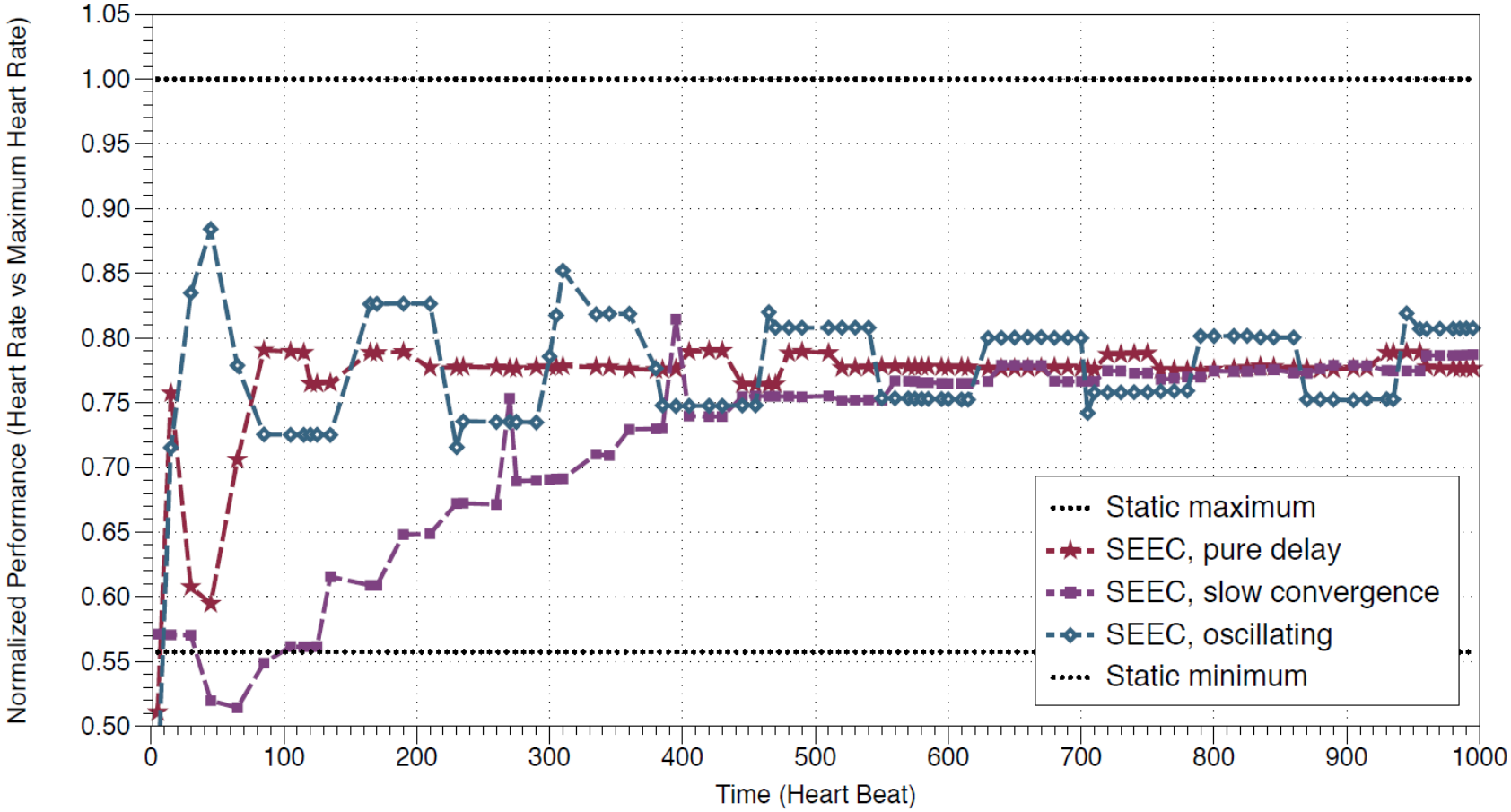


Core allocator for swaptions

SEEC - A Framework for Self-Aware Computing

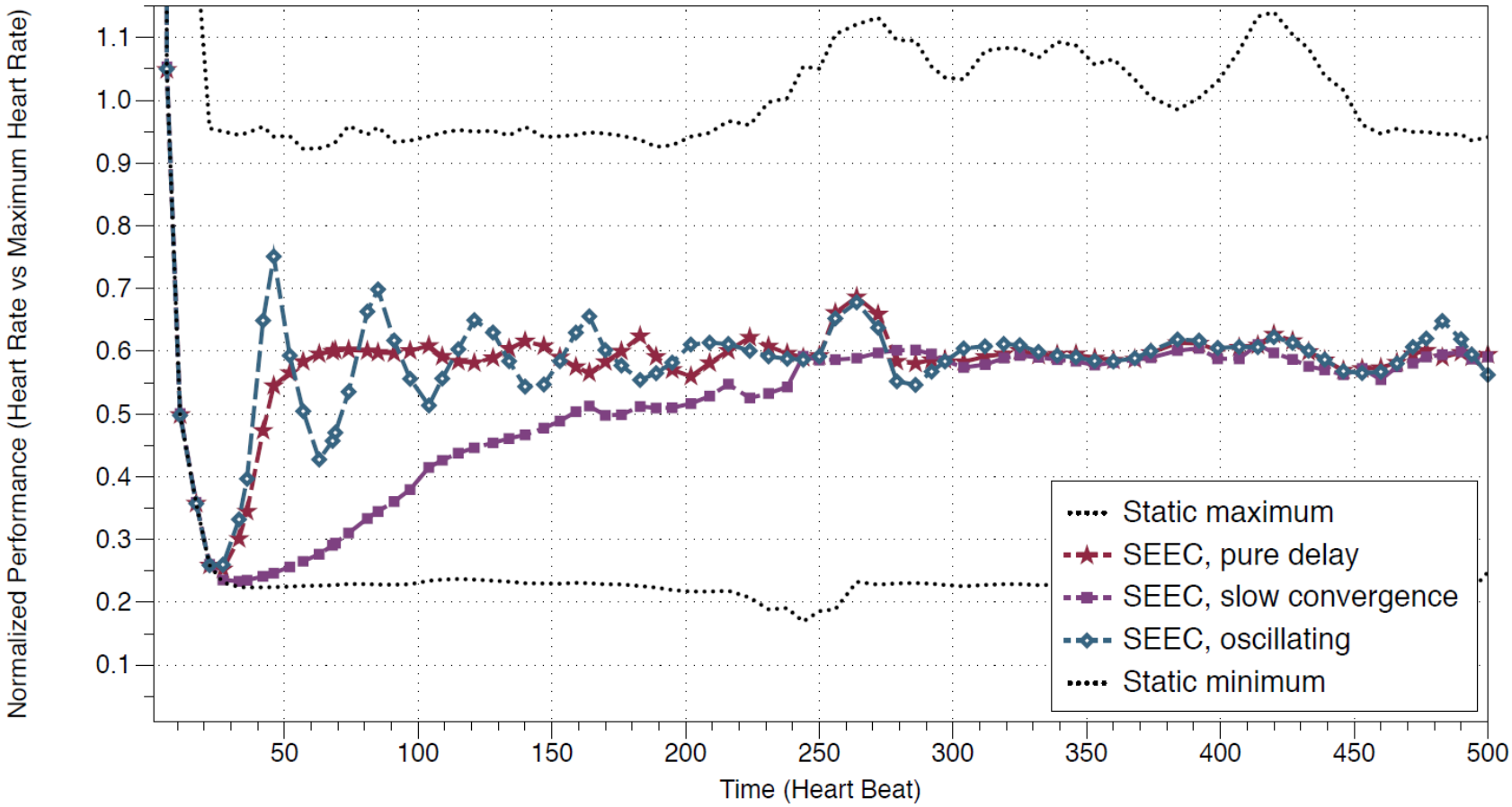


Power manager (DRAM controllers, number of cores, frequency) for swaptions



Memory allocator for STREAM (PARSEC benchmark)

SEEC - A Framework for Self-Aware Computing



Adaptive video encoder

Summary

- Separation of goals, decision making and actuation;
- Clean framework for adding observation and actuation capabilities;
- No controller hierarchy;
- Rudimentary self-awareness.

Conclusions and Outlook

Summary

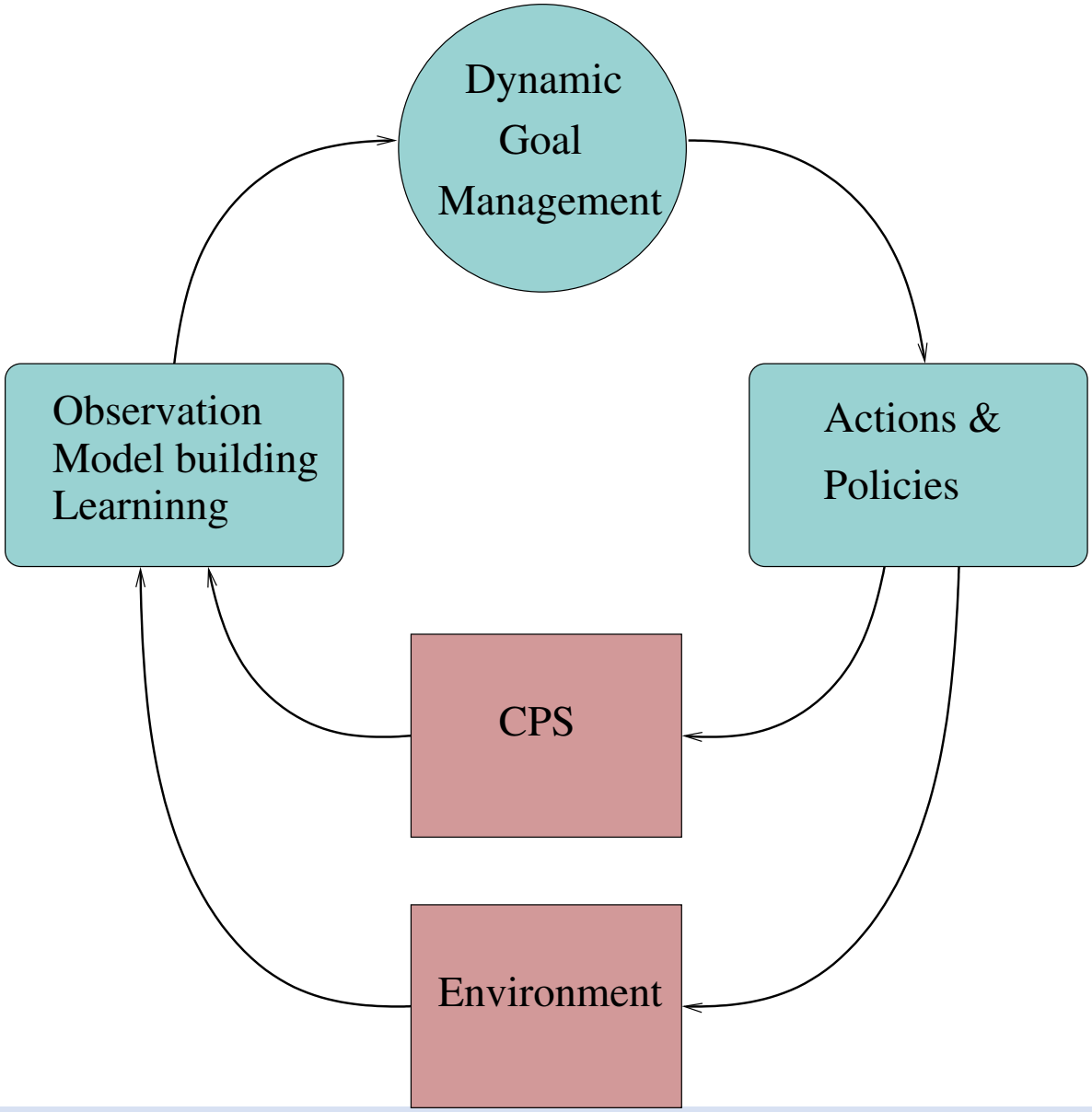
We have ...

- Set of concepts for self-awareness;
- Implementations and studies for some of them;
- An idea what self-awareness means and what it can do for us.

We still need

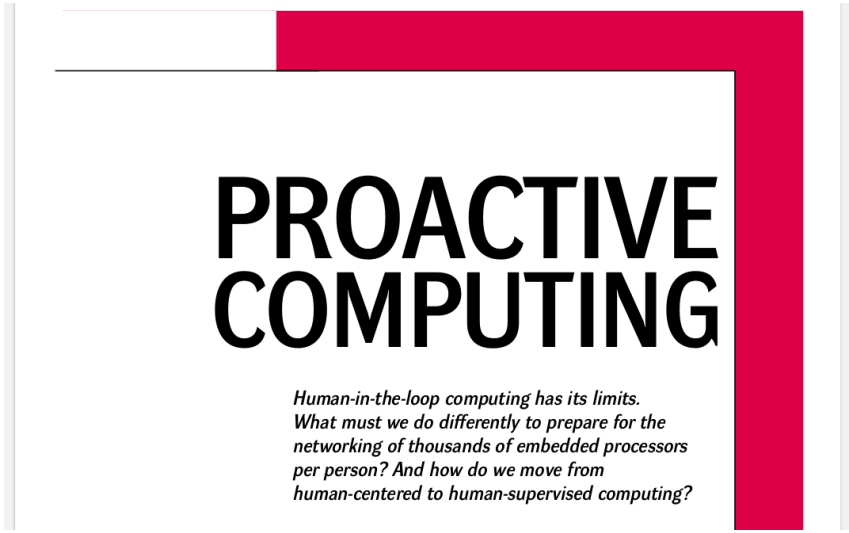
- Implementation and study complete self-aware systems;
- Comprehensive goal management;
- Comprehensive assessment in relation the goals.

Self-Aware Control Loop



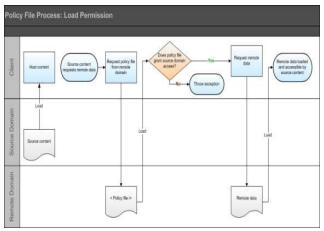
Let's Get Out

- Let's get physical
- Let's get real
- Let's get out

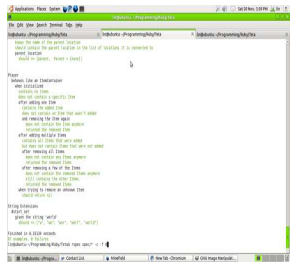


David Tennenhouse. "Proactive Computing". In: *Communications of the ACM* 43.5 (May 2000), pp. 43–50

Traditional Design Flow



Requirements specification



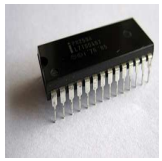
Design

Verification

Implementation

Validation

Manufacturing



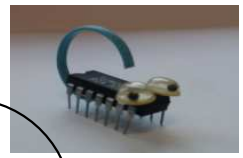
Design of Self-Aware Chips

Goals

Goal 1
Goal 2
Goal 3

Generate Subgoals

Monitor



Set parameters

Optimize

Design of Self-Aware Chips

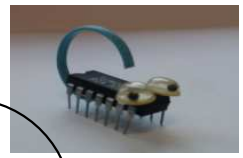
Goals

Goal 1
Goal 2
Goal 3

Continuous Optimization

Generate Subgoals

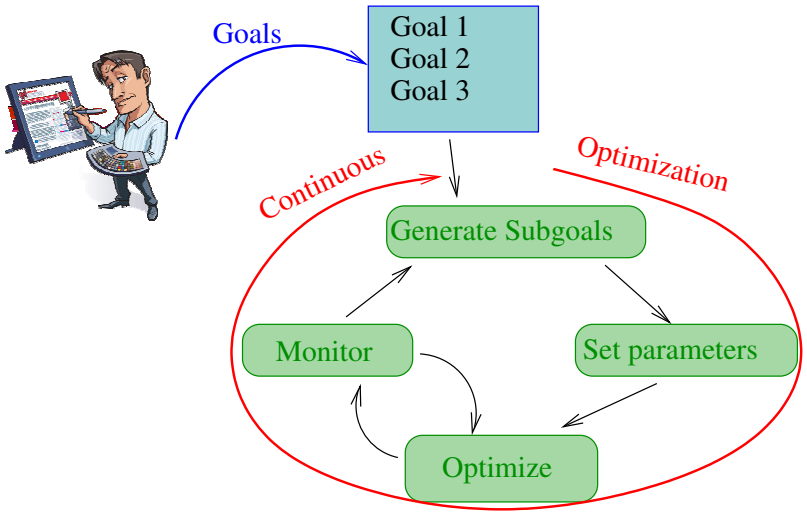
Monitor



Set parameters

Optimize

Design of Self-Aware Chips



For that to work we need Methods:

- to guarantee behavior and performance,
- to formulate and manage goals,
- for customized and efficient learning,
- to “step back and assess”.

Self-Awareness

Self-awareness, in this context, is defined by the combination of three properties that IT systems and services should possess:

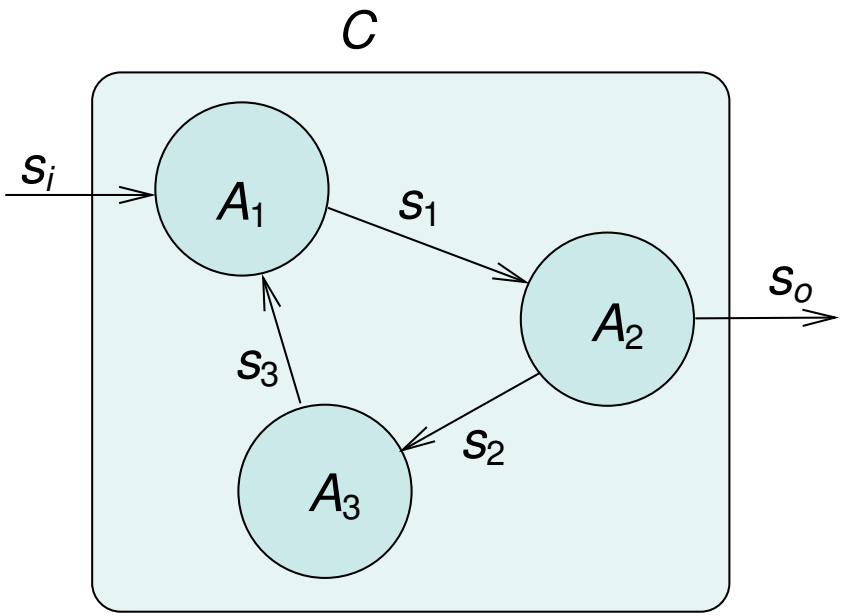
1. *Self-reflective: i) aware of their software architecture,*

- Recursive use of the term “awareness”;
- Static, not dynamic;
- The system cannot be aware of its own self-reflection.

evolves in order to ensure that their QoS requirements and respective SLAs are continuously satisfied while at the same time operating costs and energy-efficiency are optimized.

Samuel Kounev, Xiaoyun Zhu, Jeffrey O. Kephart, and Marta Kwiatkowska. “Model-driven Algorithms and Architectures for Self-Aware Computing Systems (Dagstuhl Seminar 15041)”. In: *Dagstuhl Reports* 5.1 (2015). Ed. by Samuel Kounev, Xiaoyun Zhu, Jeffrey O. Kephart, and Marta Kwiatkowska, pp. 164–196

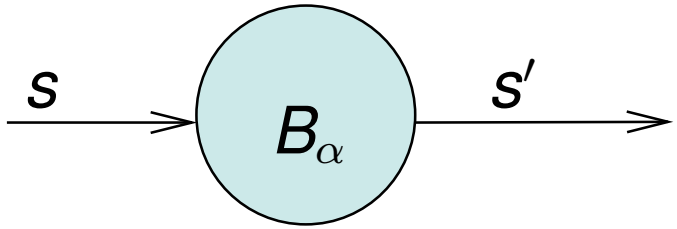
Actors in a Dynamic Dataflow Model



$$A = \langle \mathfrak{T}, I, O, z_0, f, g, \nu, \vec{\mathfrak{m}} \rangle$$

$\mathfrak{T} \subseteq \mathfrak{S}$	... set of states
$I \subseteq \mathcal{P}(S)$	... input signals
$O \subseteq \mathcal{P}(S)$	... output signals
$z_0 \in \mathfrak{T}$	... initial state
$\nu : \mathbb{N} \to \mathcal{P}(\mathbb{N})$	... input partitioning
$f : \mathcal{P}(S) \times \mathfrak{S} \to \mathcal{P}(S)$	... output encoding
$g : \mathcal{P}(S) \times \mathfrak{S} \to \mathfrak{S}$	... next state
$\vec{\mathfrak{m}} : \mathfrak{S} \to \texttt{Action}$	... a meta operator

Example Actor

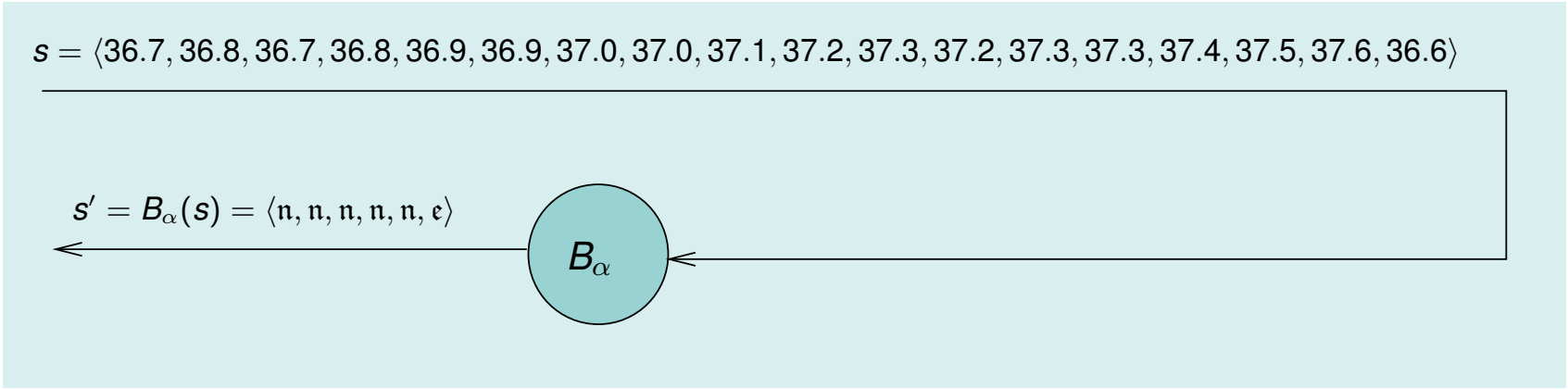


$$B_\alpha = \langle \{\}, \{s\}, \{s'\}, \cdot, \alpha, \cdot, \nu, \cdot \rangle$$

$$\nu(.) = \{3\}$$

$$\alpha(\langle t_1, t_2, t_3 \rangle) = \begin{cases} \mathfrak{l} & \text{if } (t_1 + t_2 + t_3)/3 < 35.5 \\ \mathfrak{n} & \text{if } 35.5 \leq (t_1 + t_2 + t_3)/3 < 37.5 \\ \mathfrak{e} & \text{if } 37.5 \leq (t_1 + t_2 + t_3)/3 < 38.5 \\ \mathfrak{h} & \text{if } 38.5 \leq (t_1 + t_2 + t_3)/3 \end{cases}$$

Signal Abstraction



$$\alpha(\langle t_1, t_2, t_3 \rangle) = \begin{cases} \mathfrak{l} & \text{if } (t_1 + t_2 + t_3)/3 < 35.5 \\ \mathfrak{n} & \text{if } 35.5 \leq (t_1 + t_2 + t_3)/3 < 37.5 \\ \mathfrak{e} & \text{if } 37.5 \leq (t_1 + t_2 + t_3)/3 < 38.5 \\ \mathfrak{h} & \text{if } 38.5 \leq (t_1 + t_2 + t_3)/3 \end{cases}$$

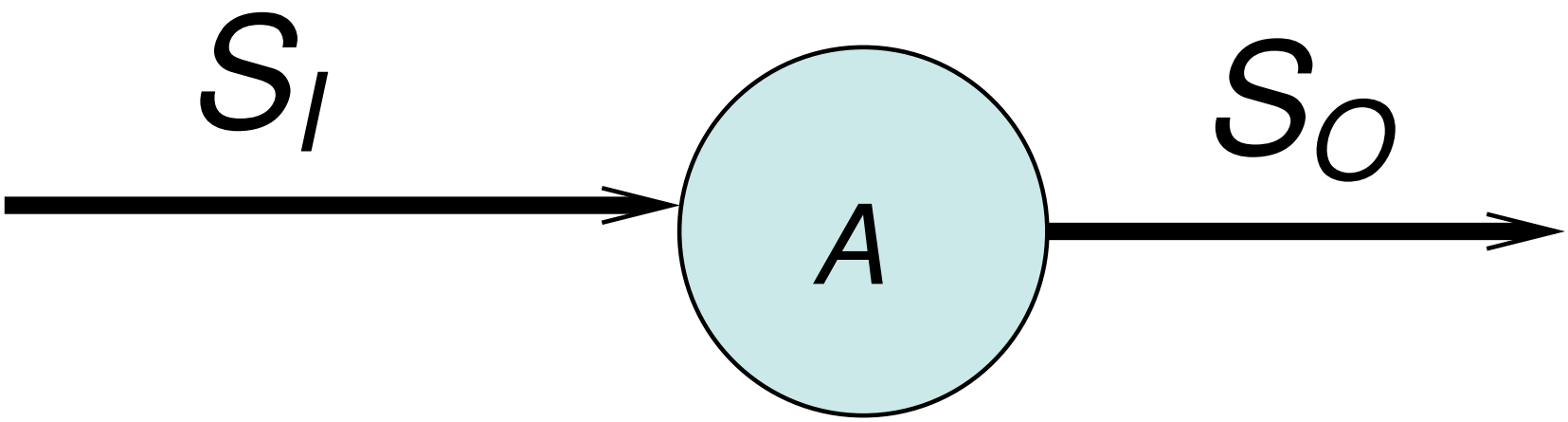
Value Abstraction

$$\alpha_V(\langle x \rangle) = \begin{cases} \mathfrak{A} & \text{if } x = \mathfrak{a} \text{ or } x = \mathfrak{b} \\ x & \text{otherwise} \end{cases}$$

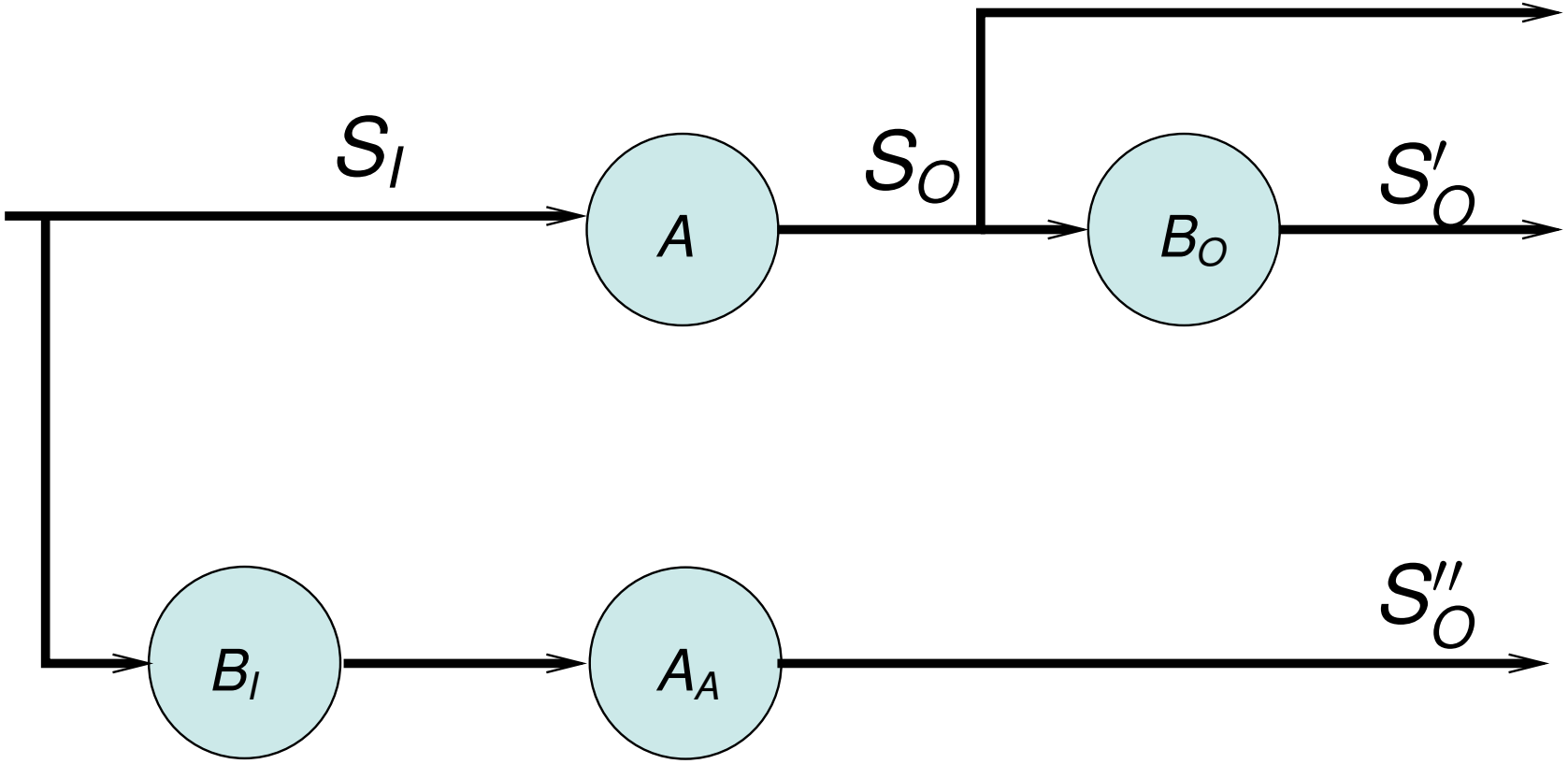
Time Abstraction

$$\alpha_t(\langle x_1, x_2 \rangle) = \begin{cases} \mathfrak{A} & \text{if } x_1 = \mathfrak{a} \text{ and } x_2 = \mathfrak{a} \\ \langle x_1, x_2 \rangle & \text{otherwise} \end{cases}$$

Actor Abstraction

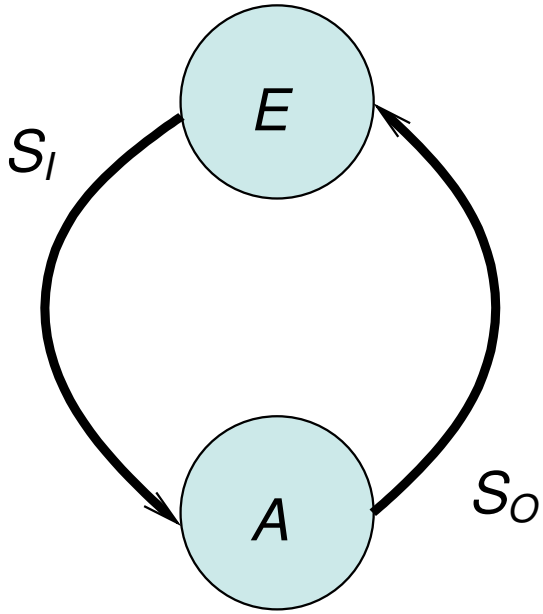


Actor Abstraction

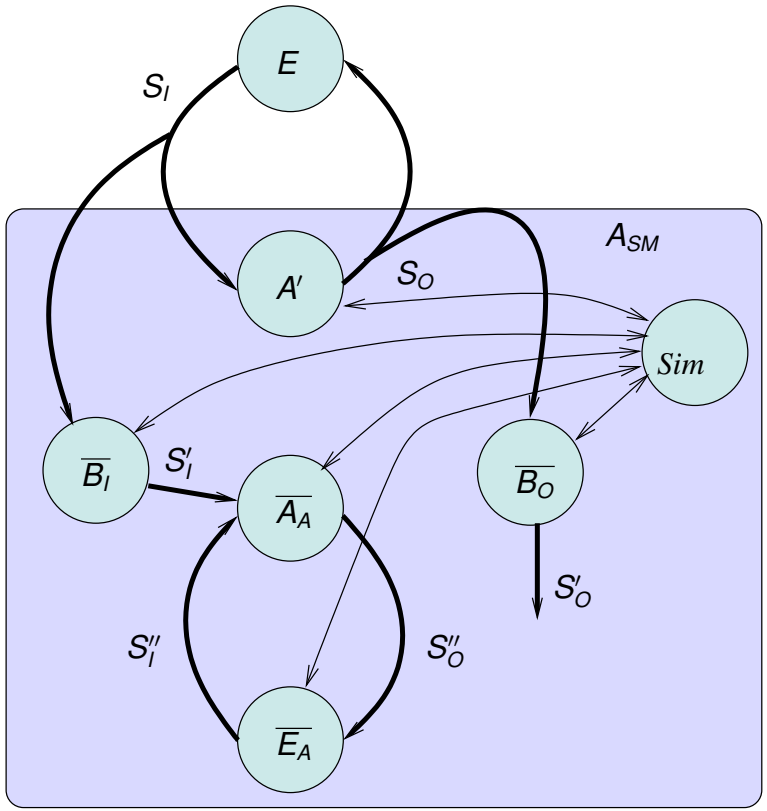


A_A is an actor abstraction of A iff $S'_O = S''_O$.

Self Mdoel



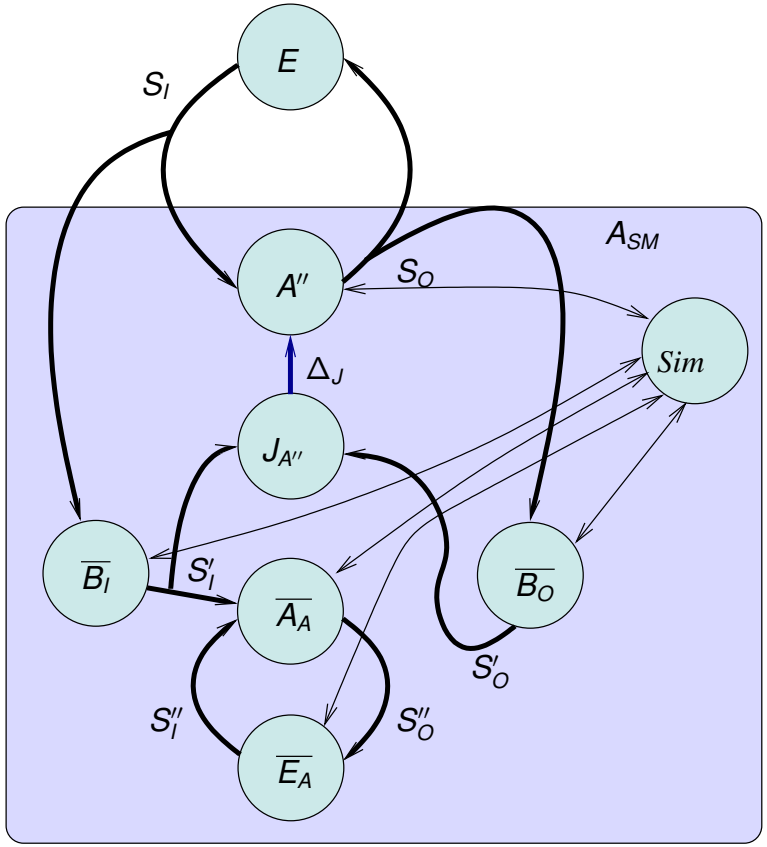
Self Model



$\bar{A}_A$ is a *simulatable actor* of A:

- $\bar{A}_A$ is an actor abstraction of A.
- It has an additional input signal denoted as *control signal*.
- It can be stopped and resumed through the control signal.
- Input signals are duplicated and controlled by the control input.
- It has an additional output signal: *status signal*.

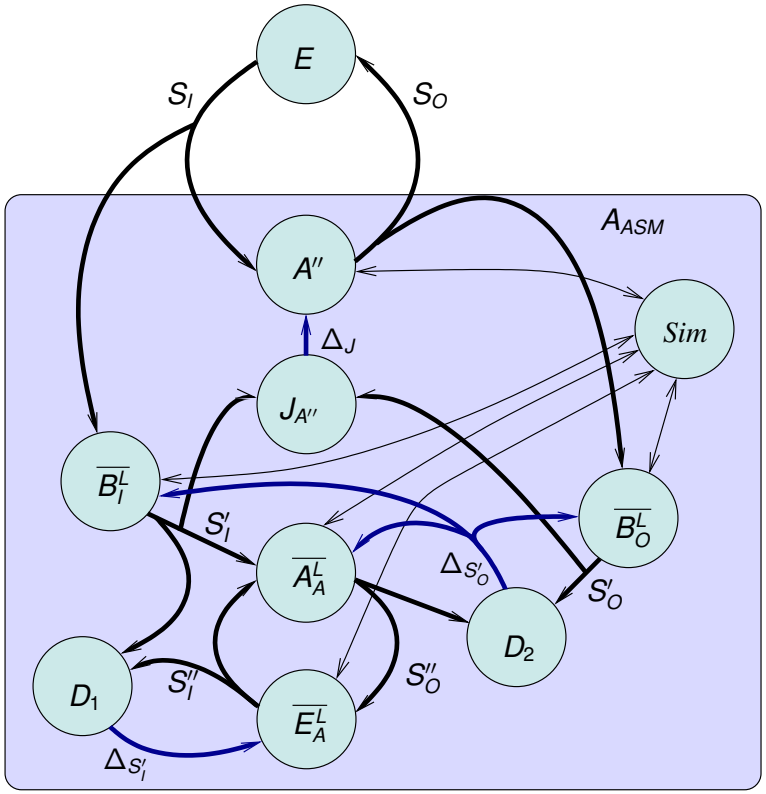
Self Assessment



$J_{A''}$ assesses the behavior and performance of actor A'':

- $J_{A''}$ monitors abstractions of inputs and outputs of A''.
- It compares the observed behavior with expected behavior.
- It reports observed differences in Δ_J.
- It maintains an assessment history.

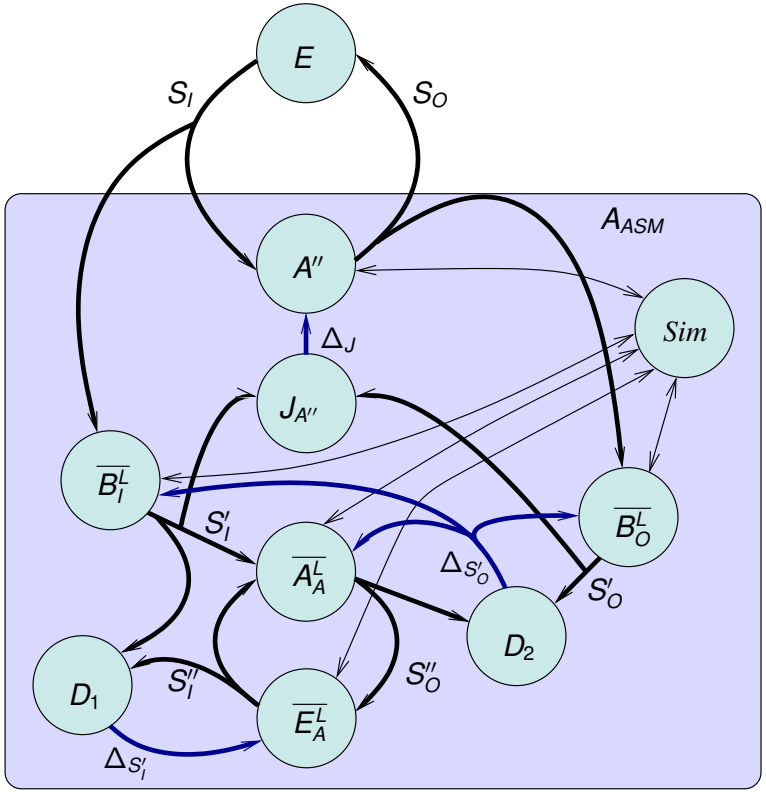
Adaptive Self-Model



A *learning actor* A^L modifies its behavior to minimize an error signal.

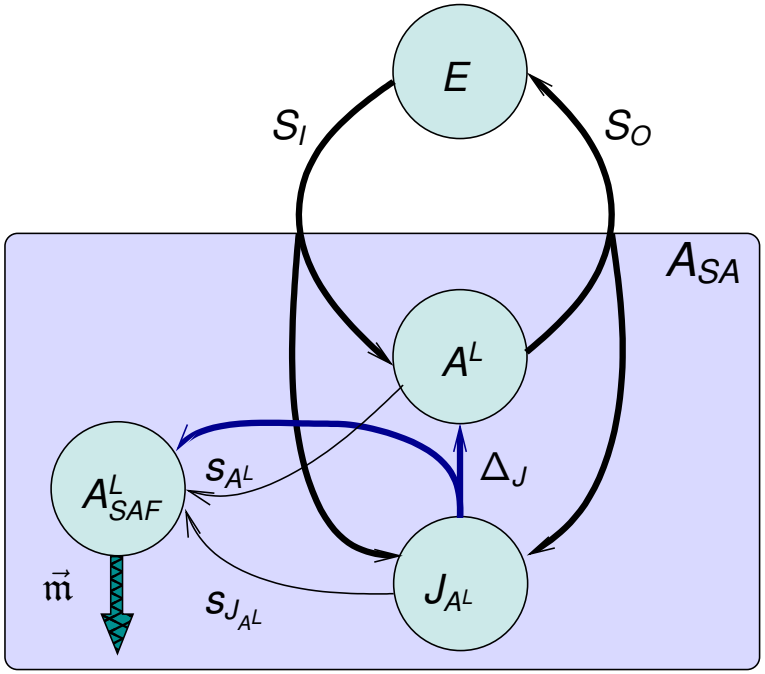
- D_1 analyses the differences between S'_I abd S''_I.
- D_2 analyses the differences between S'_O and S''_O.
- D_1 is used to improve the environment model.
- D_2 is used to improve the actor model and the signal abstractions.

Not Quite Self-Aware



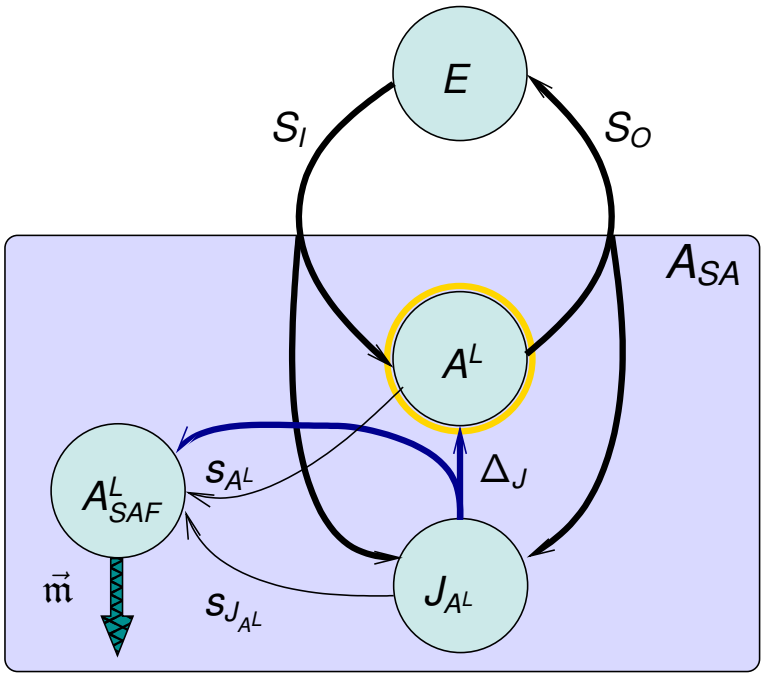
- A_{ASM} uses abstraction, simulation, learning, and a self-model.
- Self-awareness is a *process* that should be, dynamically and flexibly, applicable to a range of actors, including itself.

An Actor Facilitating Self-Awareness



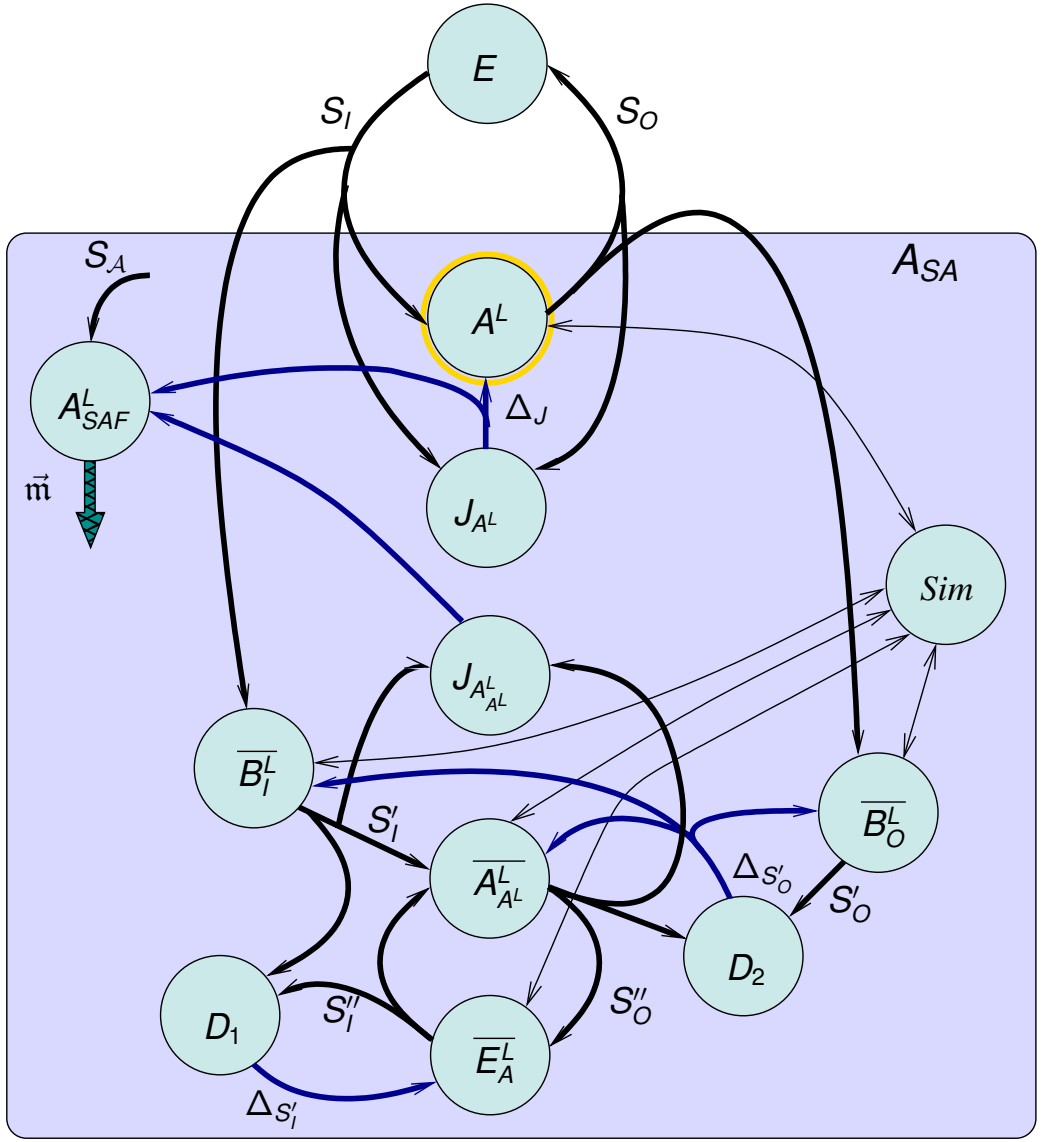
- A^L_{SAF} tracks behavior and expectations.
- It can trigger an in-depth investigation of an actor.

An Actor Facilitating Self-Awareness

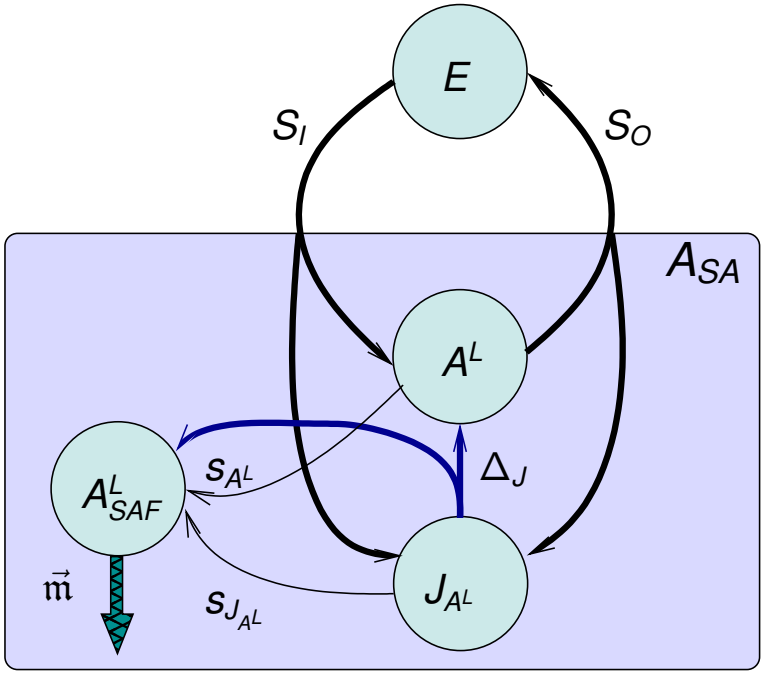


A^L_{SAF} targets A^L

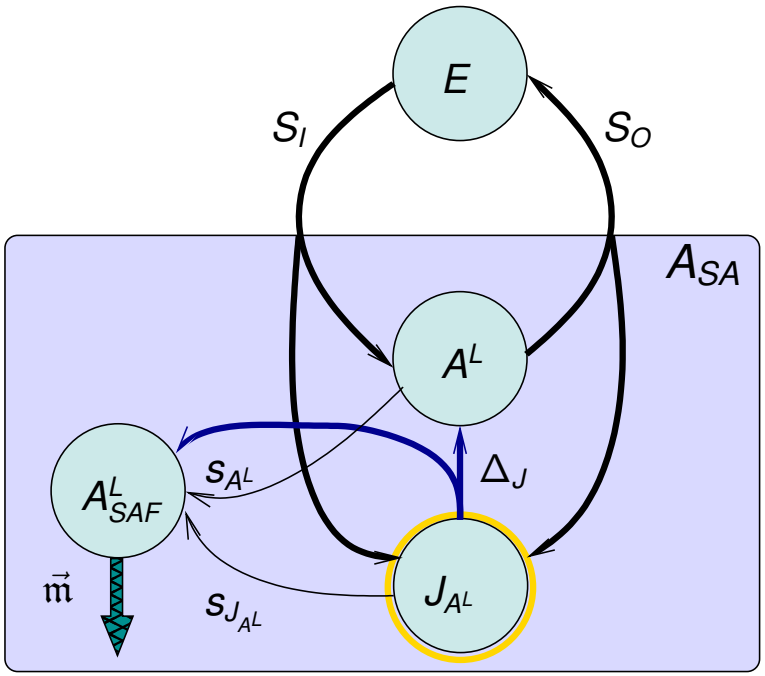
Self-Awareness Target: A^L



An Actor Facilitating Self-Awareness

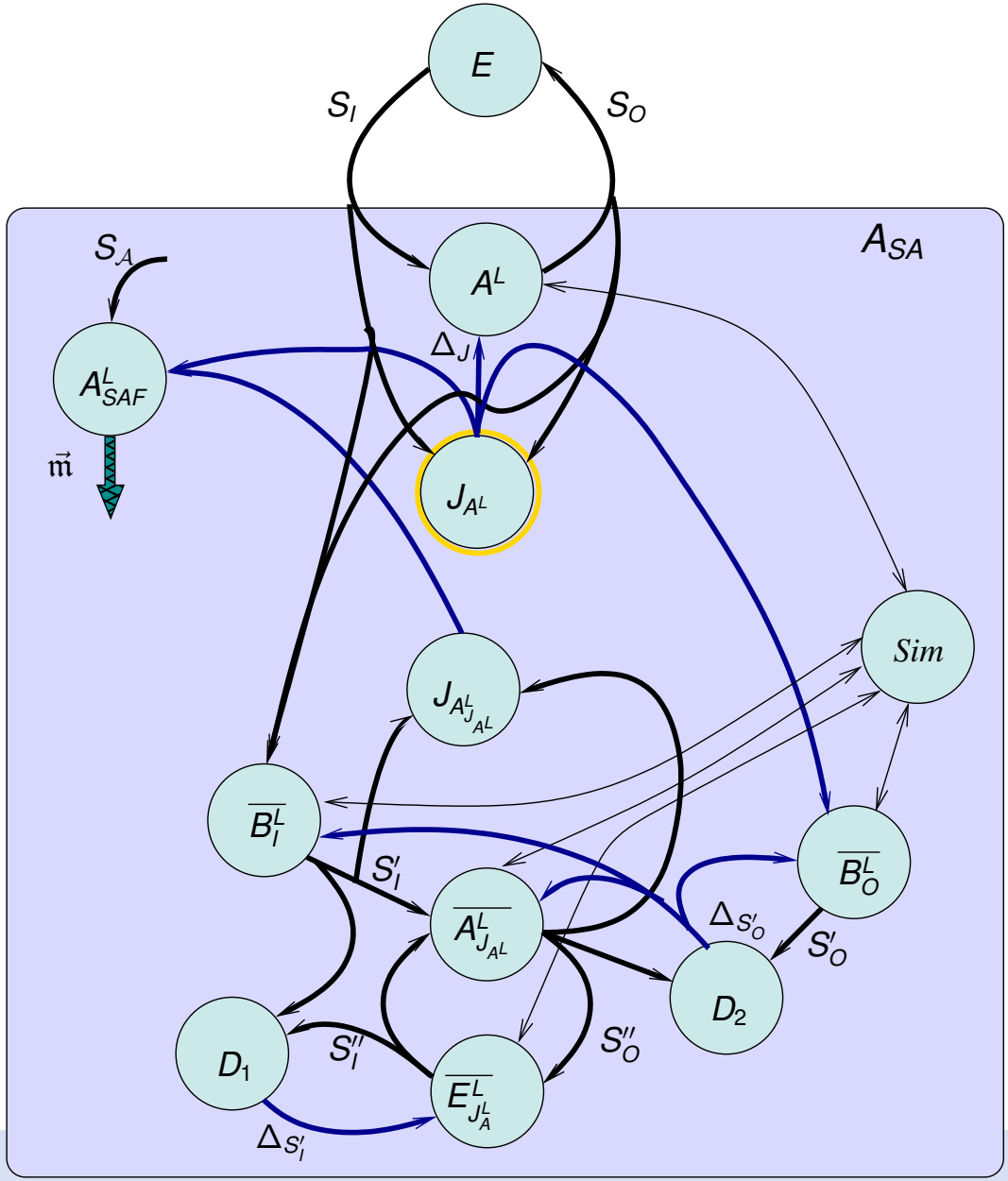


An Actor Facilitating Self-Awareness

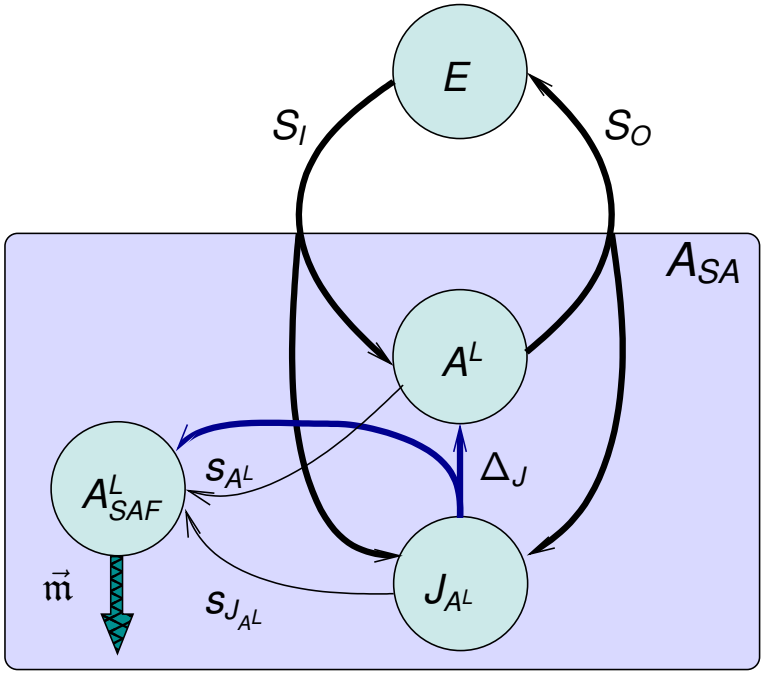


A^L_{SAF} targets J_{A^L}

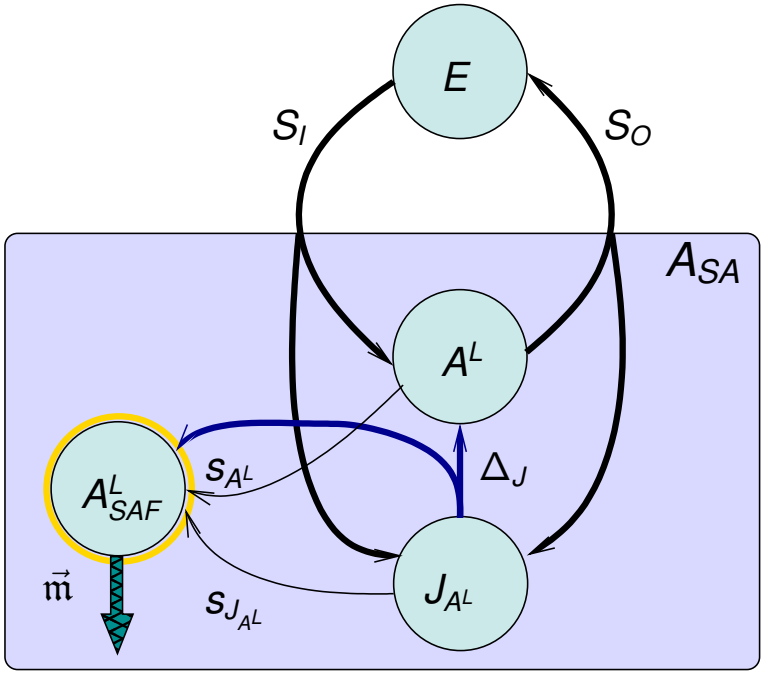
Self-Awareness Target: J_{AL}



An Actor Facilitating Self-Awareness

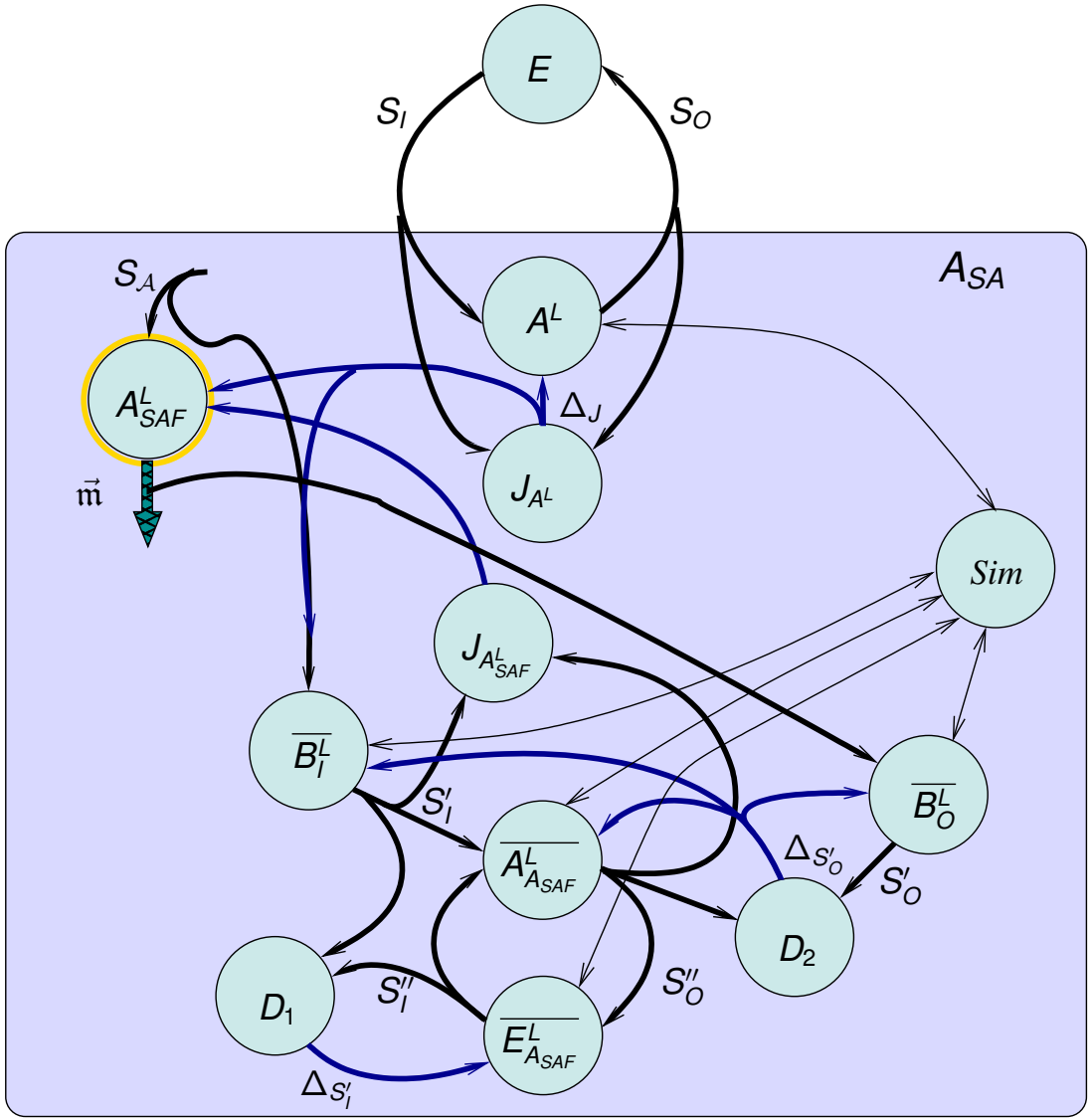


An Actor Facilitating Self-Awareness



A^L_{SAF} targets A^L_{SAF}

Self-Awareness Target: A^L_{SAF}



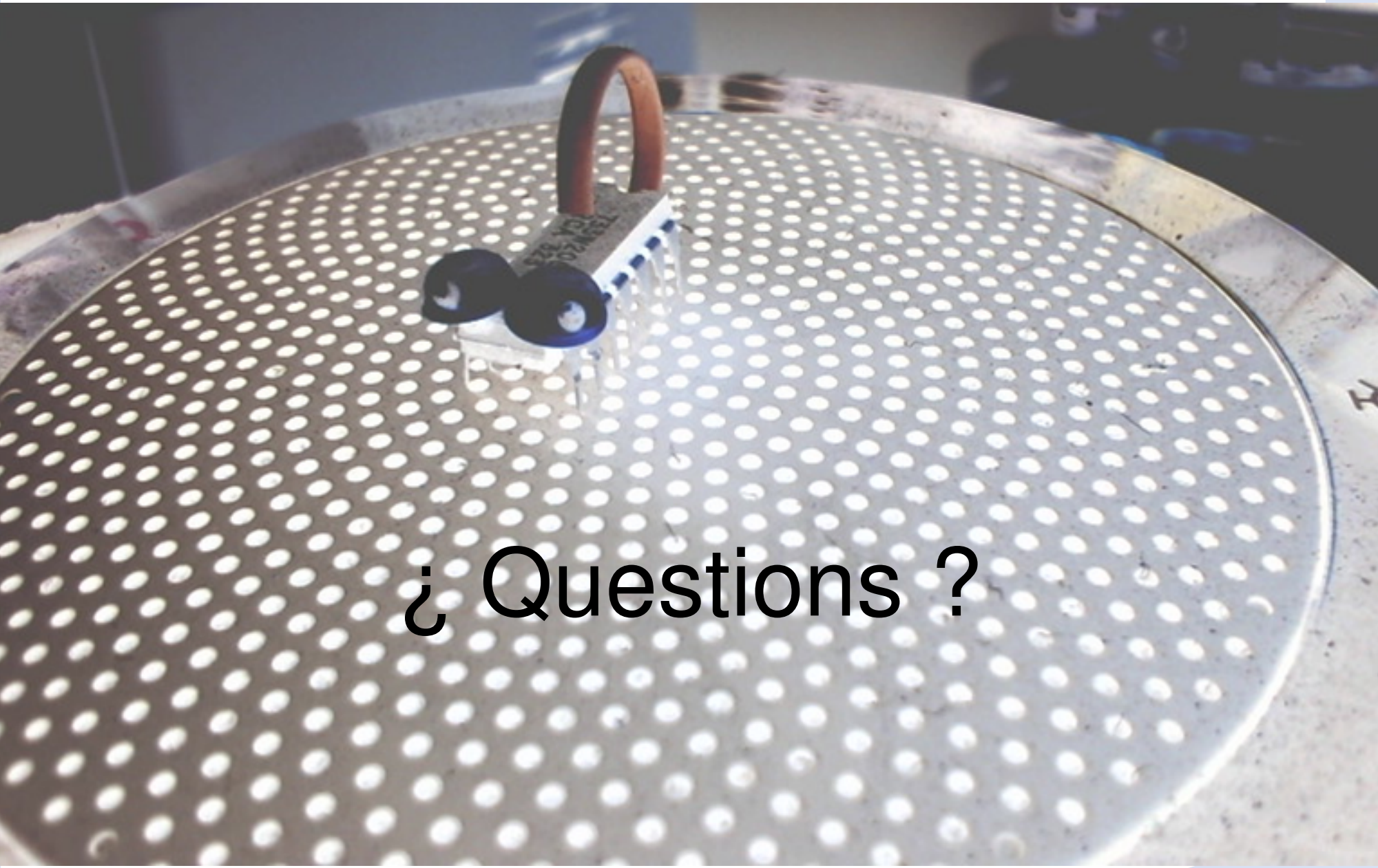
Conclusion

Promise:

- Any actor can be abstracted any number of times $\rightarrow$ no situation is too complex to analyze.
- Any actor can be subject to the scrutiny of self-awarenss.

Issues:

- Automatic, efficient abstraction techniques
- Assessment techniques
- Goal management
- Learning
- Simulation


¿ Questions ?

References I

Arman Anzanpour et al. “Self-Awareness in Remote Health Monitoring Systems using Wearable Electronics”. In: *Proceedings of Design and Test Europe Conference (DATE)*. Lausanne, Switzerland, Mar. 2017.

Nikil Dutt, Axel Jantsch, and Santanu Sarma. “Self-Aware Cyber-Physical Systems-on-Chip”. In: *Proceedings of the International Conference for Computer Aided Design*. invited. Austin, Texas, USA, Nov. 2015.

Nikil Dutt, Axel Jantsch, and Santanu Sarma. “Towards Smart Embedded Systems: A Self-Aware System-on-Chip Perspective”. In: *ACM Transactions on Embedded Computing Systems, Special Issue on Innovative Design Methods for Smart Embedded Systems* 15.2 (Feb. 2016). invited, pp. 22–27.

Farnaz Forooghifar, Amir Aminifar, and David Atienza Alonso. “Self-Aware Wearable Systems in Epileptic Seizure Detection”. In: *21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic, August 29-31, 2018*. 2018, pp. 426–432.

M. Götzinger, N. TaheriNejad, H. A. Kholerdi, and A. Jantsch. “On the design of context-aware health monitoring without a priori knowledge; an AC-Motor case-study”. In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Apr. 2017, pp. 1–5.

References II

Maximilian Götzinger et al. “Applicability of Context-Aware Health Monitoring to Hydraulic Circuits”. In: *The 44th Annual Conference of the IEEE Industrial Electronics Society*. 2018.

Maximilian Götzinger et al. “Model-free Condition Monitoring with Confidence”. In: *International Journal of Computer Integrated Manufacturing* 32.4-5 (2019).

Liang Guang, Ethiopia Nigussie, Pekka Rantala, Jouni Isoaho, and Hannu Tenhunen. “Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip”. In: *ACM Trans. Embed. Comput. Syst.* 9.3 (2010), pp. 1–24.

Liang Guang. “Hierarchical Agent-based Adaptation for Self-Aware Embedded Computing Systems”. PhD thesis. Turku, Finland: University of Turku, 2012.

Mohammad-Hashem Haghbayan, Anil Kanduri, Amir-Mohammad Rahmani, Pasi Liljeberg, Axel Jantsch, and Hannu Tenhunen. “MapPro: Proactive Runtime Mapping for Dynamic Workloads by Quantifying Ripple Effect of Applications on Networks-on-Chip”. In: *Proceedings of the International Symposium on Networks on Chip*. Vancouver, Canada, Sept. 2015.

References III

M. H. Haghbayan, A. Miele, A. M. Rahmani, P. Liljeberg, and H. Tenhunen. “A lifetime-aware runtime mapping approach for many-core systems in the dark silicon era”. In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2016, pp. 854–857.

Henry Hoffmann, Martina Maggio, Marco D Santambrogio, Alberto Leva, and Anant Agarwal. *Seec: A framework for self-aware computing*. Tech. rep. MIT-CSAIL-TR-2010-049. Cambrige, Massachusetts: MIT, Oct. 2010.

Axel Jantsch et al. “Hierarchical Dynamic Goal Management for IoT Systems”. In: *Proceedings of the IEEE International Symposium on Quality Electronic Design (ISQED 2018)*. USA, Mar. 2018.

Axel Jantsch and Kalle Tammemäe. “A Framework of Awareness for Artificial Subjects”. In: *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis*. CODES ’14. New Delhi, India: ACM, 2014, 20:1–20:3.

References IV

Anil Kanduri, Mohammad-Hashem Haghbayan, Amir-Mohammad Rahmani, Pasi Liljeberg, Axel Jantsch, and Hannu Tenhunen. “Dark Silicon Aware Runtime Mapping for Many-core Systems: A Patterning Approach”. In: *Proceedings of the International Conference on Computer Design (ICCD)*. New York City, USA, Oct. 2015, pp. 610–617.

Anil Kanduri et al. “Approximation Knob: Power Capping Meets Energy Efficiency”. In: *Proceedings of the International Conference on Computer Aided Design (ICCAD)*. Austin, Texas, Nov. 2016.

Samuel Kounev, Xiaoyun Zhu, Jeffrey O. Kephart, and Marta Kwiatkowska. “Model-driven Algorithms and Architectures for Self-Aware Computing Systems (Dagstuhl Seminar 15041)”. In: *Dagstuhl Reports* 5.1 (2015). Ed. by Samuel Kounev, Xiaoyun Zhu, Jeffrey O. Kephart, and Marta Kwiatkowska, pp. 164–196.

S. Kounev, J.O. Kephart, A. Milenkoski, and X. Zhu, eds. *Self-Aware Computing Systems*. Springer, 2017.

References V

Hedyeh A. Kholerdi, Nima TaheriNejad, and Axel Jantsch. “Enhancement of Classification of Small Data Sets Using Self-awareness - An Iris Flower Case-Study”. In: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence, Italy, May 2018.

Peter R. Lewis et al. “Architectural Aspects of Self-aware and Self-expressive Computing Systems”. In: *IEEE Computer* (Aug. 2015).

Peter R. Lewis, Marco Platzner, Bernhard Rinner, Jim Torresen, and Xin Yao, eds. *Self-Aware Computing Systems: An Engineering Approach*. Springer, 2016.

Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Leonardi. “Experiencing SAX: a Novel Symbolic Representation of Time Series”. In: *Data Mining and Knowledge Discovery* 15.2 (Oct. 2007), pp. 107–144.

T. R. Mück, B. Donyanavard, K. Moazzemi, A. M. Rahmani, A. Jantsch, and N. D. Dutt. “Design Methodology for Responsive and Robust MIMO Control of Heterogeneous Multicores”. In: *IEEE Transactions on Multi-Scale Computing Systems* PP.99 (2018), pp. 1–1.

References VI

T. S. Muthukaruppan et al. “Hierarchical power management for asymmetric multi-core in dark silicon era”. In: *Proc. of DAC*. 2013.

Katayoun Neshatpour, Farnaz Behnia, Houman Homayoun, and Avesta Sasan. “ICNN: An iterative implementation of convolutional neural networks to enable energy and computational complexity aware dynamic approximation”. In: *2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018*. 2018, pp. 551–556.

J.-S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis. “The benefits of self-awareness and attention in fog and mist computing”. In: *IEEE Computer, Special Issue on Self-Aware/Expressive Computing Systems* (July 2015), pp. 37–45.

Martin Radetzki, Chaochao Feng, Xueqian Zhao, and Axel Jantsch. “Methods for Fault Tolerance in Networks-on-Chip”. In: *ACM Computing Surveys* 46.1 (July 2013), 8:1–8:38.

Amir M. Rahmani et al. “SPECTR - Formal Supervisory Control and Coordination for Many-core Systems Resource Management”. In: *Proceedings of the 23rd ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. Williamsburg, VA, USA, Mar. 2018.

References VII

Amir M. Rahmani, Axel Jantsch, and Nikil Dutt. “HDGM: Hierarchical Dynamic Goal Management for Many-Core Resource Allocation”. In: *IEEE Embedded Systems letters* 10.3 (Sept. 2018).

Santanu Sarma, Nikil Dutt, N. Venkatasubramaniana, A. Nicolau, and P. Gupta. *CyberPhysical-System-On-Chip (CPSoC): Sensor-Actuator Rich Self-Aware Computational Platform*. Tech. rep. CECS Technical Report No: CECS TR–13–06. Irvine, CA 92697-2620, USA: Center for Embedded Computer Systems University of California, Irvine, May 2013.

Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. “On-Chip Self-Awareness Using Cyberphysical-Systems-On-Chip (CPSoC)”. In: *Proceedings of the 12th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. New Delhi, India, Oct. 2014.

Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. “CyberPhysical-System-On-Chip (CPSoC): A Self-Aware MPSoC Paradigm with Cross-Layer Virtual Sensing and Actuation”. In: *Proceedngs of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*. Grenoble, France, Mar. 2015.

References VIII

Santanu Sarma, T. Muck, L. A.D. Bathen, N. Dutt, and A. Nicolau. "SmartBalance: A Sensing-Driven Linux Load Balancer for Energy Efficiency of Heterogeneous MPSoCs". In: *Proceedings of the Design Automation Conference*. July 2015.

Santanu Sarma and Nikil Dutt. "Cross-Layer Exploration of Heterogeneous Multicore Processor Configurations". In: *VLSI Design Conference*. 2015.

Elham Shamsa, Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Axel Jantsch, and Nikil Dutt. "Goal Formulation: Abstracting Dynamic Objectives for Efficient On-chip Resource Allocation". In: *IEEE Nordic Circuits and Systems Conference (NorCAS)*. Tallinn, Estonia, Oct. 2018.

Elham Shamsa, Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Axel Jantsch, and Nikil Dutt. "Goal-Driven Autonomy for Efficient On-chip Resource Management: Transforming Objectives to Goals". In: *Proceedings of the Design and Test Europe Conference (DATE)*. Florence, Italy, Mar. 2019.

David Tennenhouse. "Proactive Computing". In: *Communications of the ACM* 43.5 (May 2000), pp. 43–50.

References IX

Nima TaheriNejad and Axel Jantsch. “Improved Machine Learning using Confidence”. In: *IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. Edmonton, Canada, May 2019.

Nima TaheriNejad, Axel Jantsch, and David Pollreisz. “Comprehensive Observation and its Role in Self-Awareness - An Emotion Recognition System Example”. In: *Proceedings of the Federated Conference on Computer Science and Information Systems*. Gdansk, Poland, Sept. 2016.

Nima TaheriNejad, M. Ali Shami, and Sai Manoj P. D. “Self-aware sensing and attention-based data collection in Multi-Processor System-on-Chips”. In: *15th IEEE International New Circuits and Systems Conference (NEWCAS)*. June 2017, pp. 81–84.

Youmin Zhang and Jin Jiang. “Bibliographical review on reconfigurable fault-tolerant control systems”. In: *Annual reviews in Control* 32 (2008), pp. 229–252.