Sensor Signal Processing

Prof. Dr.-Ing. Andreas König
Lehrstuhl Integrierte Sensorsysteme

FB Elektrotechnik und Informationstechnik
Technische Universität Kaiserslautern

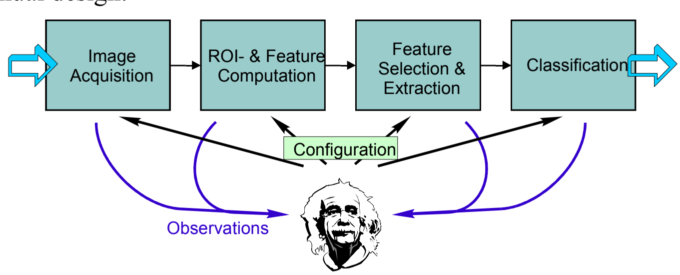
Fall Semester 2005

Course Contents

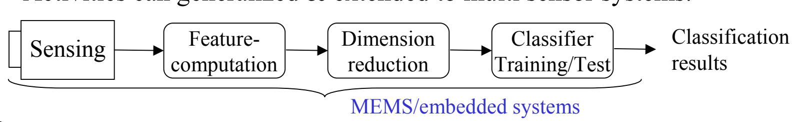
Chapter Contents

Requirements & Approach

- The introduction discussed the widespread application of intelligent sensing and recognition systems from military to consumer application
- State-of-the-art design style predominantly is still experience driven manual design:

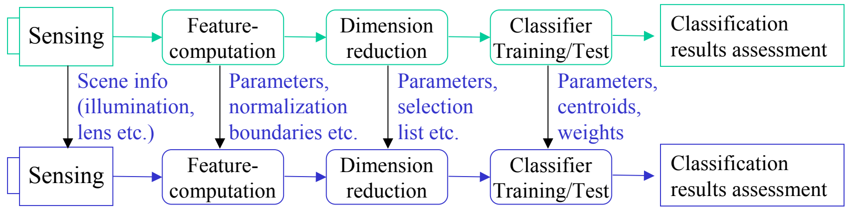


- The field of industrial image processing has seen the most intensive efforts on alleviating and automating system design and adaptation
- Activities can generalized & extended to multi sensor systems:

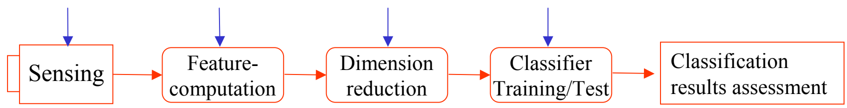


Requirements & Approach

- The practical application demands three phases in the design
- **Phase 1:** Design a training system on appropriately selected training data



- **Phase 2:** After validation of training system, take structure, parameters, and configuration values computed from training data to test system
- **Phase 3:** After validation of training/test system deploy to operation, i.e., try the system on life images/sensor registrations & store/assess results



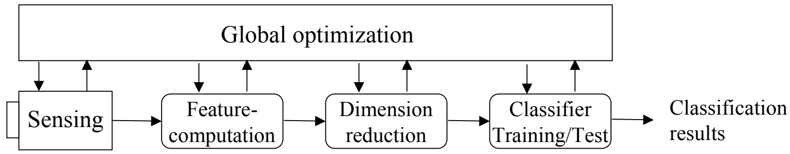
- **Phase 4:** Embedd system/sensors under constraints (real-time, power, …)

Requirements & Approach

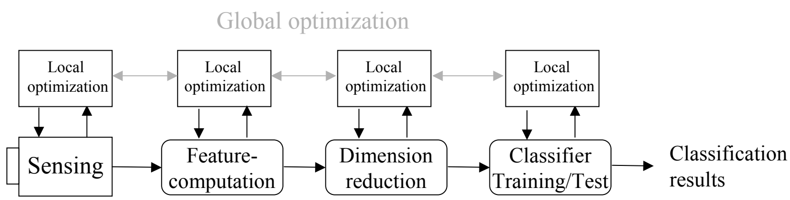
- So, we have to fix structure, i.e., select & combine (heuristic) methods, optimize setting parameters of these methods, and generate functions
- What actually could be improved in these design tasks and how ?
- **Option 1:** Optimize system design by optimum support for human expert and operator:
  - ✓ Provide effective domain-specific visualization aides to make every design step an its results transparent, e.g., feature space visualization, sample set editor etc.
  - ✓ Provide numerical criteria to assess intermediate and result data quality, e.g., q_o, q_s, q_c, E, classification rate etc.
  - ✓ Provide a toolbox of algorithmic standard cells as lego or building-blocks for system construction
- Use these features to design in an open-loop or human-centered approach an optimum system !

Requirements & Approach

- Theoretically, the dependence on design decision on higher abstraction levels should potentially lead to revisions on lower-levels

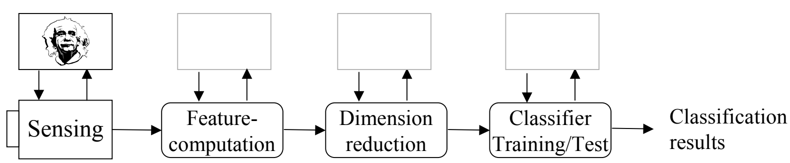


- This approach easily becomes intractable and a divide-et-impera design style is commonly pursued



Requirements & Approach

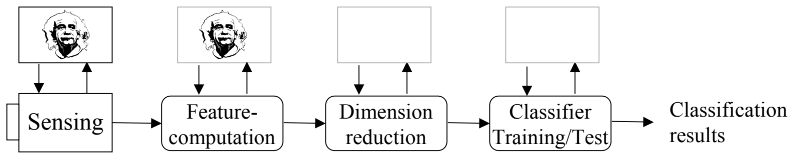
- Open-loop approaches interactively assess samples or whole sample sets at different stages of the processing and optimize moving from low to high abstraction level



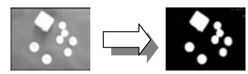
- Select sensor(s)
- Fix scene with regard to invariance
- Data acquisition

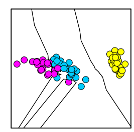
Requirements & Approach

- After sensor & scene optimization. Signal/image preprocessing fllowd by feature computation takes place



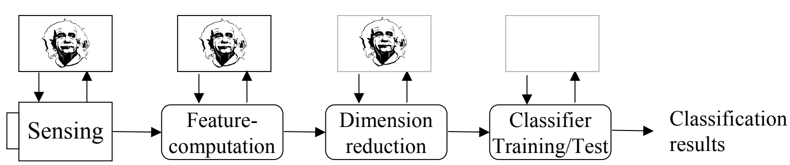
- Select heuristic method(s)
- Combine methods
- Generate application-specific processing functions
- Assess signal/image preprocessing
- Assess raw feature data

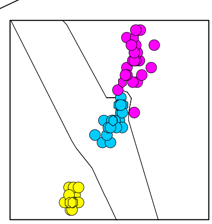




Requirements & Approach

- Systematic dimensionality reduction follows with the objective to achieve a lean and well-performing robust system

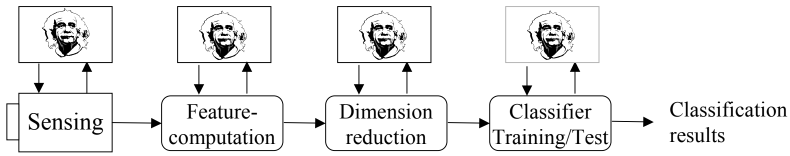




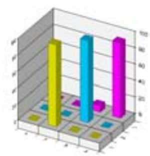
- Select reduction method(s)
- Combine methods
- Assess compacted feature data

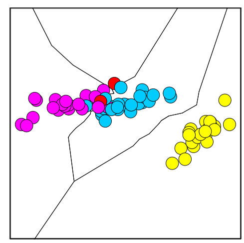
Requirements & Approach

- Finally, (hierarchical) classification based on the optimized feature space is investigated



- Select classification method(s)
- Combine methods to hierarchy
- Assess classification results (training/test/loo/on-line operation)



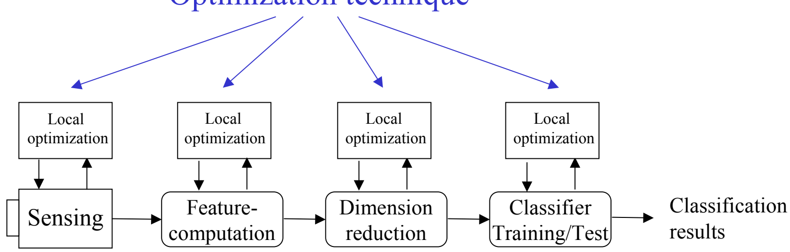


```
Rate Tr : 100 %
Rate Te :  97.333 %
```


Requirements & Approach

- **Option 2:** Use prerequisites and techniques of **Option 1**, in particular the numerical assessment measures and optimize in closed-loop operation

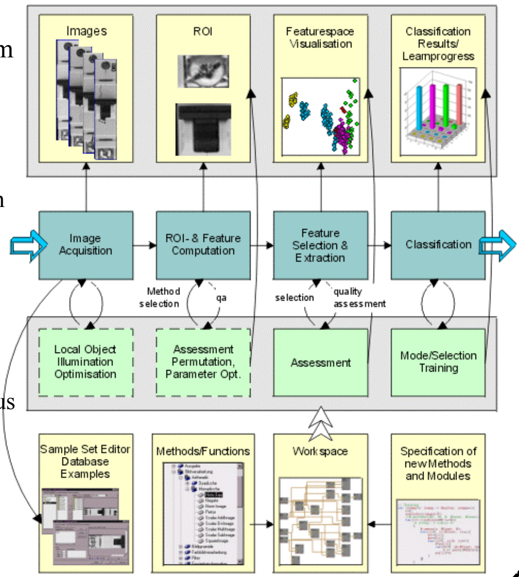
Optimization technique



- **Option 2** requires appropriate optimization techniques for the underlying optimization problems and available assessment function
- Gradient descent techniques, for instance require derivable cost functions !
- *Potentially, automated system design can include restricted global mechanisms/properties in the optimization process*

Requirements & Approach

- First architecture & implementation: QuickCog-system
  - Fast & consistent design
  - Assessment and optimization
  - Intra/inter level optimization
  - Holistic modelling and simulation
  - Opportunistic & parsimonious
  - DR (AFS) salience: physical savings !



Optimization Techniques

- In addition to assessment criteria, optimization techniques are employed for the optimization implied by the semi-automatic or automatic activities
- Heuristics and gradient descent techniques have been introduced so far
- Random search and evolutionary computation techniques offer the practical advantage to reach better minimum on arbitrary cost function

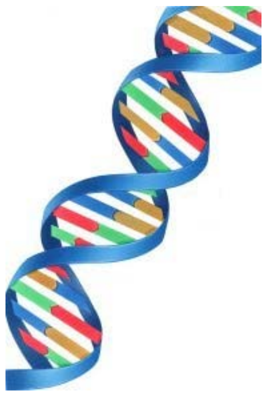
Optimization techniques

deterministic | non-deterministic

heuristics e.g., SBS/SFS | gradient descent (initialization !) | stochastic pseudo-grad., SA | evolutionary comp. GA, EV, GP, PSO

Evolutionary Computation

- Algorithms mimicking nature and the evolution of living beeings for the optimization of technical devices and systems have become commonplace
- Simulated Darwinian selection using populations, selection, recombination (cross over), mutation operators:



Source: Carlyn Iverson - The American Heritage ® Children's Science Dictionary

Initialize population
(Individual → chromosomes)
↓
Assess population (fitness computation) ← Next generation
↓
Fitness threshold/ Step limit → Return fittest
↓
Recombination (cross over)
↓
Mutation → Next generation

Contribution from Peters & König acknowledged

Evolutionary Computation

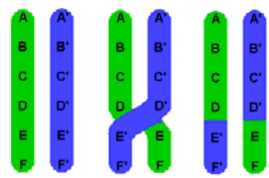
- **Selection techniques:**

- To preserve good solutions in the optimization process, a certain number (N_e=3) of individuals are copied without modifications to the next generation (**elitism**).
- To avoid clustering of similar (image processing) solutions, a certain number N_r of randomly initialized individual ($N_r = N_e$) are added to the new generation (**genetic diversity**).
- The remaining members are build either by **recombination** (probability P_C =0.5) or **cloning** (probability P_{Cl} = 1-P_C).
- Suitable parents are selected by a common tournament-selection with multiple candidates (t=3).

Contribution from Peters & König acknowledged

Evolutionary Computation

- **Recombination (cross over):**
- „Single-Point-Crossover“
- The crossover point will be given by a random split through the sequence of image processing algorithms.
- Only one child is produced (sequence before the break from the first parent, sequence after the break from the second parent).



- Multi-point cross over possible

Contribution from Peters & König acknowledged

Optimization Techniques

- Random change of existing individuals can generate a leap in fitness
- Creation of new species, unsuccessful ones get extincted
- Bitflip in binary (GA) implementations, random de/increment in real-valued representations/optimization problems


Source: Andersen Consulting, 1995

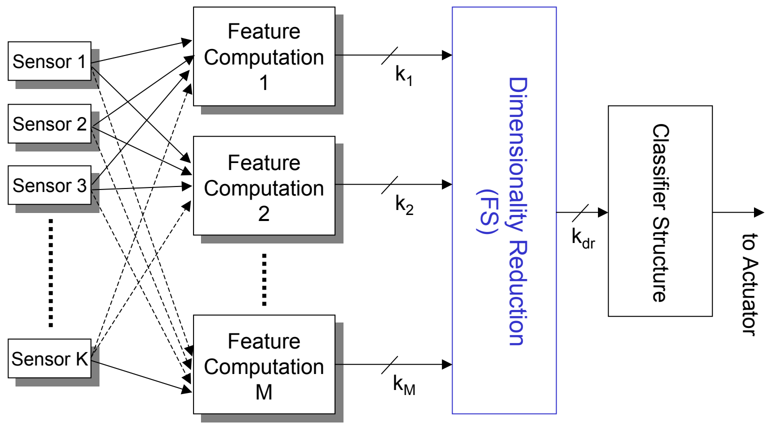
- Parameter mutation
- Node mutation (replace operator in processing chain)
- Topological/structural mutation (change graph structure of designed system)

- Numerous applications depending on problem encoding (geno/phenotype)
- Several (sensitive) parameters: population size, elitism, ... , mutation rate

Contribution from Peters & König acknowledged

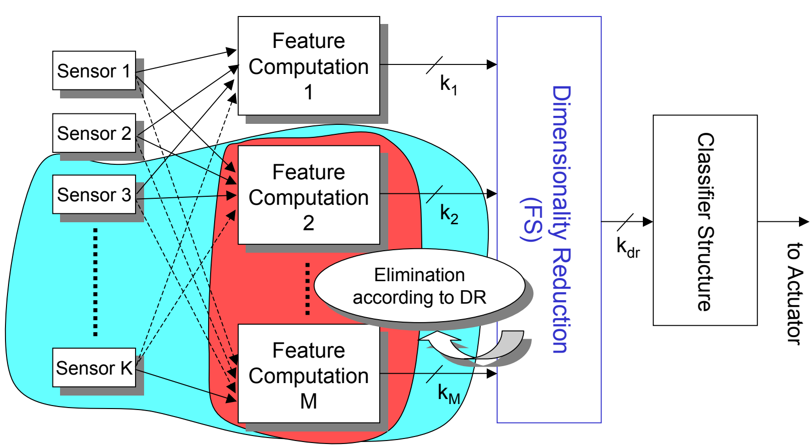
Application to AFS

- Automatic feature selection naturally offers itself for GA implementation
- Binary strings can perfectly represent the selection variables



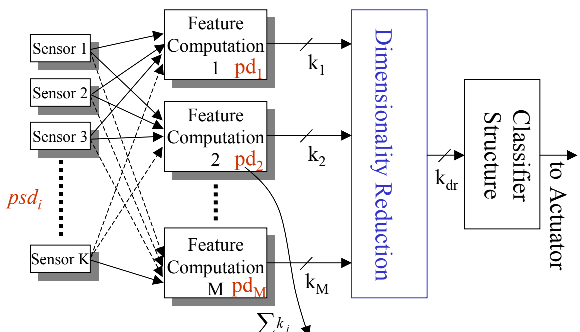
Application to AFS

- Structural simplification of first-cut design according to DR/AFS results:



Application to AFS

- Pursuing a single objective is often not sufficient/effective
- Several feature space criteria and constraints, e.g., explicit feature cost, can be combined in multiobjective approach



$$pd = \sum_{i=1}^{\sum k_j} pd_i * \delta(c_i, 1)$$

$$\mathbf{y} = \mathbf{f}(\chi) = (f_1(\chi), f_2(\chi), \ldots, f_n(\chi)) = (q_{si}(\chi), \ldots, pd(\chi), dr(\chi))$$

Application to AFS

- Majority of approaches do not consider cost (e.g., power dissipation) of features/grouped features within feature subset selection.
- One rare example by (Paclik & Duin 2002).
- Our approach incorporates inspiration from this work and evolutionary computation.

- Expression:

$$\boxed{K = q_{ov} + A \times \left(1 - \frac{C_S}{C_T}\right)}$$

where

A : weight of feature cost parameter
C_S : sum of active feature costs
C_T : sum of all feature costs

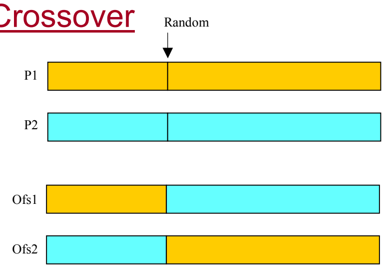
Contribution from Iswandy & König acknowledged

Application to AFS

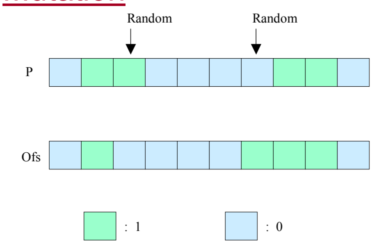
- AFS with optimization using Genetic Algorithm:

- Representation: binary (switch variables)
- One point Crossover; rate = 0.85
- Mutation; rate = 0.01
- Reproduction: best 10 % parents and offsprings

Crossover



Mutation



Contribution from Iswandy & König acknowledged

Sensor Signal Processing
Systematic Design

Application to AFS

- Consist of 4 features, 3 classes and 150 patterns (75 patterns for train and test).
- Dataset is repeated 4 times with different arbitrary cost assignment per feature.

Iris Dataset	Q_{ov}	Cost	Classification rate (%)	Selected Features
16 features	0.95417	156	90.333	All
FS (5 of 16)	0.98200	65	94.667	3, 4, 8, 12, 16
FS + Cost 1	0.98056	7	96.000	7, 12
FS + Cost 2	0.96318	16	93.333	2, 12

- Cost 1 : [2, 3, 14, 16, 10, 8, 4, 15, 6, 5, 20, 3, 3, 12, 18, 17]
- Cost 2 : [4, 1, 20, 18, 3, 4, 17, 20, 1, 2, 15, 15, 2, 3, 18, 22]

Contribution from Iswandy & König acknowledged

Application to AFS

- Eye-Image dataset from eye-tracker application study
- Consist of 3 Groups: Gabor (12), ELAC (13) and LOC (33 features)
- Each group has 2 classes and 133 patterns (72 patterns for train and 61 for test)

Eye-Image Dataset	Q_{ov}	Cost	Classification rate (%)	Selected Features
58 features	0.95482	165308	98.361	All
FS (17 of 58)	1.00	62713	98.361	1, 3, 8, 9, 11, 12, 14, 15, 16, 18, 21,28, 29, 34, 38, 54, 58
FS + Cost	0.99976	21675	98.361	12, 14, 18, 21, 37, 38, 39, 54

- Cost:

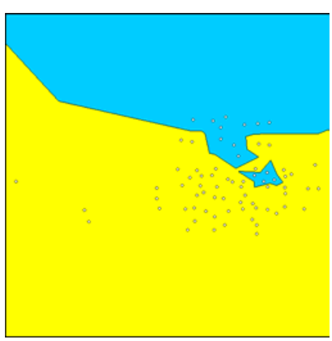
Gabor : 6358 / feature
ELAC : 3179 / feature
LOC : 1445 / feature

Contribution from Iswandy & König acknowledged

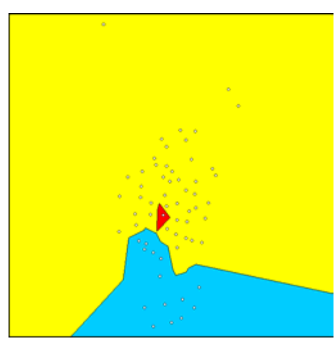
Application to AFS

- Visualization of Eye-Image Dataset (Gabor)
- Recognition result: 98.361% (1 error) without FS

Train



Test

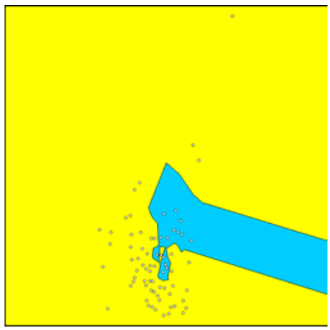


Contribution from Iswandy & König acknowledged

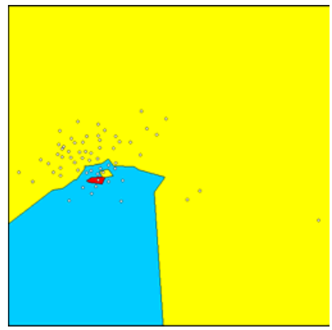
Application to AFS

- Visualization of Eye-Image Dataset (ELAC)
- Recognition result: 98.361 % (1 error) using FS

Train



Test

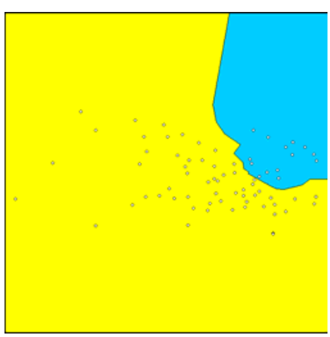


Contribution from Iswandy & König acknowledged

Application to AFS

- Visualization of Eye-Image Dataset (LOC)
- Recognition result: 98.361 % (1 error) using feature selection with acquisition cost

Train



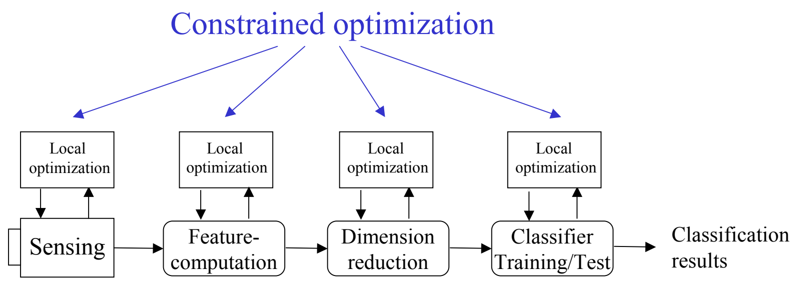
Test

Contribution from Iswandy & König acknowledged

- Generalization and stability are open issues in AFS

System Embedding

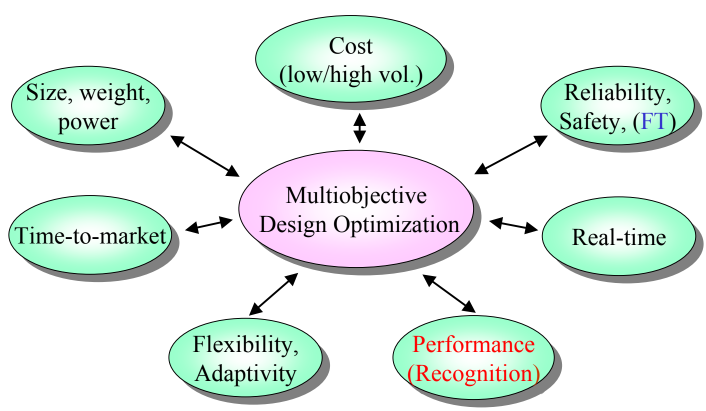
- In the early phase of development, special hardware was common for, e.g., real-time image processing systems
- State-of-the-art hardware is getting more and more powerful, so that the majority of applications can run on same platform, they were designed on
- For special requirements (size, power, ...) this might not be feasible
- Special purpose deployment hardware platforms impose additional constraints:



Sensor Signal Processing
Systematic Design

System Embedding

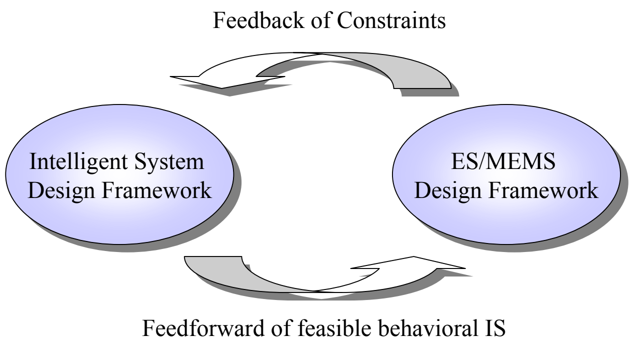
- Reduced numerical precision, parasitics, or other non-ideal properties can affect deployed system performance !
- These constraints must be included/met in training (HW-in-the-loop-learning)



System Embedding

Sensor Signal Processing
Systematic Design

- Already in the design phase or in the application-specific configuration phase, the existing constraints have to be added into the optimization
- Collective adaptation principles can compensate general as well as instance specific deviations (manufacturing tolerance, yield)

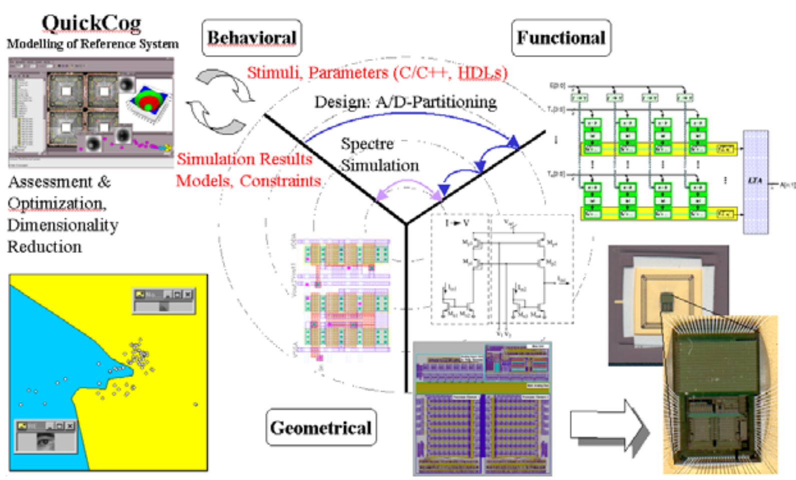


- Requires combination of different design abstraction levels & tools

System Embedding

Sensor Signal Processing
Systematic Design

- Example for the holistic design of low-power sensing & recognition systems on circuit level (GAME-project, DFG SPP 1076 VIVA)



Summary

- The chapter introduced to the principles of efficient and (semi-) automated design/configuration of multi-sensor systems for recognition tasks
- Open-loop and closed-loop optimization approaches were discussed
- The role of evolutionary computation for that aim was pointed out
- Feature selection was chosen as an application example, extending the concept to multi objective optimization including explicit feature cost
- The concepts were extended to the issue of deployment onto dedicated hardware platforms, e.g., MEMS or embedded systems, under ressource constraints
- Additional assessment criteria and optimization techniques, e.g., Pareto approaches and particle swarm optimization can be employed
- Efficient tools are required in addition to the principle design methodology