Multi-Objective Intrinsic Evolution of Embedded Systems (MOVES)

Paul Kaufmann, Marco Platzner

Computer Engineering Group

University of Paderborn

{paulk, platzner}@uni-paderborn.de

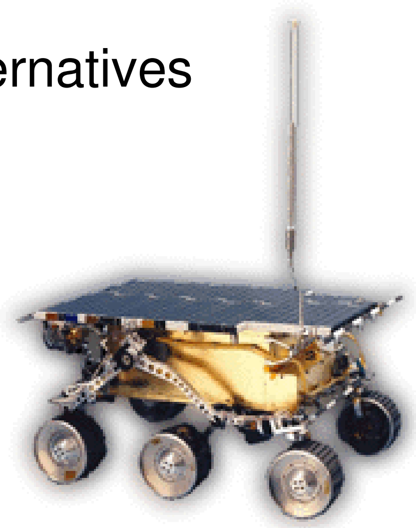
Outline

- motivation/architecture
- Δ to last status meeting
- tackling scalability
- application: prosthetic hand control
- summary

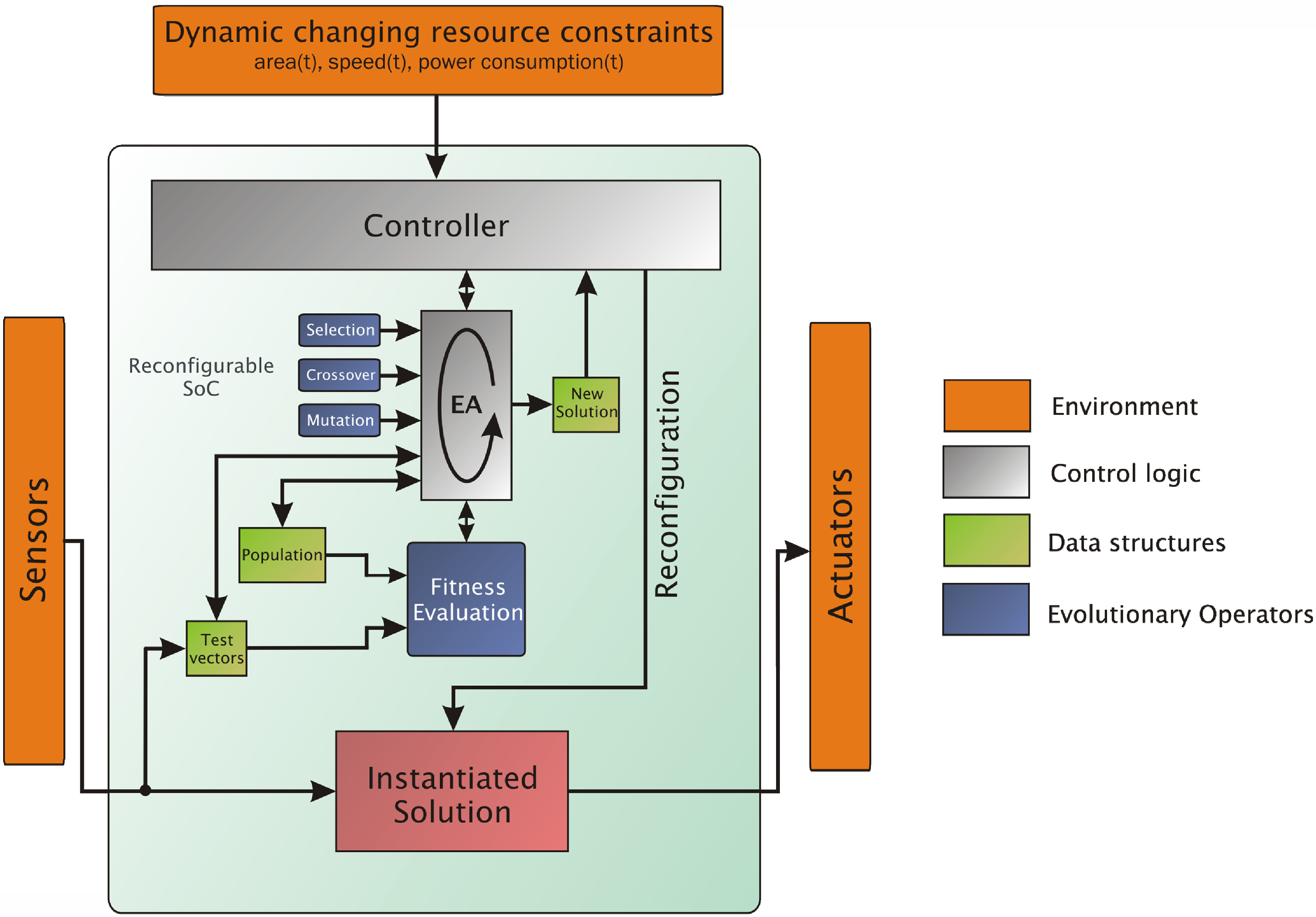
Motivation / Vision

- investigate intrinsic evolution as a mechanism to achieve self-adaptation and –optimization for autonomous embedded systems

- an embedded system ...
  - adapts to slow changes by simulated evolution
    - typically, change of environment
  - adapts to radical changes by switching to pre-evolved alternatives
    - typically, change in computational resources
  - requires intrinsic evolution for autonomous operation



Architecture



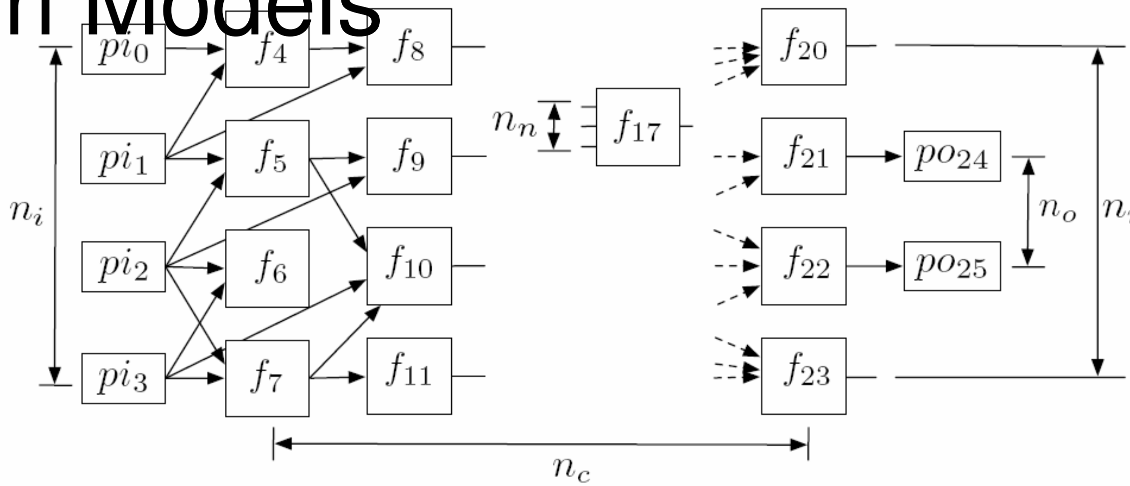
Δ to Last Status Meeting

- last status meeting
  - hardware representation models
    - cartesian genetic programming (CGP), ECGP
  - multi-objective evolutionary optimizers (MOEAs) SPEA2, TSPEA2
    - evolved arithmetic circuits and hashing functions
  - evaluation framework
    - MOVES toolbox for experimenting with evolvable hardware (EHW)

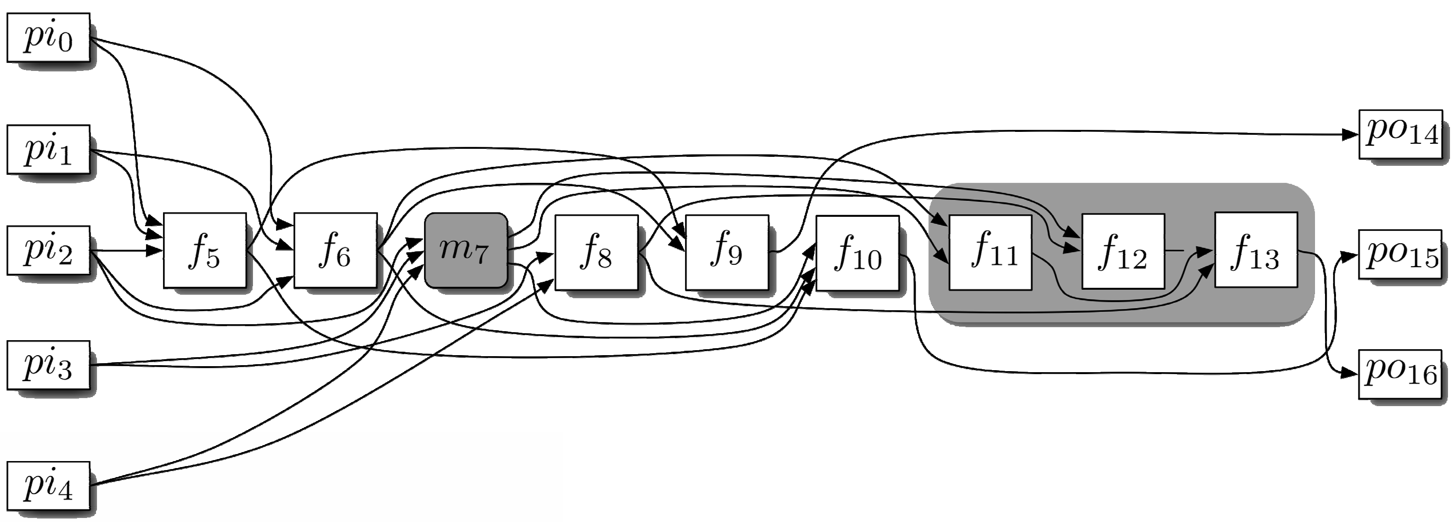
- new work done
  - evolutionary algorithms
    - tackling scalability [Kaufmann & Platzner, GECCO '08]
    - comparing GA with MOEAs [Knieper et al., BICC '08]
  - reconfigurable SoC
    - hardware accelerator for k-NN thinning [Schumacher et al., ERSA '08]
  - application examples
    - prosthetic hand controllers [Glette et al., AHS '08] [Glette et al., ICES '08]

Hardware Representation Models

- cartesian genetic program
  [Miller & Thomson, EuroGP '00]

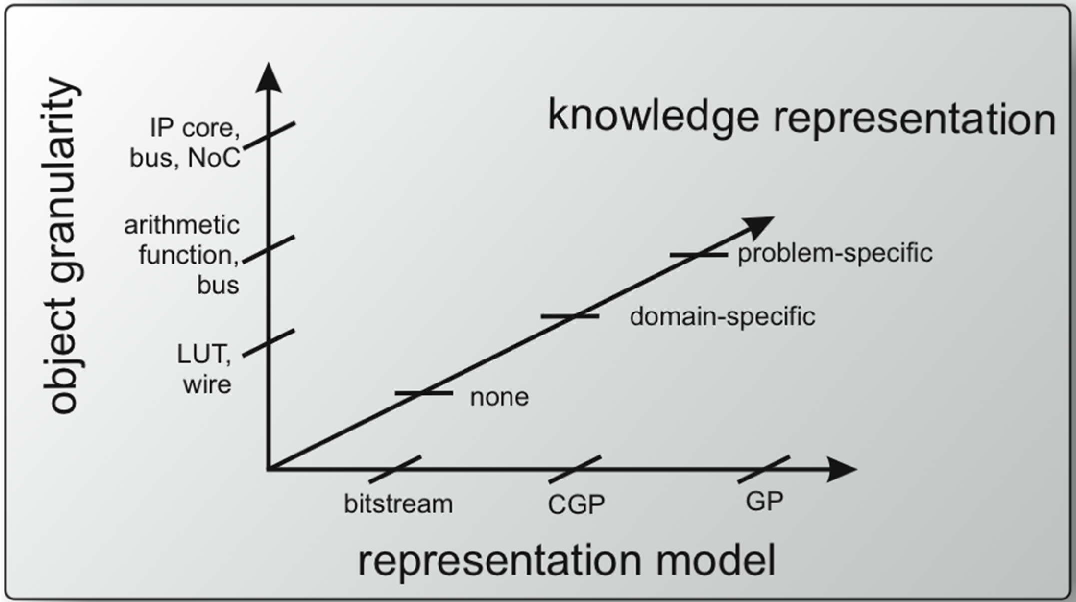


- ECGP [Walker & Miller, EuroGP '04]
  - single row of functional blocks, 1+4 evolutionary strategy
  - automated module creation: compress / expand operators
    - nodes between two randomly chosen nodes define a new module



Scalability

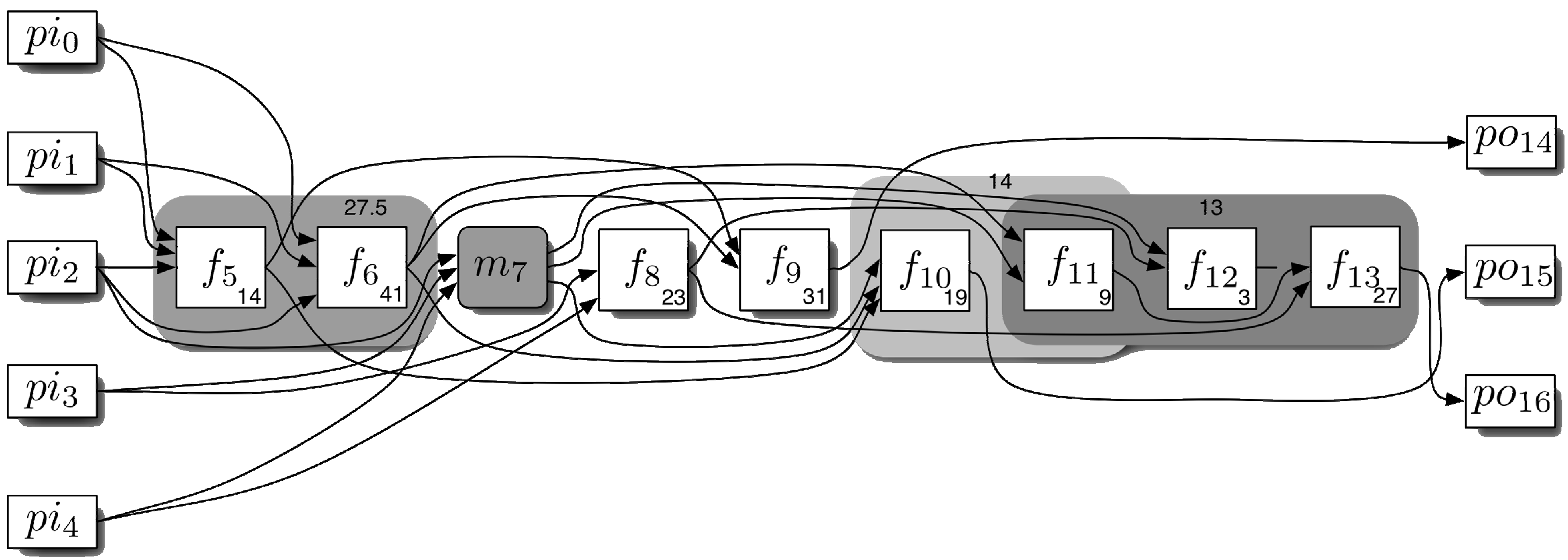
- scalability can be tackled along three dimensions
  - object granularity
  - knowledge representation
  - representation model



- new automated module creation and propagation approaches
  - instead of randomly aggregating nodes to modules, consider
    - age-based modules (inspired by ‘organs’)
    - cone-based modules (inspired by classical circuit synthesis)

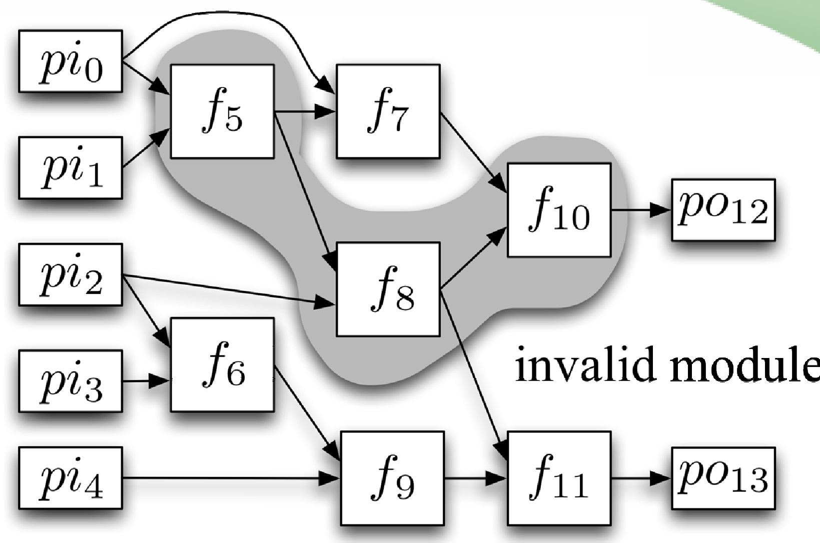
Age-based Module Creation

- prefer aged nodes over randomly selected ones
  - each primitive node (function and inputs) is assigned an age which equals the number of generations it is untouched
  - module creation: tournament selection based on the mean age of module candidates

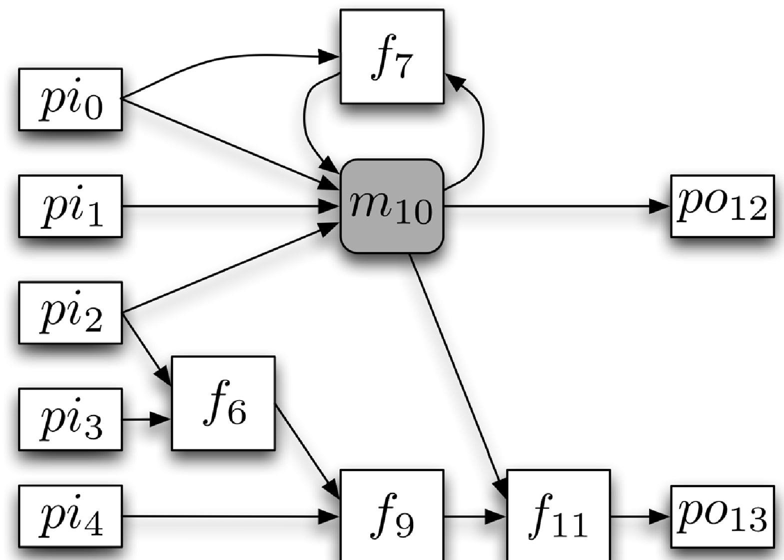


Cone-based Module Creation

- modules have to be cones
  - for every node in the module there exists a path to the module root which is entirely in the cone
  - considers connections; aged-based and random module creation aggregate primitive nodes without considering connections
- avoid reconvergent paths
  - modules must be convex sub-graphs



(a)

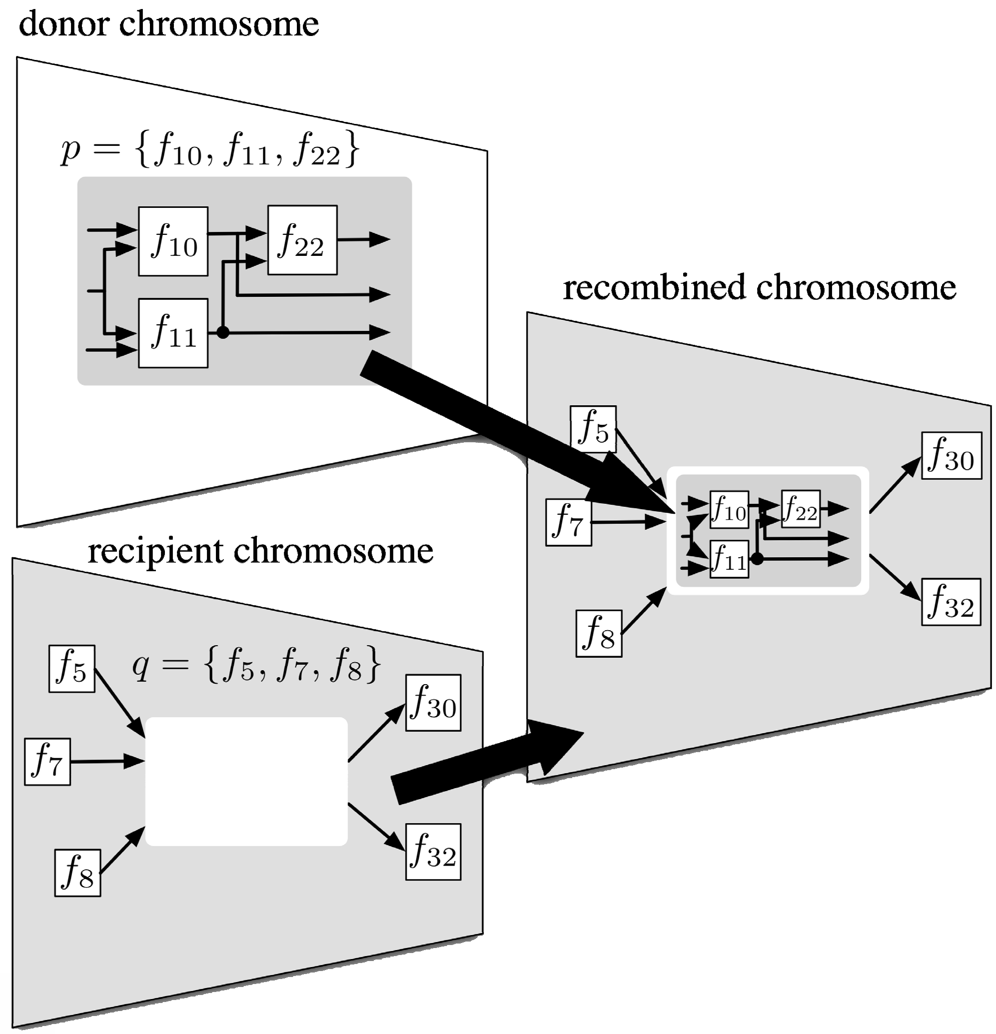


(b)

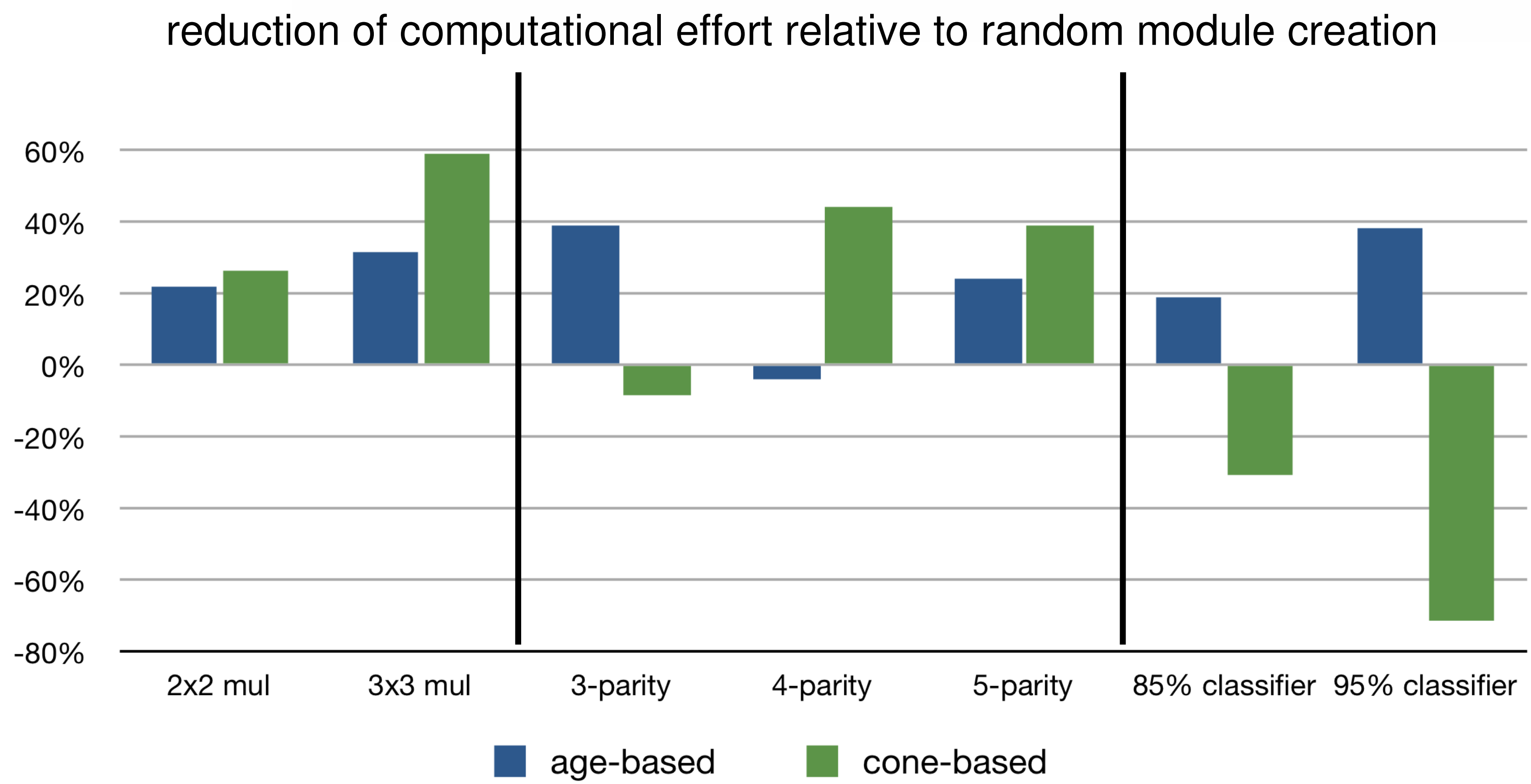
Cone-based Crossover

- propagate modules (cones) between chromosomes
  - use genetic algorithm instead of evolutionary strategy

- preserve routing between module and chromosome



Results: Module Creation



- age-based module creation better for six out of seven benchmarks
- cone-based module creation better for multipliers
  - multipliers are more complex and more regularly structured than the other benchmarks

Results: Cone-based Crossover

reduction of computational effort relative to 1+4 ES,
GA with population sizes of 5 and 50

	2x2 mul	3x3 mul	3-parity	4-parity	85% emg classifier	95% emg classifier
GA-5	**3.77%**	**71.51%**	-370.94%	-1,217.21%	-8.76%	-13.10%
GA-50	-53.99%	-341.88%	-130.39%	-1,256.51%	-57.86%	-36.39%

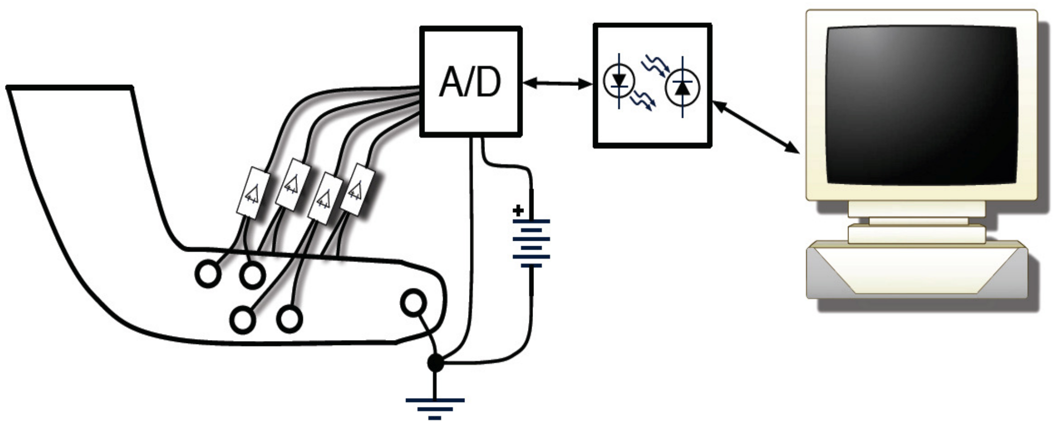
- GA-5 outperforms ES only for multipliers
- GA-50's increased population size not beneficial
  - potential for recombination does not outweigh the larger computational effort per generation

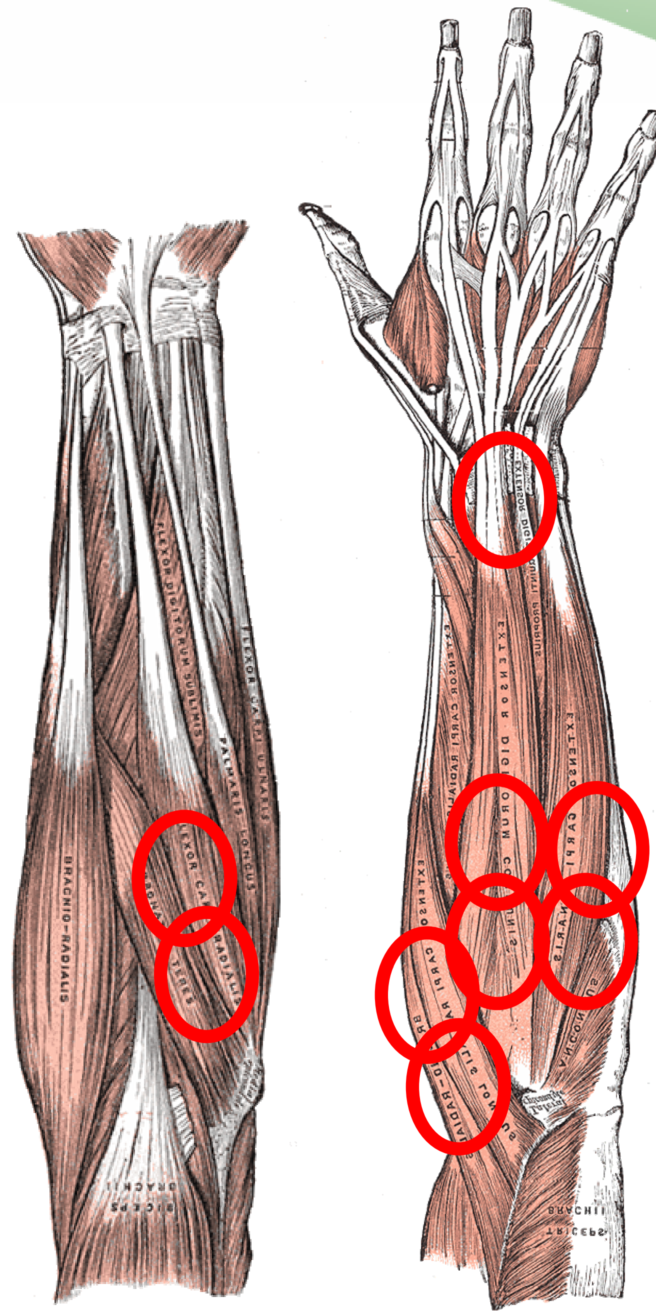
Application: Prosthetic Hand Control

- classification of electromyographic (EMG) signals
  - evolvable hardware previously applied for this task [Kajitani et al., ICES '98] [Torresen, ICES '01]
- investigation of different EHW approaches
  - collaboration with Uni Oslo [Glette et al., ICES '08]
- comparison of EHW with conventional classifiers
  - collaboration with Uni Passau and Uni Oslo [Glette et al., AHS '08]
- towards 'real-world' application
  - longer-term measurements, measurements on amputees
  - collaboration with Winkler Orthopädietechnik, Minden

System Setup

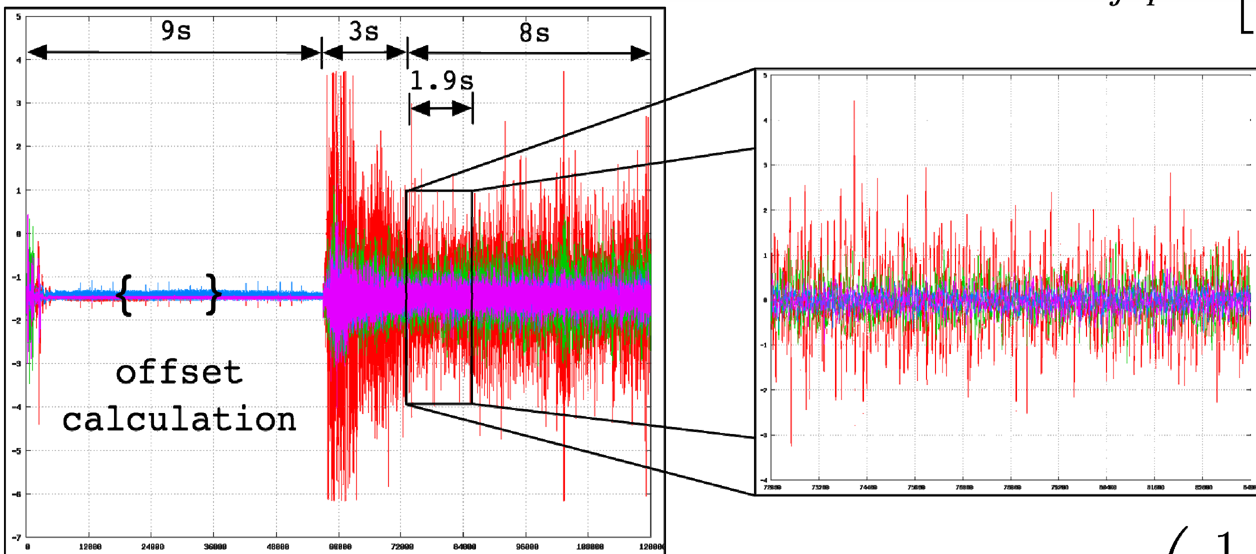
- four EMG electrode pairs placed on forearm
- signals recorded and classified on a PC

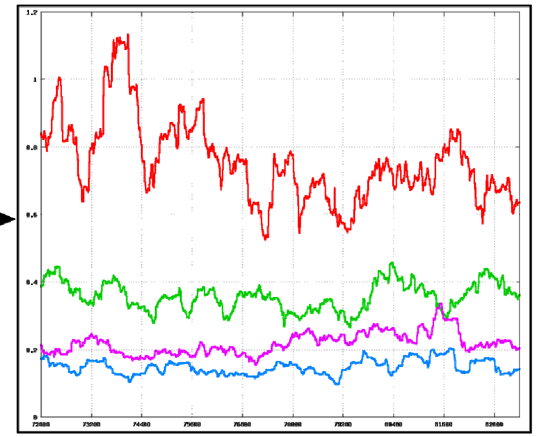




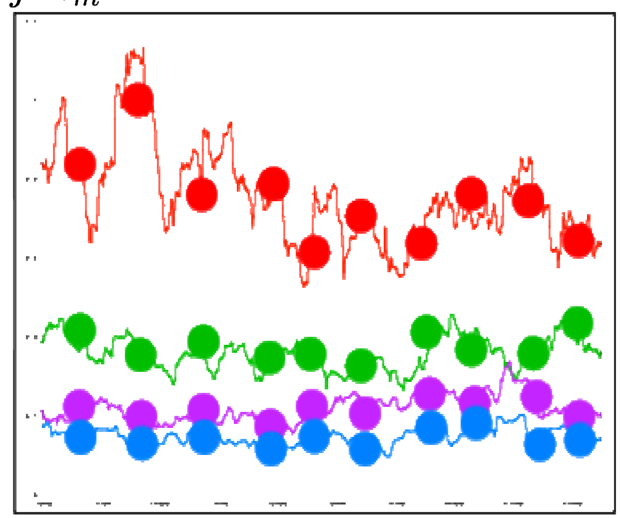
Signal Preprocessing and Feature Extraction

$$d'_{jkp} = \left[\frac{1}{w_s}\sum_{i=j}^{j+w_s-1}(d_{ikp} - o_{kp})^2\right]^{\frac{1}{2}}$$





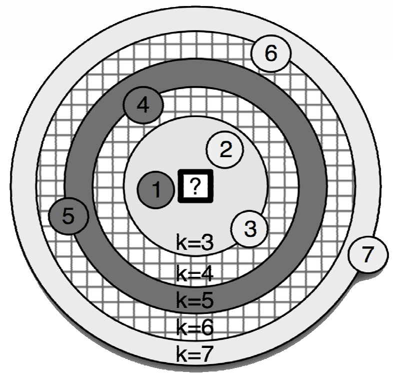
$$f_{l_m kp} = -\log\left(\frac{1}{w_f}\sum_{j=l_m}^{l_m+w_f-1} d'_{jkp}\right)$$



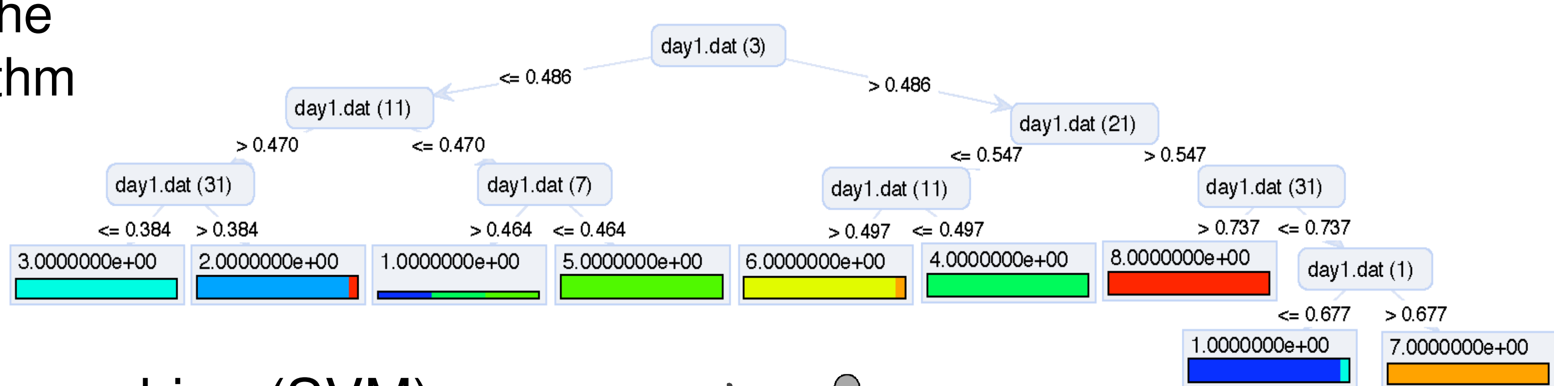
- pick first 1.9 s of the "steady state" phase
- smooth signal by RMS
- extract 10 features per channel
- 40 features for a single contraction (movement)

Conventional Classifiers

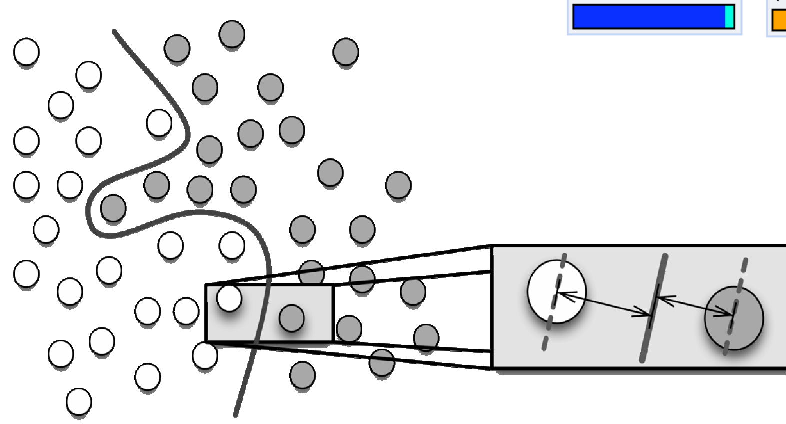
- kth-nearest neighbor (kNN)
  - baseline method, k=5



- decision tree (DT)
  - built from if-then rules, each leaf represents a category decision
  - trained by the C4.5 algorithm

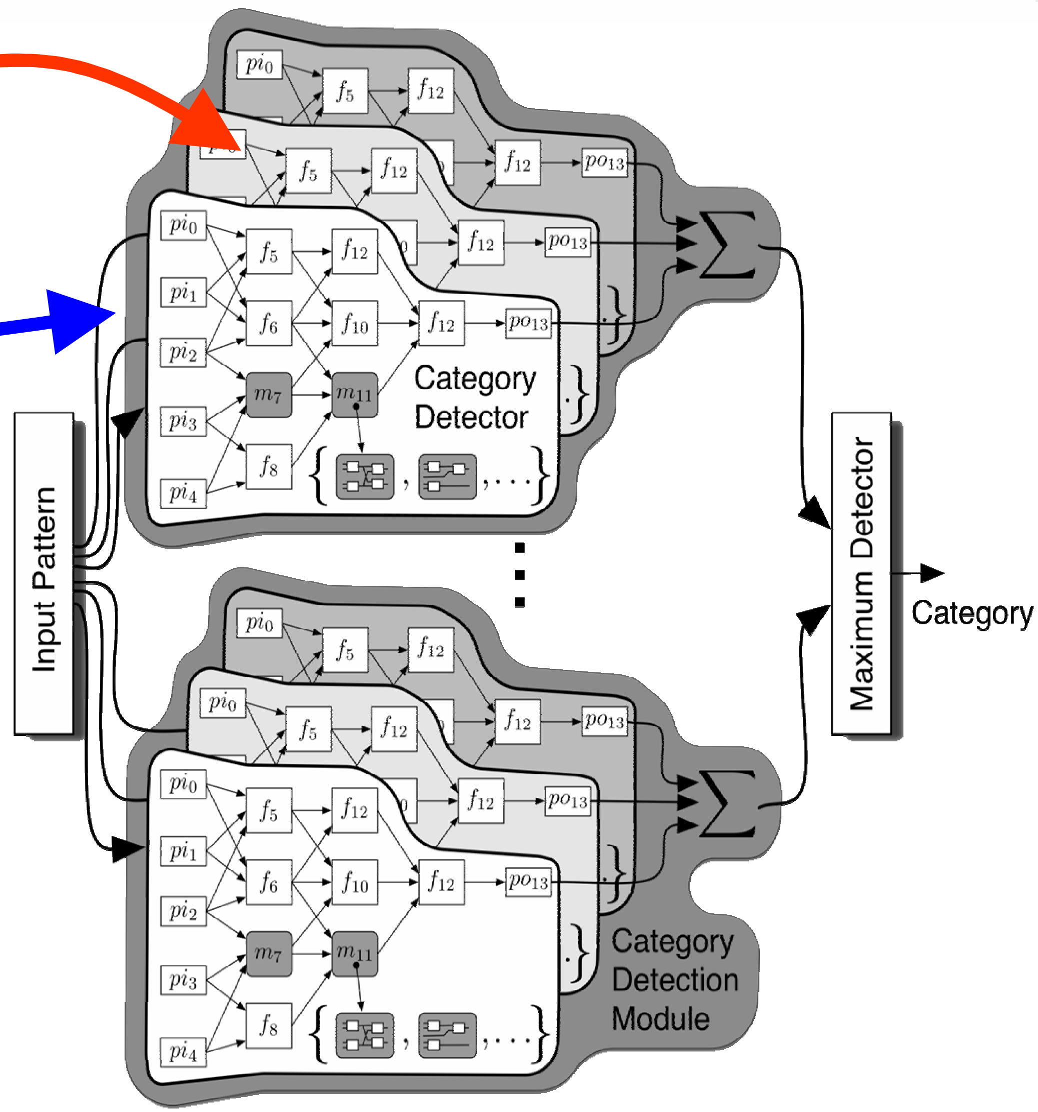


- support vector machine (SVM)
  - regarded as one of the most powerful classification methods today



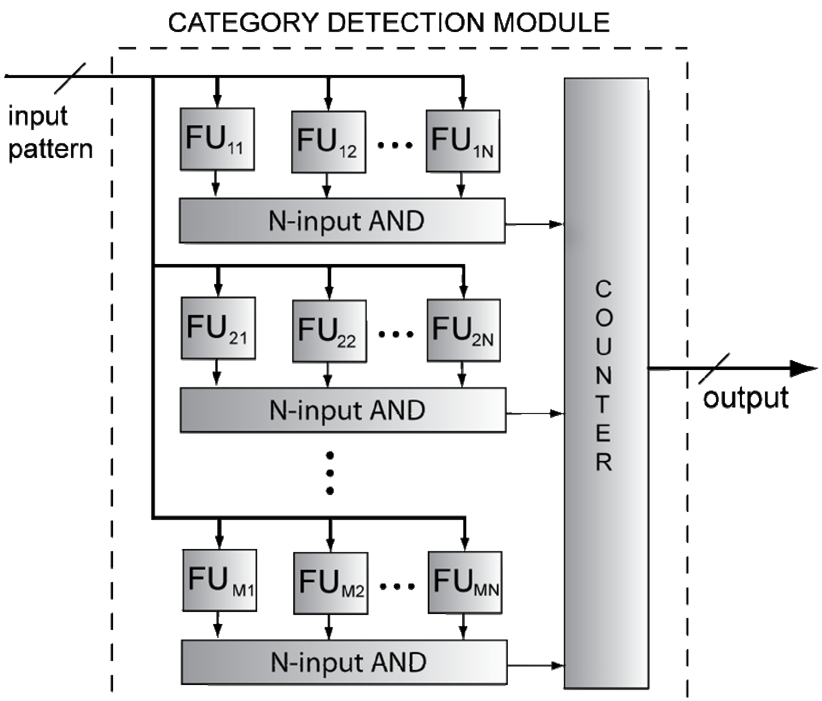
EHW1: ECGP + age-based Module Creation

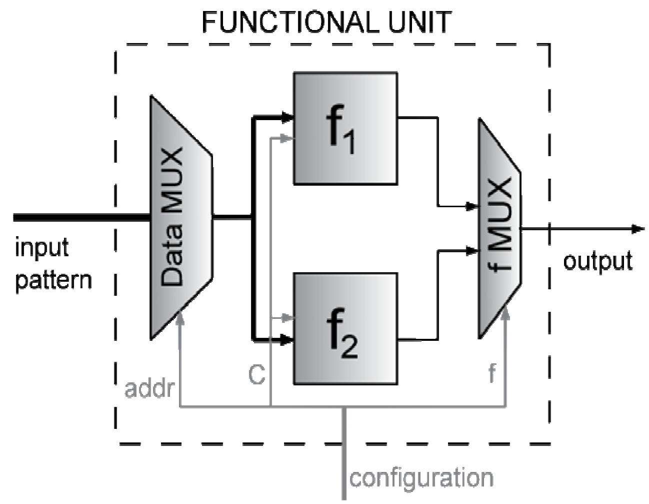
- multiple category detectors (ECGP chromosomes) form a category detection module
- maximum detector selects the category with the most activations



EHW2: Functional Unit Row Architecture

- previously applied to face image and sonar signal recognition [Glette et al., AHS '07] [Glette et al., ICES '07]
- architecture tailored to classification tasks
- implemented as virtual reconfigurable circuit for fast online evolution

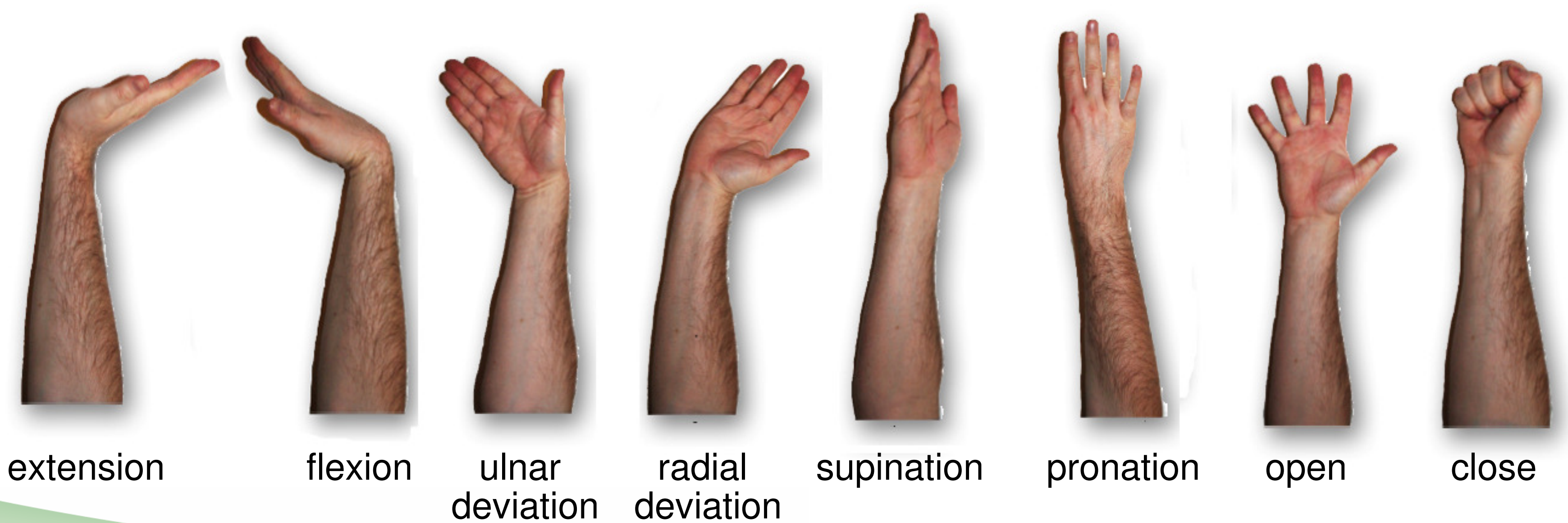




f	Description	Function
0	Greater than	O = 1 if I > C, else 0
1	Less than or equal	O = 1 if I ≤ C, else 0

Experimental Setup

- EMG signal sampled from one subject on three consecutive days, each with 20 iterations of 8 movements
- leave-one-out cross validation
  - experiments for separate days: day1, day2, day3
  - combining all days: day1-3
- training on two days and leaving one day out for testing: 2of3
  - believed to be a more realistic scenario



Example Results: Training and Test Errors (over all movements)

TEST ERRORS (GENERALIZATION)

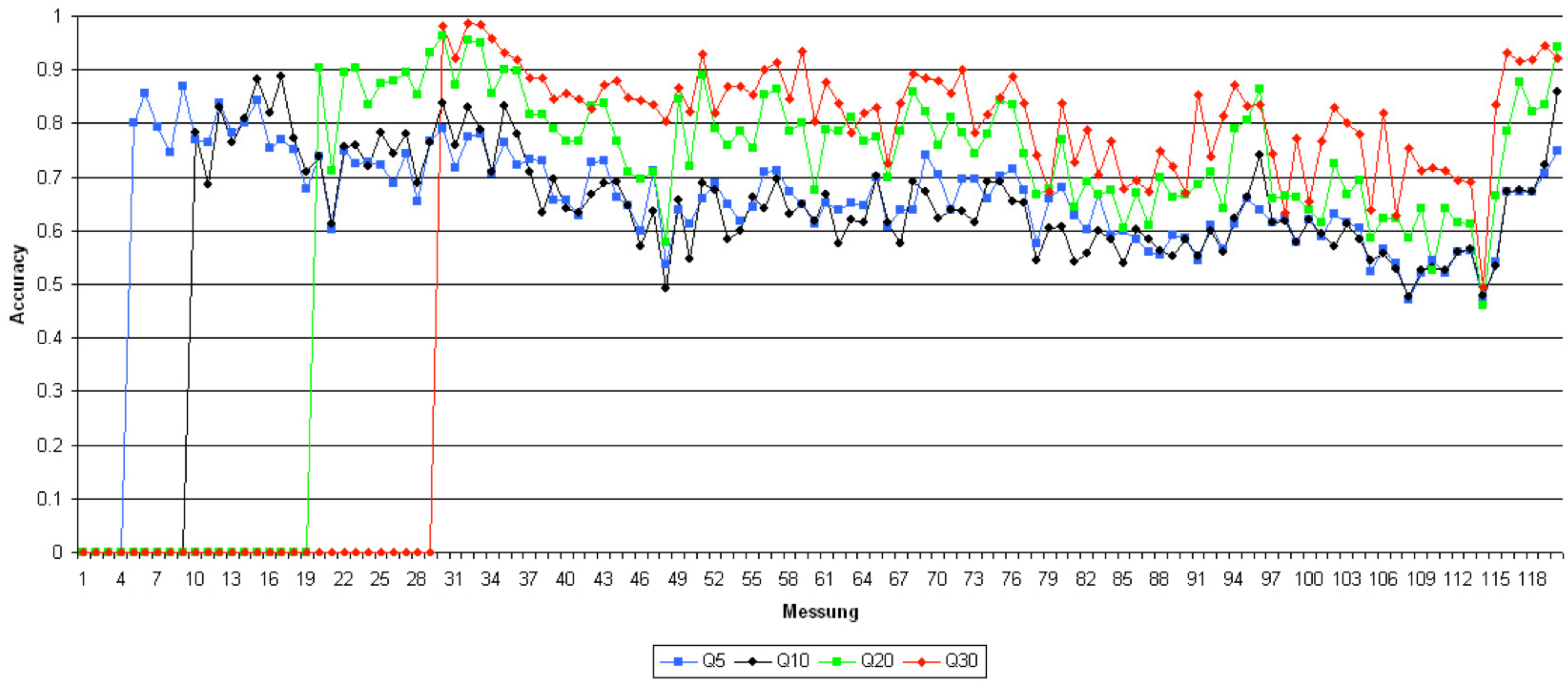
	Day1	*Day2*	*Day3*	*Day1–3*	*2of3*
kNN	3.5 %	4.6 %	4.6 %	4.5 %	5.6 %
DT	9.7 %	11.3 %	10.5 %	9.0 %	15.9 %
SVM	4.2 %	4.0 %	2.6 %	4.5 %	5.4 %
EHW1	9.8 %	4.0 %	5.3 %	9.0 %	10.6 %
EHW2	9.0 %	4.6 %	4.0 %	4.9 %	8.4 %

TRAINING ERRORS (APPROXIMATION)

	Day1	*Day2*	*Day3*	*Day1–3*	*2of3*
kNN	2.8 %	2.7 %	4.0 %	3.6 %	3.4 %
DT	1.4 %	2.0 %	1.3 %	1.8 %	2.0 %
SVM	1.4 %	0.7 %	2.0 %	3.6 %	2.6 %
EHW1	5.0 %	5.0 %	5.0 %	5.0 %	5.0 %
EHW2	1.9 %	0.8 %	0.0 %	3.6 %	2.6 %

Longer-term Measurements

- 121 measurements over 22 days, at different day times
  - classify 11 movements with SVM; 5/10/20/30 data sets used for training
  - classify every 10 ms, keep 15 most recent results in queue and vote



→ potential for continous (self-)adaptation

Summary

- Δ to last status meeting
  - evolutionary algorithms
    - tackling scalability [Kaufmann & Platzner, GECCO '08]
    - comparing GA with MOEAs [Knieper et al., BICC '08]
  - reconfigurable SoC
    - hardware accelerator for k-NN thinning [Schumacher et al., ERSA '08]
  - application examples
    - prosthetic hand controllers [Glette et al., AHS '08] [Glette et al., ICES '08]
- collaborations
  - Bernhard Sick & Thiemo Gruber, University of Passau
  - Jim Torresen & Kyrre Glette, University of Oslo
  - Winkler Orthopädietechnik, Minden
- tutorial on Evolvable Hardware @ ARCS'08
  - introduction and experiments with MOVES toolbox
  - together with Kyrre Glette and Jim Torresen

Thank you for your attention!