Microcontroller Implementation of a Multi Objective Genetic Algorithm for Real-Time Intelligent Control

Martin Dendaluce
Juan José Valera
Vicente Gómez-Garay
Eloy Irigoyen
Ekaitz Larzabal

Agenda

- Introduction and Motivation
- Proposed Hybrid Control Scheme
- Implementation
- Results
- Conclusions and Future Work

Introduction and Motivation

- Soft-Computing → excellent for complex problems [1]
  - Fuzzy Logic
  - Neurocomputing

  diverse embedded real-time implementations

  - Evolutionary Computation

  few on-target applications

  - … [2]
- Research line
  - Hybridation of Intelligent Techniques [3]
  - Implementation on high performance Platforms [4]
- Interesting: lower level industrial approach

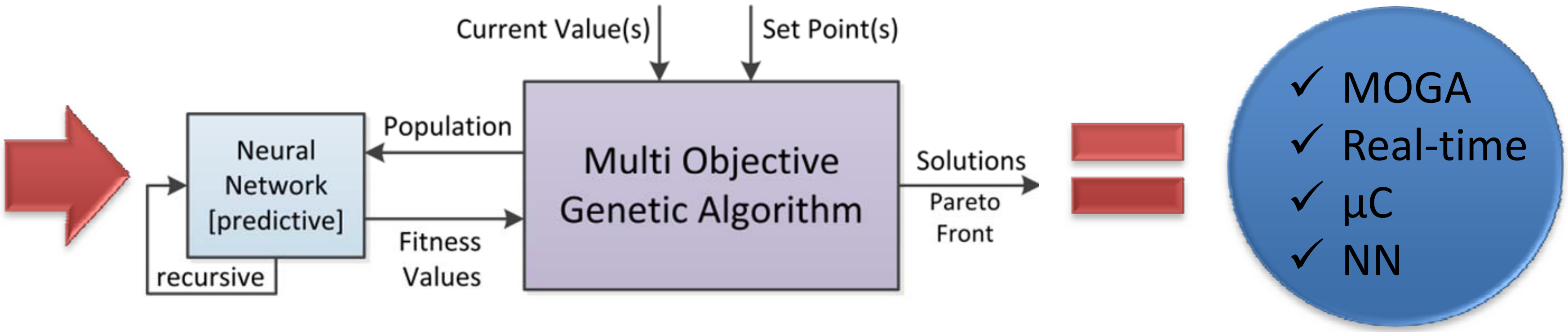
References: 1) Rudas and Fodor, 2008; 2) Palit and Popovic, 2005; 3) Valera et al., 2012; 4) Larzabal et al., 2013

Proposed Hybrid Control Scheme

- MOGA for optimization of control actions: NSGA-II [5]
  - Good for complex, non-linear, non-convex problems
  - Computational cost = $O(MN^2)G$

O: platform/implementation constant
M: objective number
N: population size
G: generations

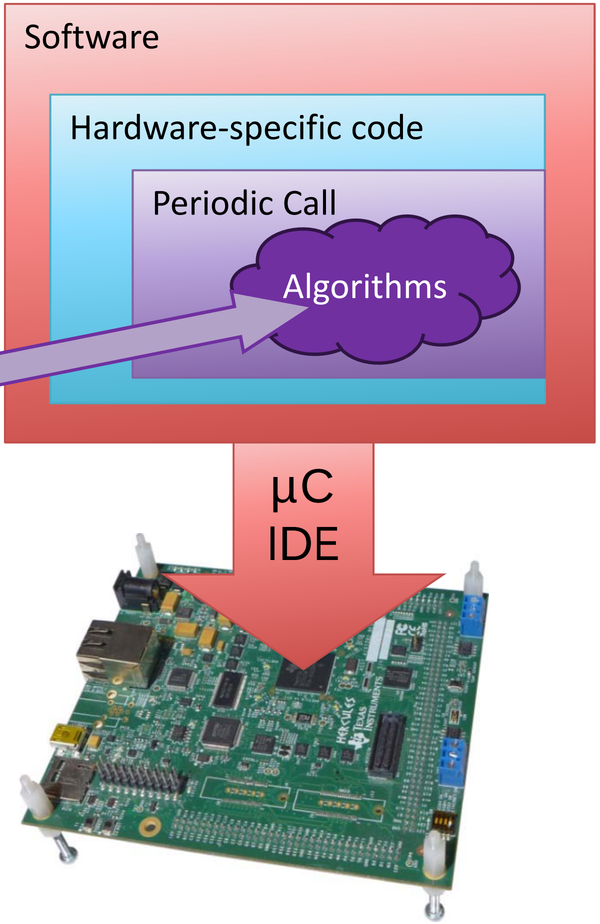
- Neural Network for Fitness evaluation
  - Modelling of nonlinear multivariable systems



References: 5) Deb, 2002

Implementation

- ARM-based redundant 180MHz 32bit µC
- Flexible software structure
  - Coded in C and speed-optimized
  - NSGA2-II Deb's Code adaptation
  - NN: Simulink™ code-generation
- Additional HiL setup
  - For future algorithm validations



Implementation

- Tested NSGA-II setups

	Population	Generations	Chromosome	Objectives
«Small»	15	20 / 24	6 / 10	2
«Big»	60	60	6	2

- Used Fitness functions
  - Algebraic functions (simple and heavy) → generic
  - Nonlinear differential model (horizon=5)
  - NARX Neural Network (horizon=5)
    - Main layer: 8 neurons, hidden layer: 12 neurons

→ Twin-Rotor system

Results

- Small MOGA < 28ms , Big MOGA < 1s
- < 100ms ← Pop.=15, Gener.=24, NN (Horizon=5)
- Code size: ~79KB (Algebraic) , ~105KB (NN)
- RAM: ~12/85KB (Small/Big MOGA), ~14KB (NN)
- NN execution ≈ 34.7μs x 5
- Maximum population = 113 (256KB-RAM limit)

- Extended expression for computation cost proposed:

$$O(MN^2)G \rightarrow T = C'(MN^2)G + C'(NF)G = C'[N[MN + F)]G \quad \rightarrow C' \approx 0.0022$$

Platform computation constant Fitness evaluation Cost

Results

- Small MOGA < 28ms , Big MOGA < 1s
- < 100ms ← Pop.=15, Gener.=24, NN (Horizon=5)
- Code size: ~79KB (Algebraic) , ~105KB (NN)
- RAM: ~12/85KB (Small/Big MOGA), ~14KB (NN)
- NN execution ≈ 34.7µs x 5
- Maximum population = 113 (256KB-RAM limit)

	Big MOGA test		Small MOGA test		Twin Rotor test	
Fitness function	Simple (1)	Heavy (2)	Simple (1)	heavy (2)	Nonlinear Model	NN
Objectives	2	2	2	2	2	2
Population	60	60	15	15	15	15
Generations	60	60	20	20	24	24
Chromosome size	6	6	6	6	10	10
Prediction horizon	-	-	-	-	5	5
Constraints	-	-	-	-	-	-
Code size	79.0 KB	79.7KB	78.4 KB	79.2 KB	101.8 KB	104.3 KB
RAM occupation	85.5 KB	85.5 KB	11.6 KB	11.6 KB	14.9 KB	13.5 KB
T_{cycle} Avg./Worst [ms]	922/970	998/1085	21.5/24.2	28.1/32.3	74.6/82.1	88.1/96.1
C factor (see section 2)	0.002134	0.002310	0.002389	0.003122	0.006907	0.008157
t_{eval} Average	0.335 µs	21.8 µs	0.335 µs	21.8 µs	136.2 µs	173.5 µs
C' factor (see eq. 3)	0.002130	0.002130	0.002378	0.002378	0.002371	0.002371

Conclusions and Future Work

- Feasibility of MOGA + NN on reasonable µC
- Relatively low cycle times possible
- Great potential for embedded intelligen control
- Degrees of freedom for future research
  - Algorithm development
  - Optimization and use of other platforms (DSP, FPGA....)
- Future work
  - Algorithm development with HiL
  - Hybrid hardware implementation: Processor + FPGA

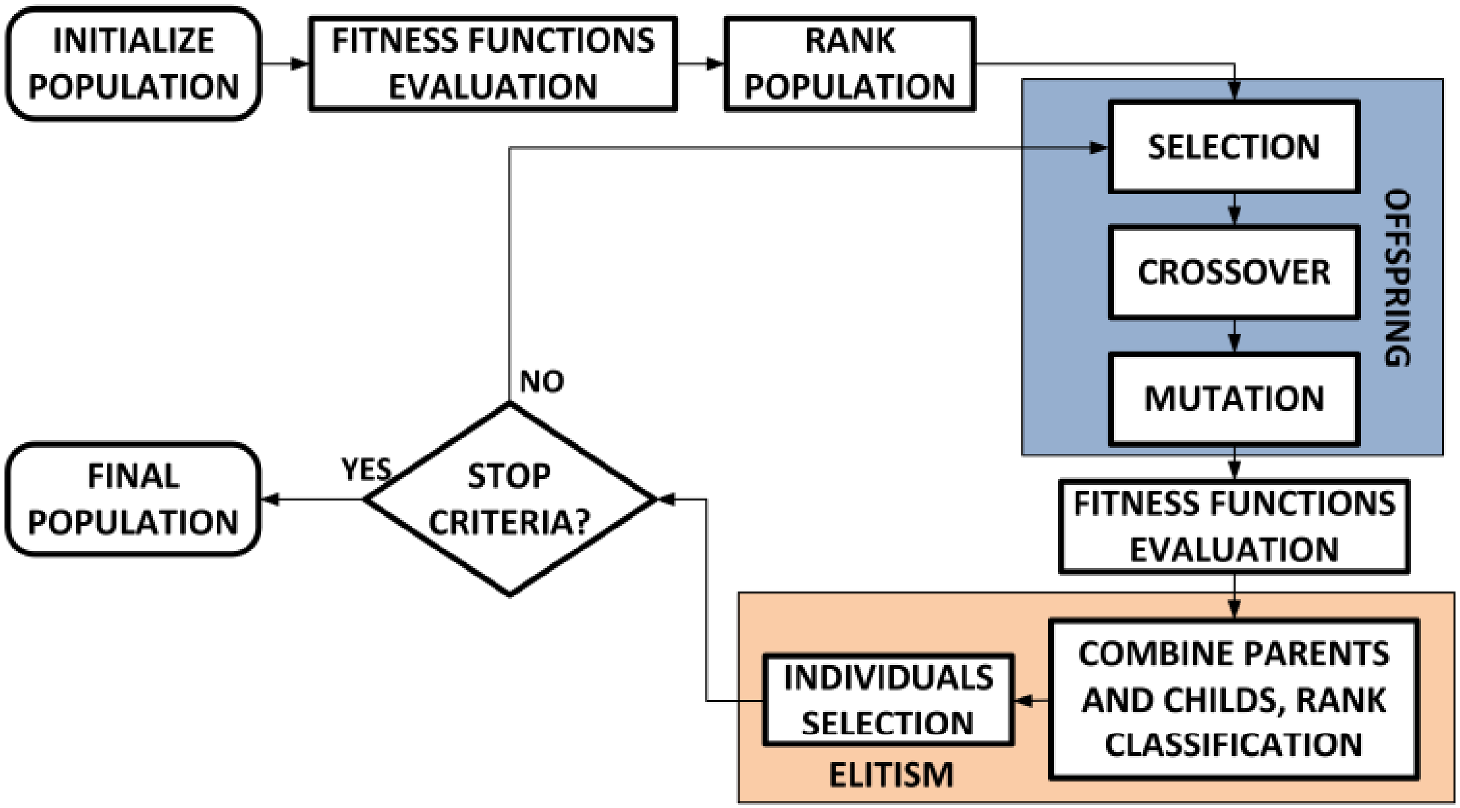
End

Thank you for your attention

Questions?

Contact for further questions: martin@dendaluze.com

NSGA-II Diagram



Reference: Martin Dendaluce et al. 2013

HiL testing setup