ISD Programme 2009:

Artificial intelligence 2

Lecture 6, 2010-11-25

Applications of EAs

Applications of EAs

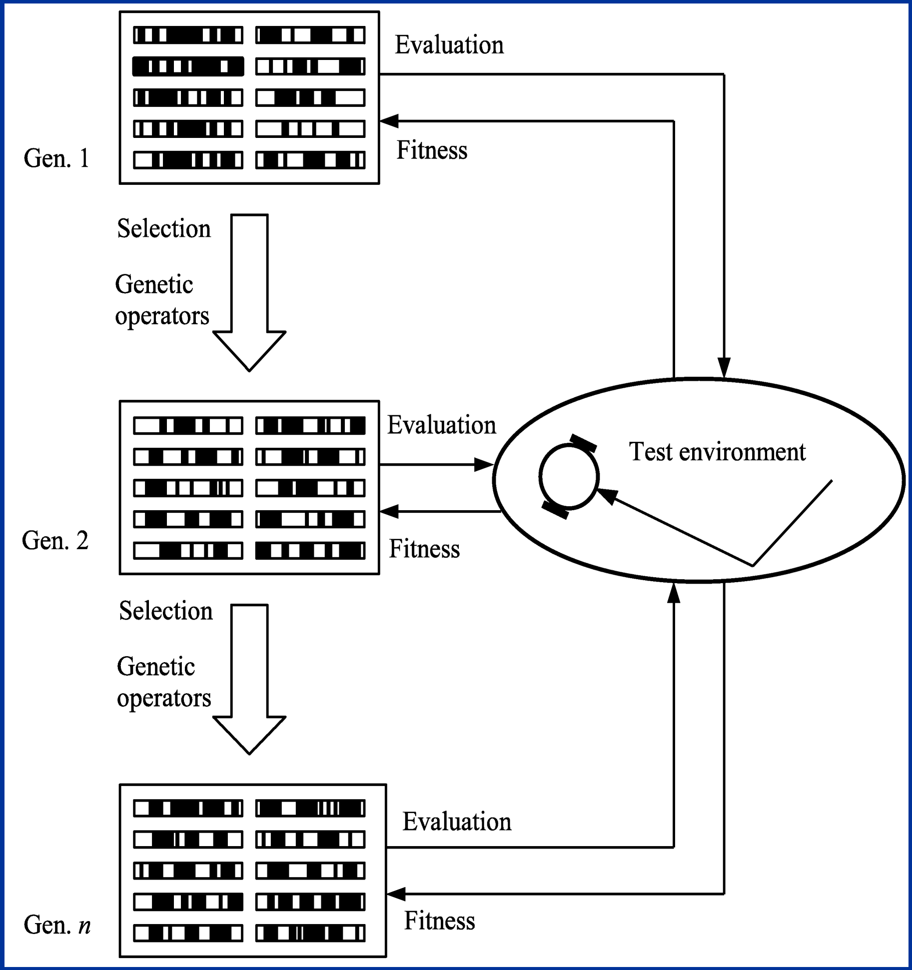
- (1) *Evolution of efficient gaits with an autonomous biped robot using visual feedback.*
- (2) *Evolutionary optimization of bipedal gait in a physical robot.*
- (3) *Structural evolution of central pattern generators for bipedal walking in 3D simulation.*
- (4) *A general-purpose transportation robot - a summary of work in progress.* ***Mandatory paper!***
- (5) *Evolving 3D model interpretation of images using graphics hardware.*
- (6) *Optimization of brake utilization for heavy-duty trucks.*
- (7) *Driver sleepiness detection - DROWSI.*

Evolutionary robotics

- **ER is a subfield of robotics, in which evolutionary algorithms (EAs) are used for generating robotic brains or bodies, or both.**

Evolutionary robotics

- Initial population generated randomly.
- A robotic control system decoded from each chromosome.
- The resulting control system is then evaluated in a robot, in a given environment
- The fittest individuals reproduce.



Application 1

Online optimization of gaits in a real, physical robot I

Evolution of efficient gaits with an autonomous biped robot using visual feedback

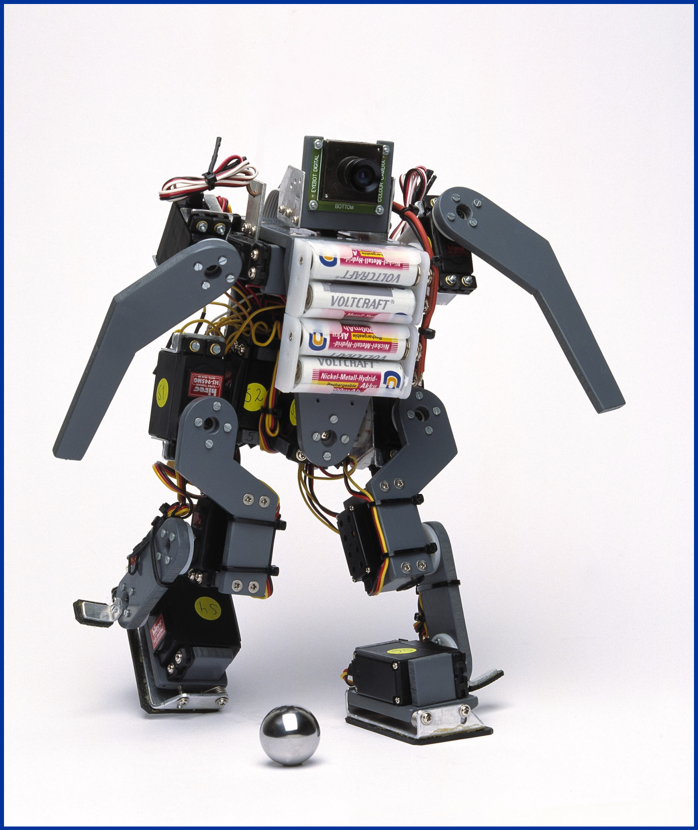
- K. Wolff, P. Nordin

Chalmers University of Technology,
Göteborg, Sweden

Application 1

The robot

- Humanoid robot Elvina
  - 28 cm tall
  - fully autonomous robot
  - vision and proximity
  - 14 dof



Application 1

Experiment set up

- Goal:
  - optimize the robots gait, make it walk faster, straigther, and in a more robost way, than it previously did.

Application 1

Representation

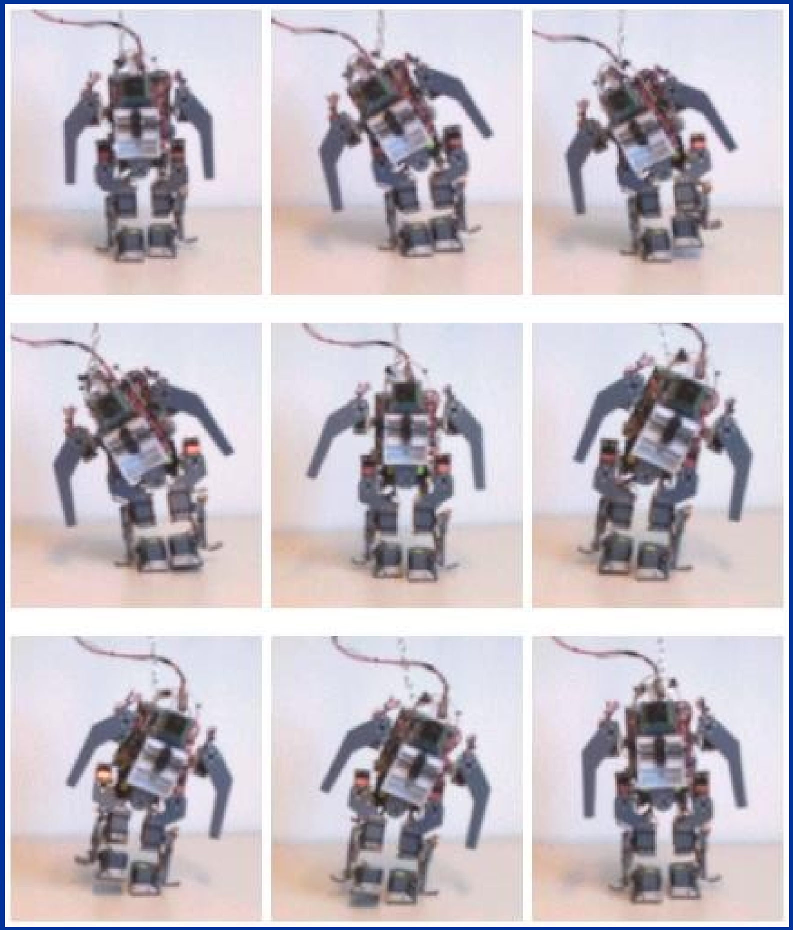
- A chromosome, specifing a gait cycle:

2,	80,	100,	4,	136,	127,	107,	249,	106,	182,	99,	128,	150,	42,
5,	81,	84,	5,	136,	29,	106,	242,	127,	180,	100,	128,	152,	300,
2,	80,	84,	4,	136,	16,	12,	94,	252,	169,	100,	128,	150,	292,
3,	74,	89,	5,	135,	14,	78,	171,	253,	174,	100,	128,	151,	108,
3,	79,	165,	4,	157,	127,	137,	251,	149,	172,	104,	128,	150,	55,
5,	85,	149,	3,	154,	214,	129,	252,	161,	177,	97,	128,	150,	300,
2,	92,	12,	157,	248,	215,	132,	250,	164,	179,	101,	128,	150,	214,
4,	89,	13,	81,	192,	215,	133,	252,	165,	183,	99,	128,	151,	42,
3,	90,	103,	5,	137,	131,	107,	244,	106,	185,	101,	128,	151,	157,

Application 1

Gait

- Elvina's walking cycle:



Application 1

Implementation

- Standard GA, tournament selection
- Creep mutation
- Averaging crossover

Application 1

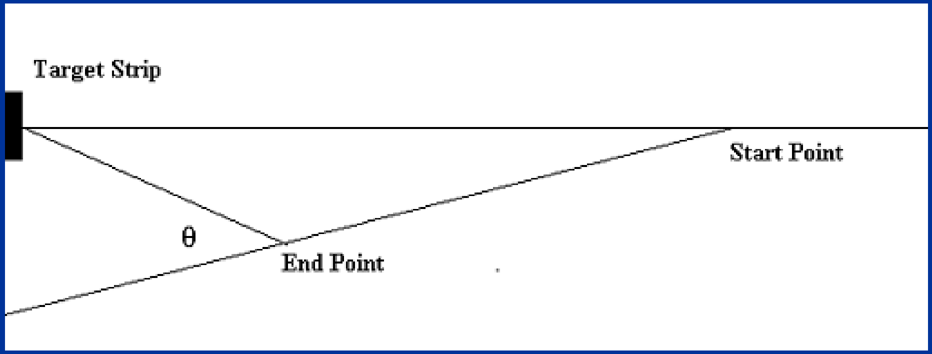
Evolutionary algorithm

- Implementation
  - Population
    - 30 individuals
    - Individuals randomly created with a uniform distribution of genes, over a given, empirical search range
  - Steady-state tournament selection
  - Crossover: $$c_{ki} = p_{ki} + \alpha_{ki}(p_{1i} - p_{2i})$$
  - Mutation: $$c_{ki,mutate} = c_{ki} + \delta_{ki} m_{ki}$$

Application 1

Fitness

- The *camera* is used to determine how *straight* the robot moved during the trial.
  –



- The angular deviation, Θ, is the difference from the desired (straight) path of locomotion and the performed path.

Application 1

Fitness

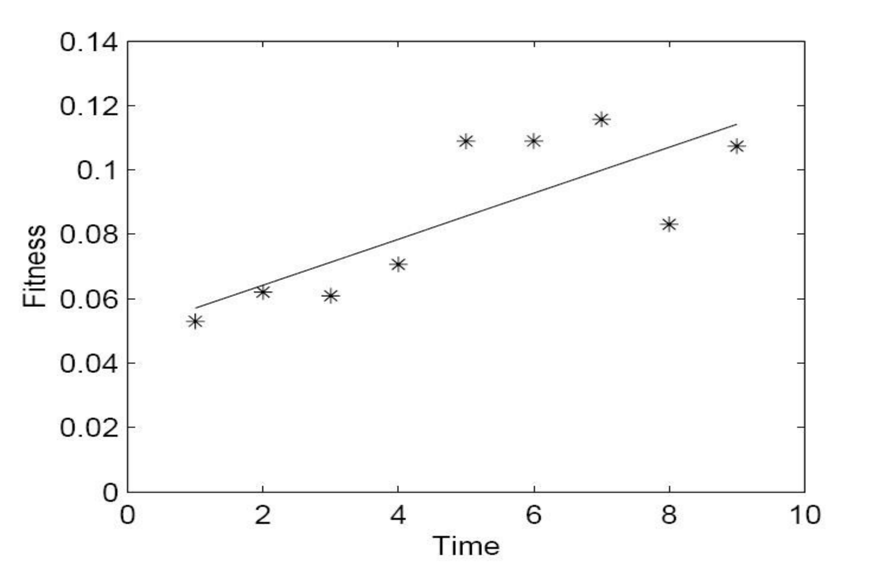
- Fitness is a product of walking velocity and how straight the robot walked:

$$score = v(d_0, d_f, t) \cdot s(\theta, d_f)$$

Application 1

Results

- The best evolved individual fitness: **0.17**
- The best hand-coded gait fitness: **0.11**, i.e. **55%** improvement (mostly due to a straighter path of locomotion)



Conclusions from application 1

- Lesson learned:
  - Evolving efficient gaits with real physical hardware is a challenging task…
    - It is time consuming. Feedback is slow, and the experiment requires manual supervision all the time.
    - It is extremely demanding for the hardware!
    - On-line evolution in hardware constrains the number of generations.

Application 2

Online optimization of gaits in a real, physical robot II

Evolutionary optimization of bipedal gait in a physical robot

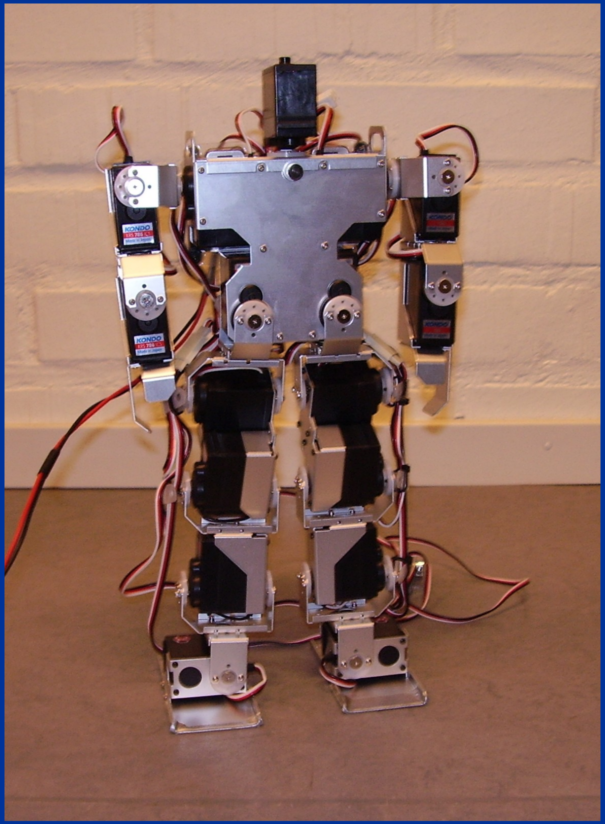
- K. Wolff, D. Sandberg, M. Wahde

Chalmers University of Technology, Göteborg, Sweden

Application 2

EA in a real robot

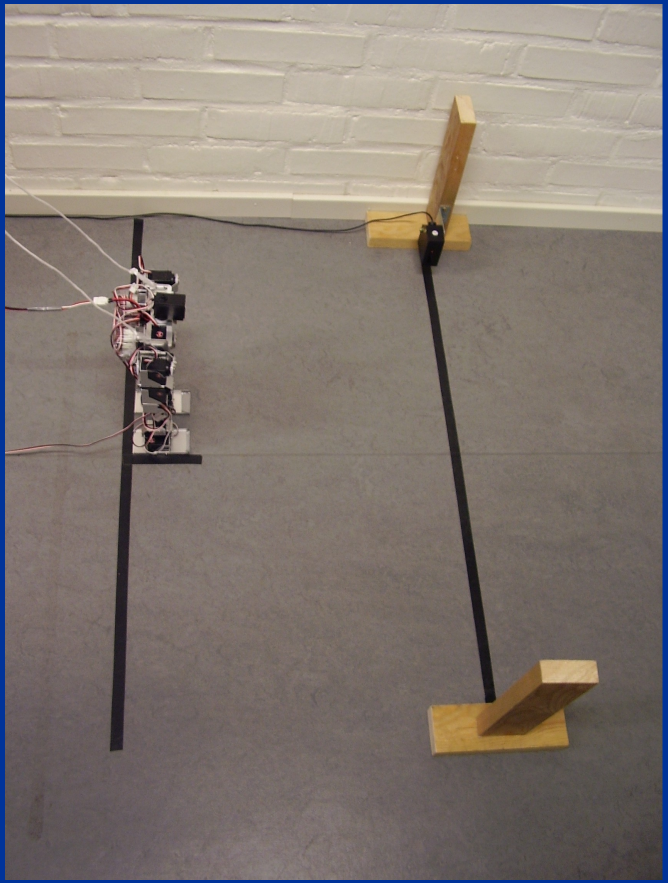
- The Kondo robot
  - 17 DOFs
  - No sensors
  - FAST!



Application 2

Experiment

- Online optimization of hand-coded gait pattern
- Similar to previous experiment, but new states were added.



Application 2

Fitness

$$f = \frac{T_{\mathrm{SG}}}{T},$$

- T_{SG} = time for individual executing the standard gait.

Application 2

Standard gait and best gait

TABLE I

THE STANDARD GAIT, CONSISTING OF SIX STATES THAT ARE EXECUTED CYCLICALLY. THE PARAMETERS SHOWN IN ITALICS ARE KEPT CONSTANT DURING THE EVOLUTIONARY OPTIMIZATION PROCEDURE, I.E. THEY ARE UNAFFECTED BY PARAMETRIC MUTATIONS, SEE ALSO SUBSECT. II-D.1.

State	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9	φ_{10}	φ_{11}	φ_{12}	φ_{13}	φ_{14}	φ_{15}	φ_{16}	φ_{17}	R
1	9	11	90	120	150	169	90	*97*	85	49	127	*80*	*103*	69	114	46	*78*	6
2	9	11	90	120	175	169	90	*97*	85	49	127	*79*	*103*	107	172	23	*78*	3
3	30	11	90	90	175	169	90	*97*	85	49	127	*79*	*102*	116	128	73	*76*	3
4	30	11	90	90	175	169	90	*73*	119	82	124	*104*	*79*	88	111	67	*102*	6
5	9	11	90	60	175	169	90	*73*	72	15	150	*104*	*79*	88	111	64	*102*	3
6	9	11	90	90	150	169	90	*73*	65	66	93	*107*	*75*	102	124	64	*102*	3

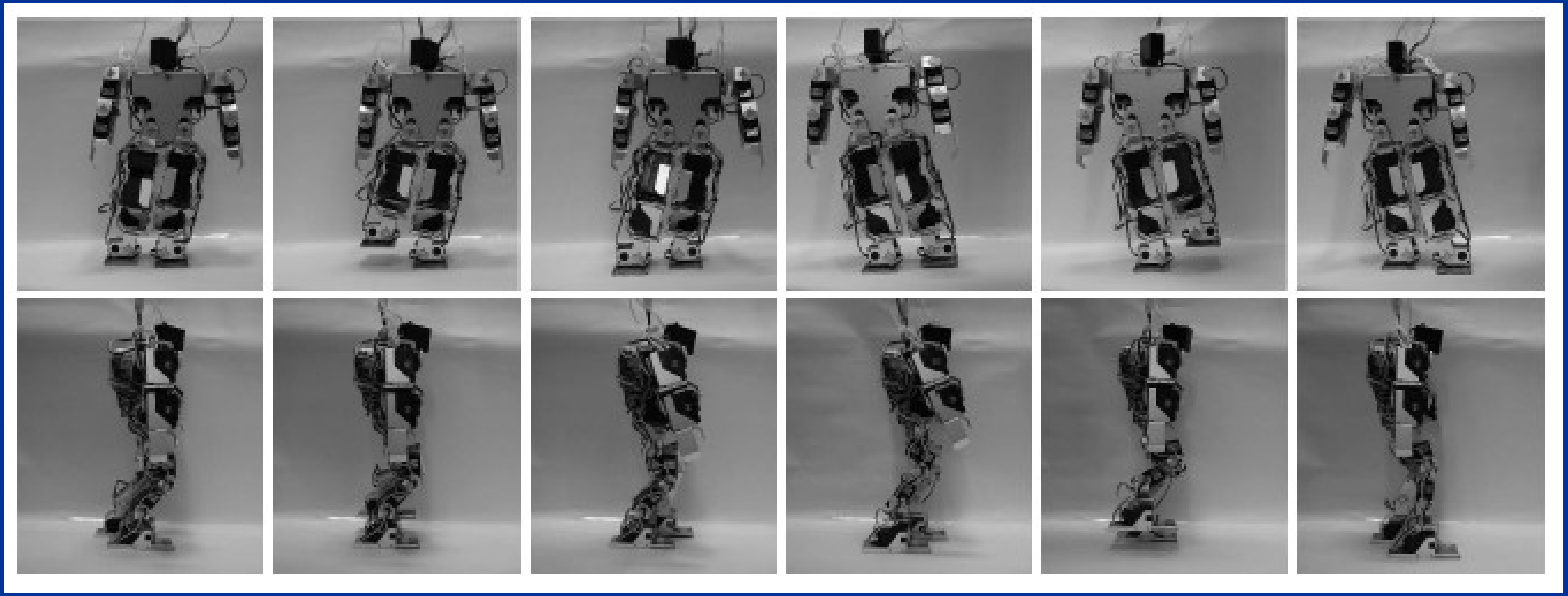
TABLE II

THE BEST GAIT, CONSISTING OF EIGHT STATES. NOTE THAT THE PARAMETERS SHOWN IN ITALICS ARE UNAFFECTED BY PARAMETRIC MUTATIONS.

State	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9	φ_{10}	φ_{11}	φ_{12}	φ_{13}	φ_{14}	φ_{15}	φ_{16}	φ_{17}	R
1	10	11	90	120	150	169	90	*97*	85	49	127	*80*	*103*	69	114	46	*78*	5
2	9	11	90	120	175	169	90	*97*	85	49	127	*79*	*103*	113	172	23	*78*	0
3	31	11	90	90	175	169	90	*97*	82	49	127	*79*	*102*	121	128	73	*76*	0
4	30	11	90	89	175	169	90	*85*	99	65	125	*91*	*90*	104	122	71	*89*	3
5	30	11	90	90	175	169	90	*79*	108	73	124	*97*	*84*	96	119	68	*95*	4
6	30	11	90	90	175	169	90	*73*	117	82	124	*104*	*79*	88	116	67	*102*	3
7	9	12	90	60	178	172	87	*73*	72	15	154	*104*	*79*	88	111	64	*102*	3
8	9	11	95	90	147	174	90	*73*	65	71	94	*107*	*75*	102	124	64	*102*	0

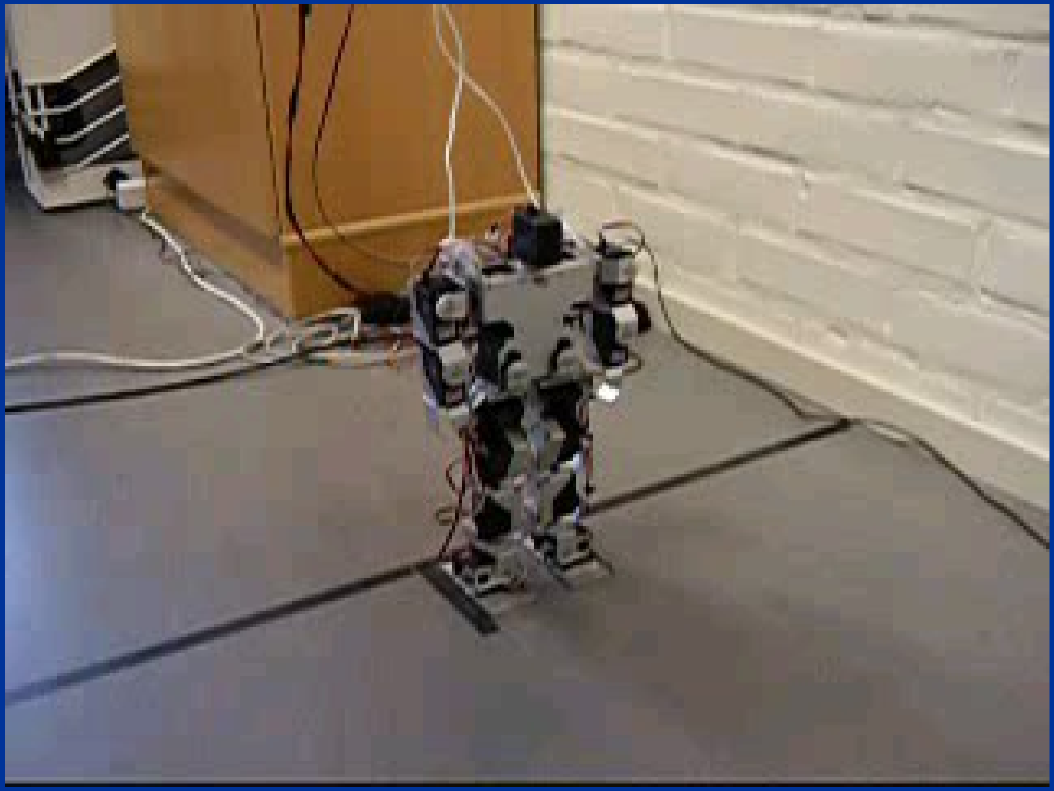
Application 2

Gait



Application 2

Best evolved gait



Conclusions from applications 1 and 2

- Application 1:
  - A more stable gait was obtained.
- Application 2:
  - The walking speed increased by 65%.
  - Structural modifications of the gait program.
- Possible to obtain significant improvements of bipedal gaits with an EA in a real physical bipedal robot.
- Typical experiment duration: 24 man-hours (Application 2, 900 evaluated individuals).

Application 3

Structural evolution of central pattern generators for bipedal walking in 3D simulation

- K. Wolff, J. Pettersson, A. Heralic, M. Wahde.

Adaptive Systems Research Group,
Chalmers University of Technology,
Göteborg, Sweden

Application 3

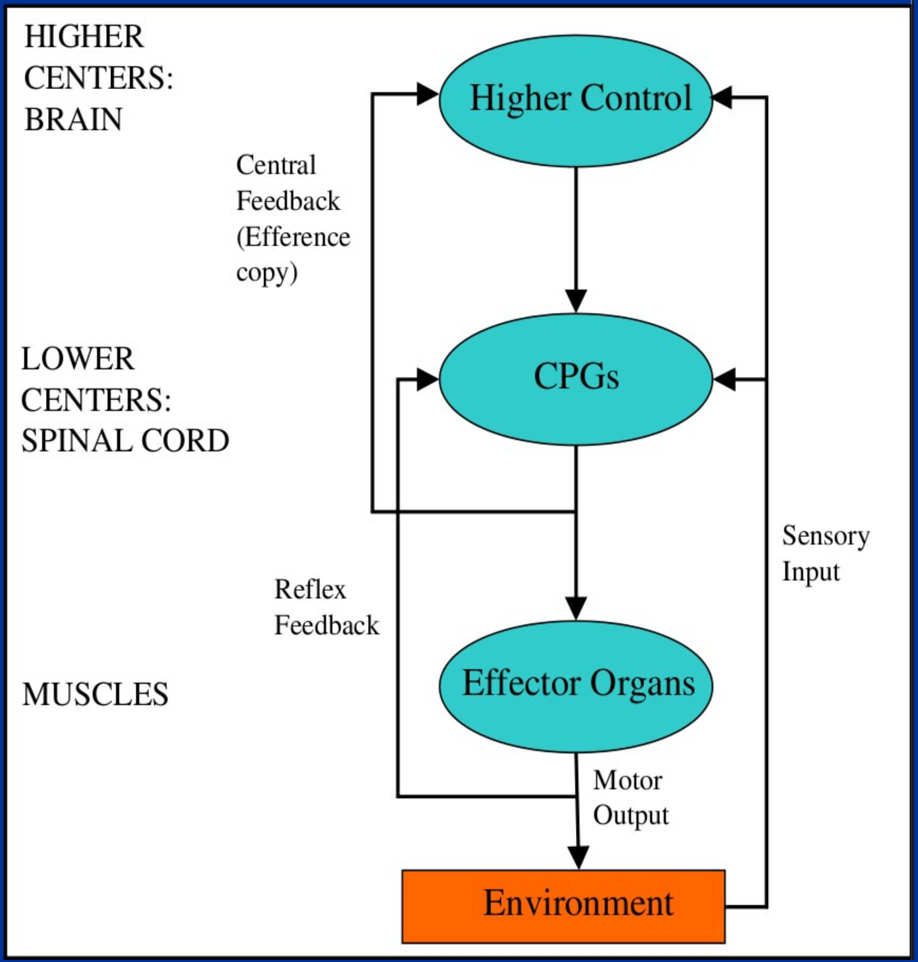
Objective

- Bipedal gait synthesis for a simulated robot by structural evolution of CPG networks.
  - That is, both CPG network parameters and feedback network interconnection paths are determined using an evolutionary algorithm (EA).

Application 3

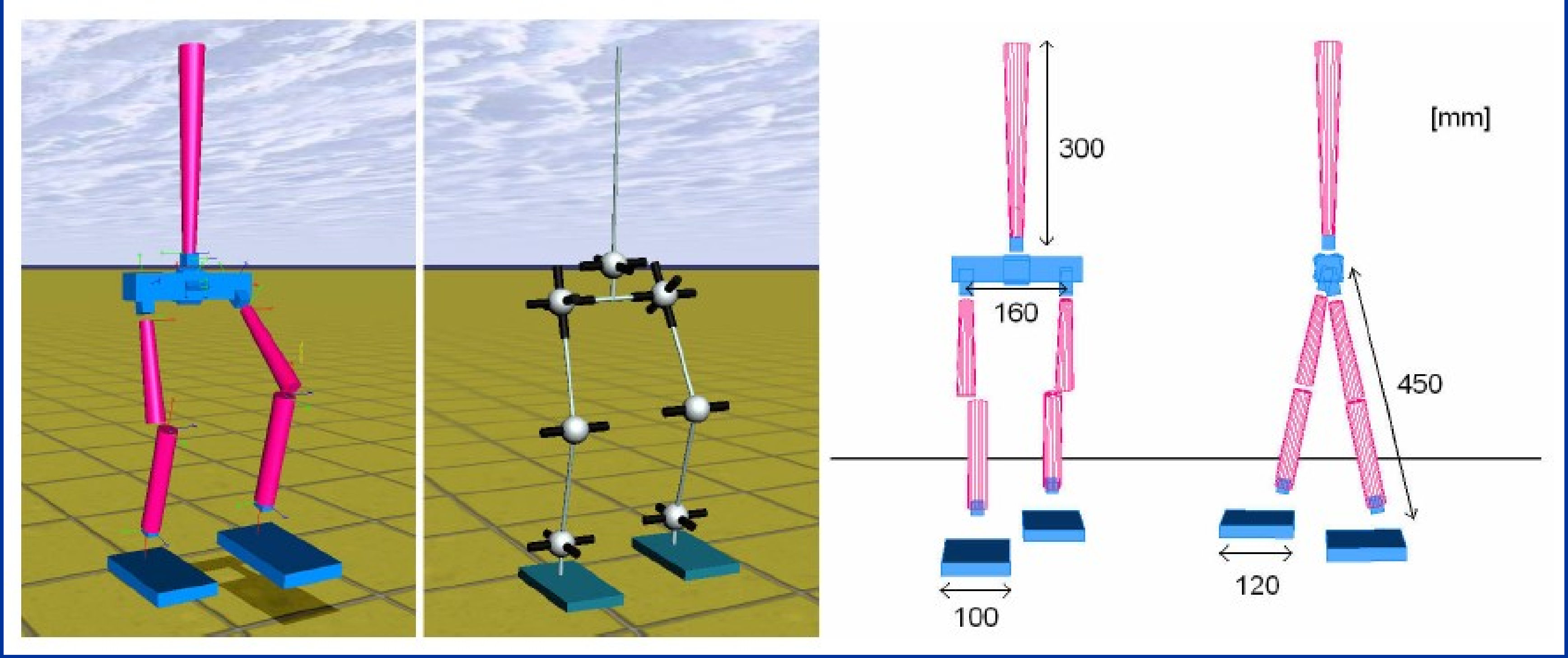
Motor Systems Hierarchy

- Key elements:
  - Central pattern generator (CPG)
  - Higher motor centers
  - Feedback circuits
- Hierarchical organization:
  - Allows for the lower levels to control reflexes
  - Higher levels give commands without having to specify the details



Application 3

The robot



Application 3

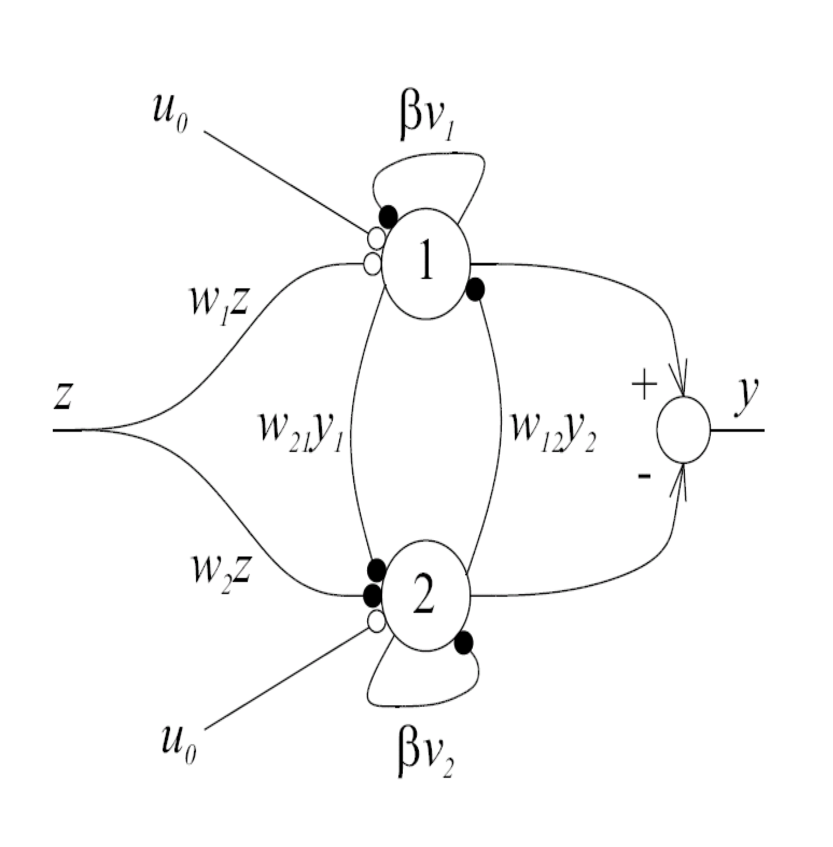
Central Pattern Generators

- CPGs are neural circuits capable of producing *oscillatory output* given *tonic (non-oscillating) input*
- CPGs have been extensively studied in animals:
  - simple animals; lamprey, salamander
  - complex animals; cats
- Observations support the notion of CPGs in humans:
  - treadmill training of patients with spinal cord lesion

Application 3

The Matsuoka oscillator

$$
\begin{aligned}
\tau_u \dot{u}_i &= -u_i - \beta v_i + \sum_{j=1}^{n} w_{ij} y_j + u_0, \\
\tau_v \dot{v}_i &= -v_i + y_i, \\
y_i &= \max(0, u_i),
\end{aligned}
$$



u_i = inner state

v_i = degree of self inhibition

τ_u and τ_v time constants

u_0 = bias (tonic input)

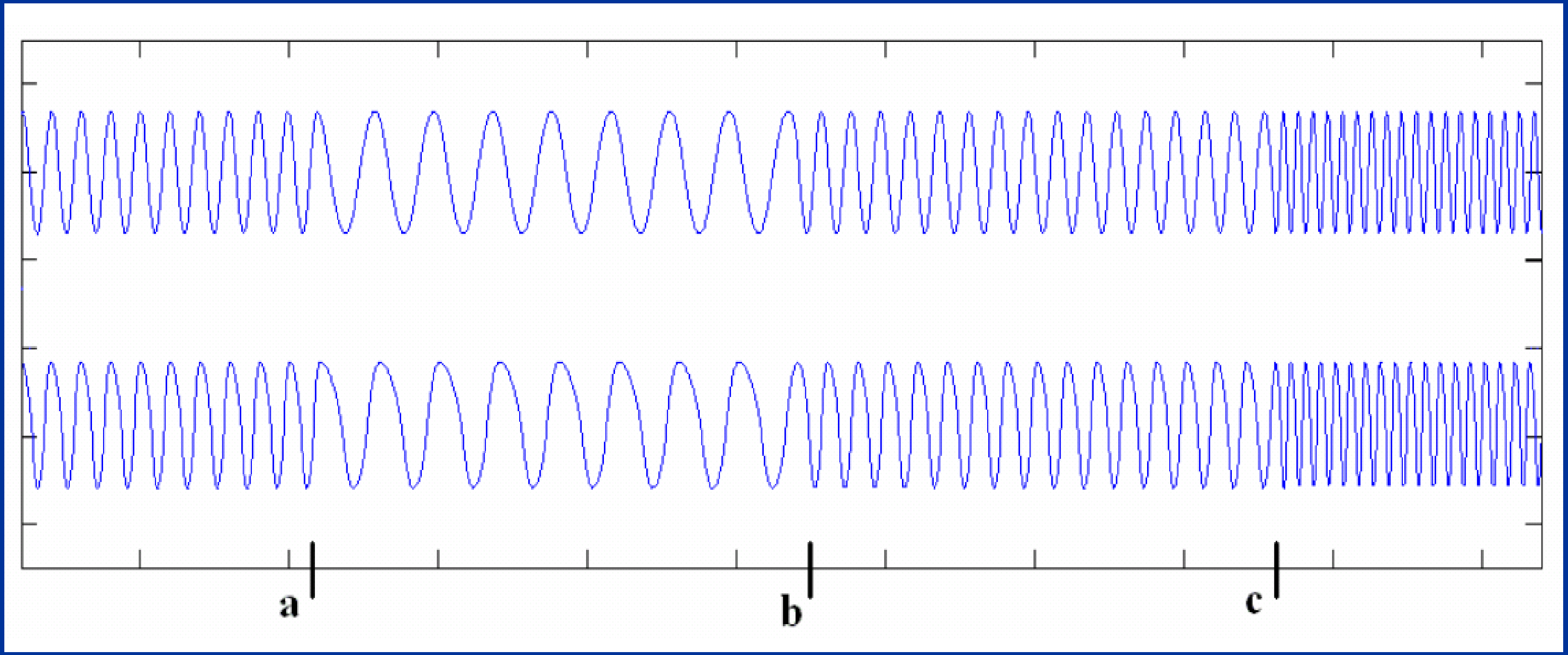
w_{ij} = connection weights

y_i = output

Application 3

The Matsuoka oscillator

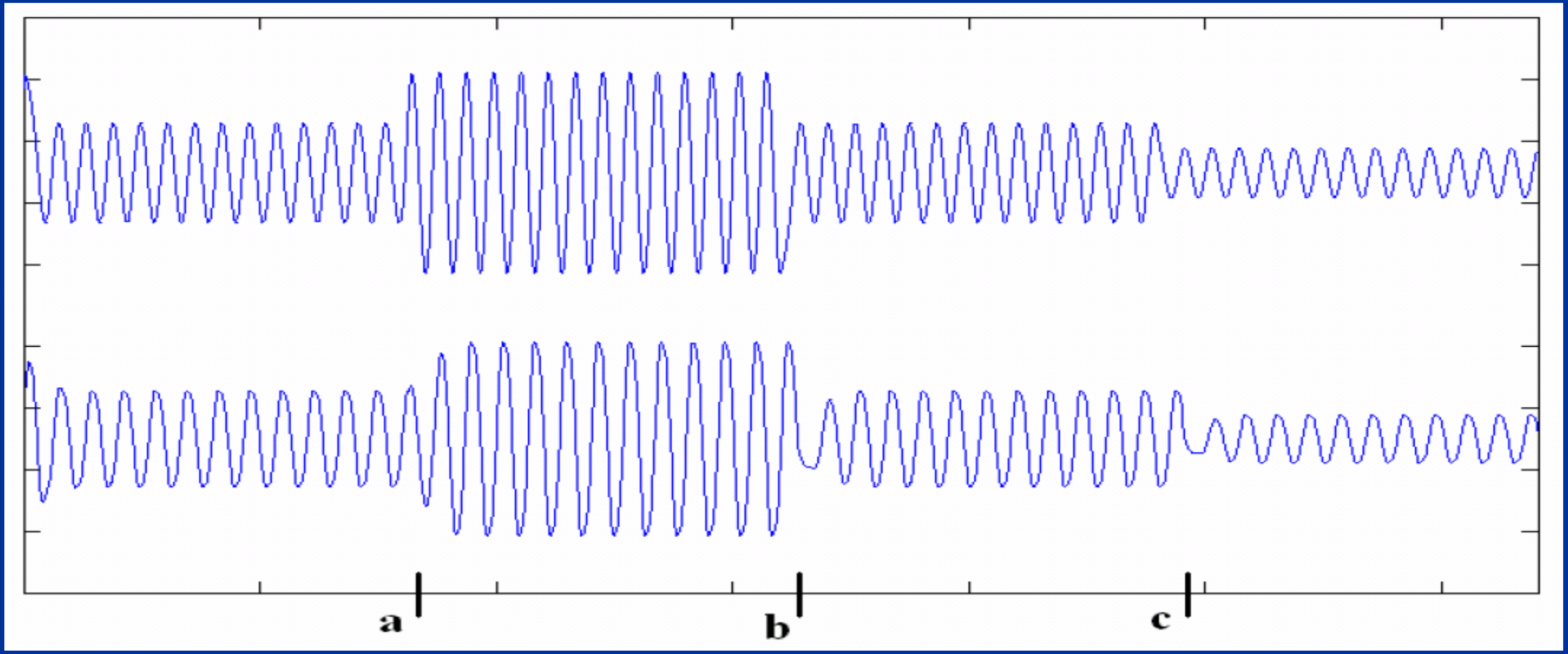
- Frequency variation occurs if the time constants τ_u and τ_v are varied.



Application 3

The Matsuoka oscillator

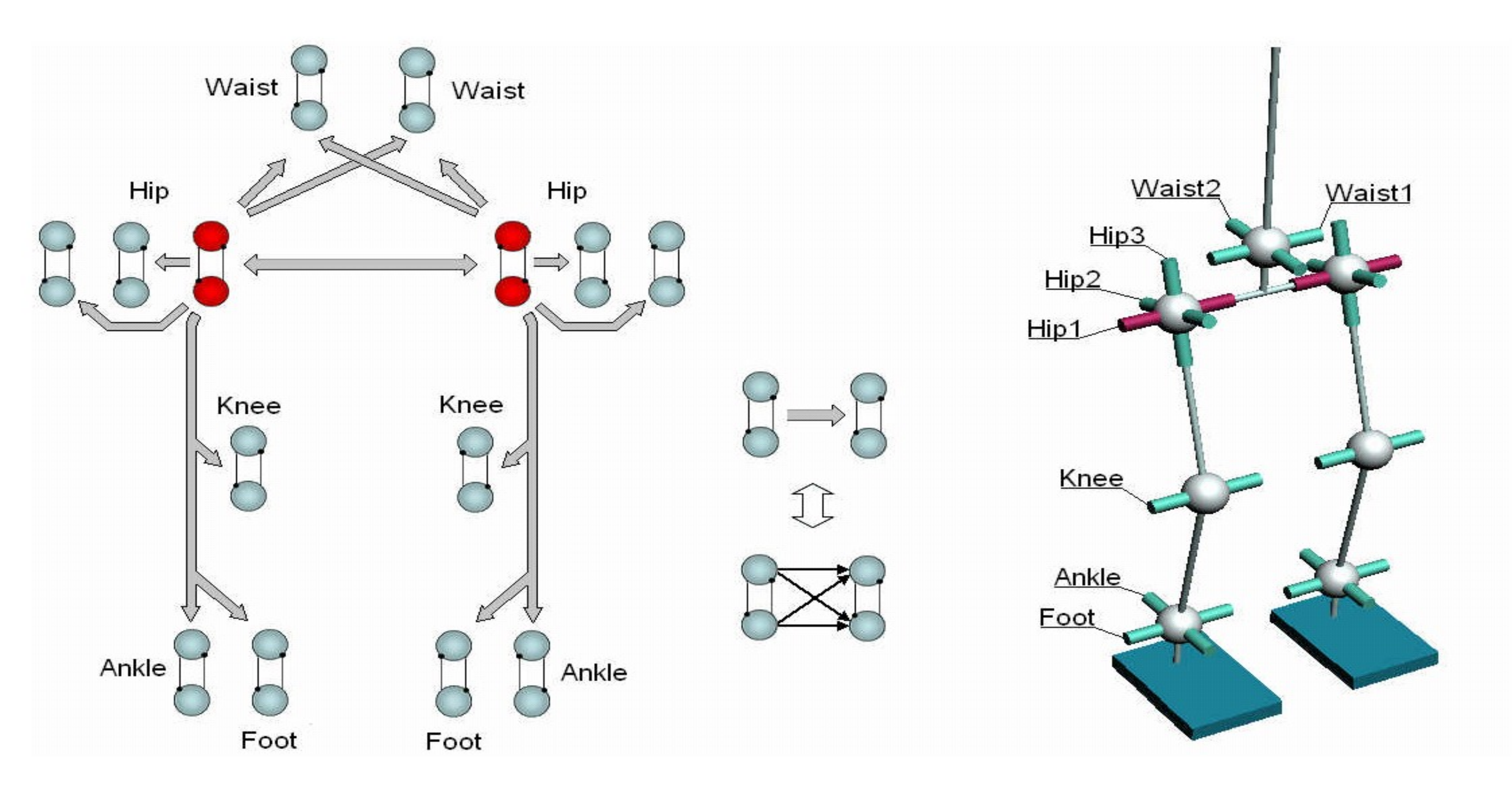
- Amplitude variation occurs if the bias u_0 is varied



Application 3

CPG network

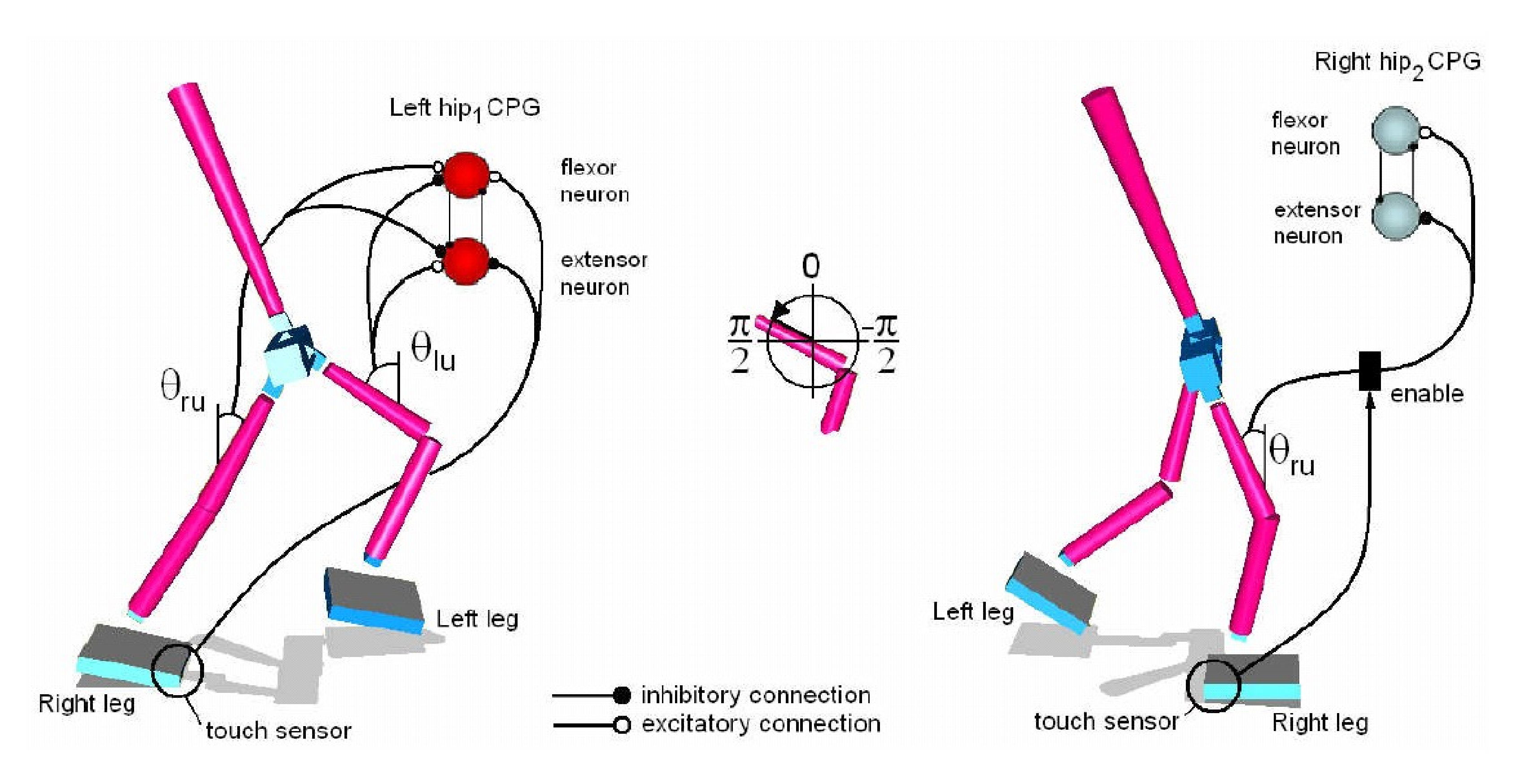
- An arrow indicates the *possibility* of connections



Application 3

Feedback network

- Waist, thigh, and leg angles, and foot contact



Application 3

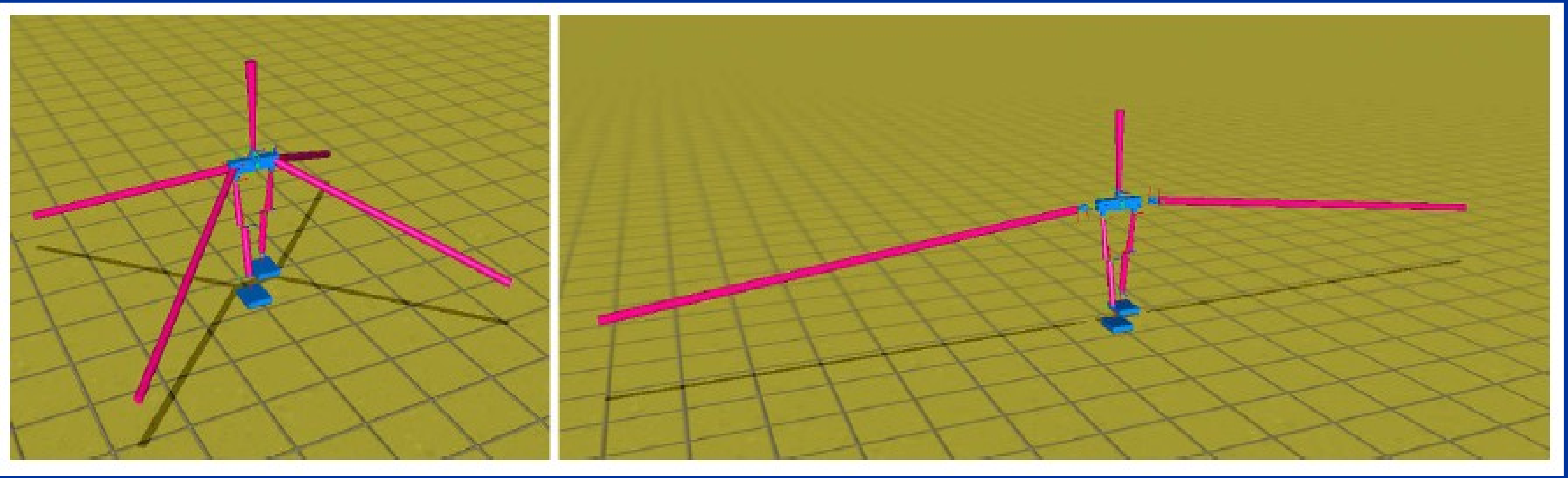
GA optimization

- Intuitive design and manual tuning of parameters of CPG networks is difficult and time-consuming => *optimal performance cannot be guaranteed!*
- Evolutionary algorithms are very suitable for this kind of ”open-ended” optimization and design.

Application 3

Support structure

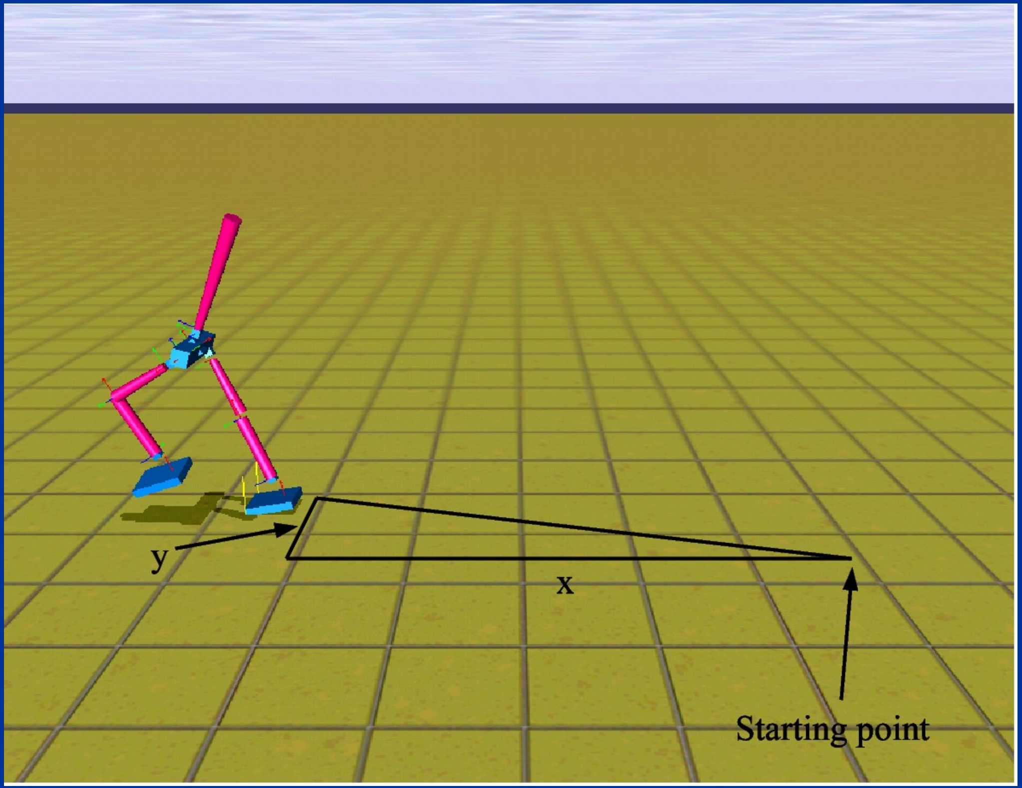
- A massless support structure was used in the early stages of the EA runs, in order to generate natural, upright gaits.
- Helps the robot to balance.



Application 3

Evolutionary algorithm

- Objective function: *f (i)* = |***x*** - ***y***|
  - *[Distance walked forward] – [sideways deviation]*



Application 3

Evolutionary algorithm

- A ”standard” GA
  - Population of 180 individuals
  - Mutation, no crossover
  - Tournament selection, size: 8, $p_{sel} = 0.75$
  - Fitness function: $f = |\boldsymbol{x} - \boldsymbol{y}|$
    - *[Distance walked forward] – [sideways deviation]*

Application 3

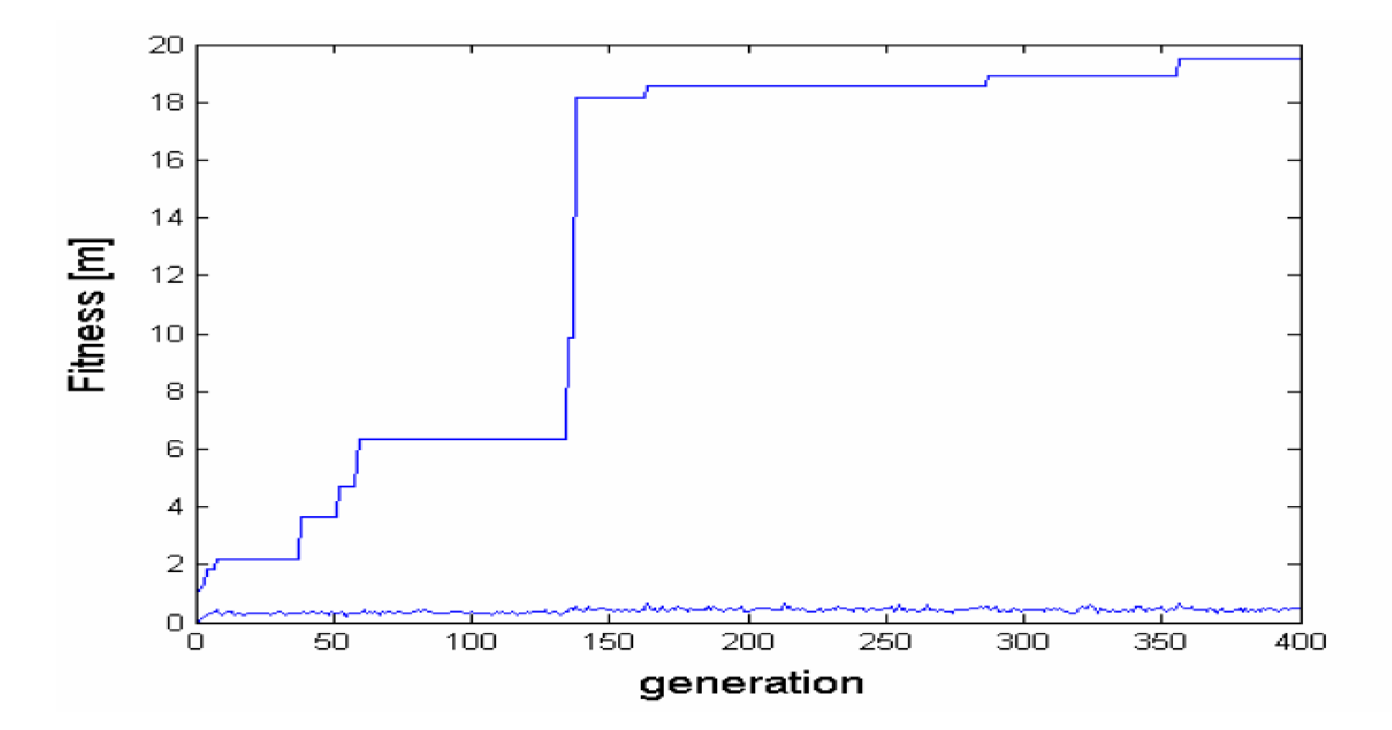
Evolutionary algorithm

- Genome, fixed length
  - CPG network chromosome:
    - Binary, len = 32, connection[i] = 0,1
    - Real, len = 32, weights (sign and strength)
  - Feedback network:
    - Real, len = 20, weights (sign and strength)
  - Three chromosomes with 84 genes

Application 3

Results

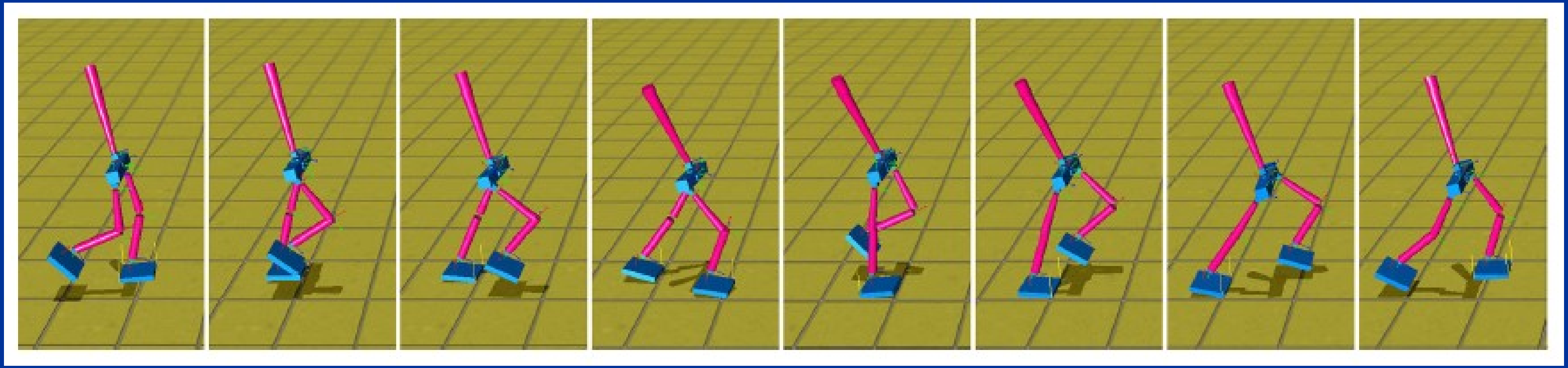
- Fitness progress:
  - Fitness landscape with sparse, narrow peaks (low average fitness after many generations).



Application 3

Results

- Best individual



Application 3

Conclusions from application 3

- Stable bipedal gait was generated.
- Support structure:
  - Four point did not help much (=> cheating)
  - Two point support was useful
  - Without support, often stuck in local optima
- More feedback could lead to improved control and robustness
- Only *straight line* locomotion has been investigated in this study!
- Transfer the results to a real robot in the future.

Simulation vs. Real robots

- **Simulation:**
  - Fast, (relatively) easy to parallelize.
  - EAs usually require a large number of evaluations => Simulations are dominating in the ER field.
  - Hard to **model** [environment + robot] accurately ***=> Reality gap!***
- **Real, physical robots:**
  - Much more challenging!
  - Evaluation of individuals slow (real time).
  - But in ER no model is required in order to control the robot = > reality gap is NOT an issue! **=> Motivates the use of ER (in real robots).**

Application 4

A general-purpose transportation robot

- a summary of work in progress

Application 4

Project partners

- Sweden:
  - Chalmers University of Technology, Göteborg
- Japan:
  - Waseda University, Tokyo
  - University of Tsukuba
  - Future University, Hakodate

Application 4

Research objectives

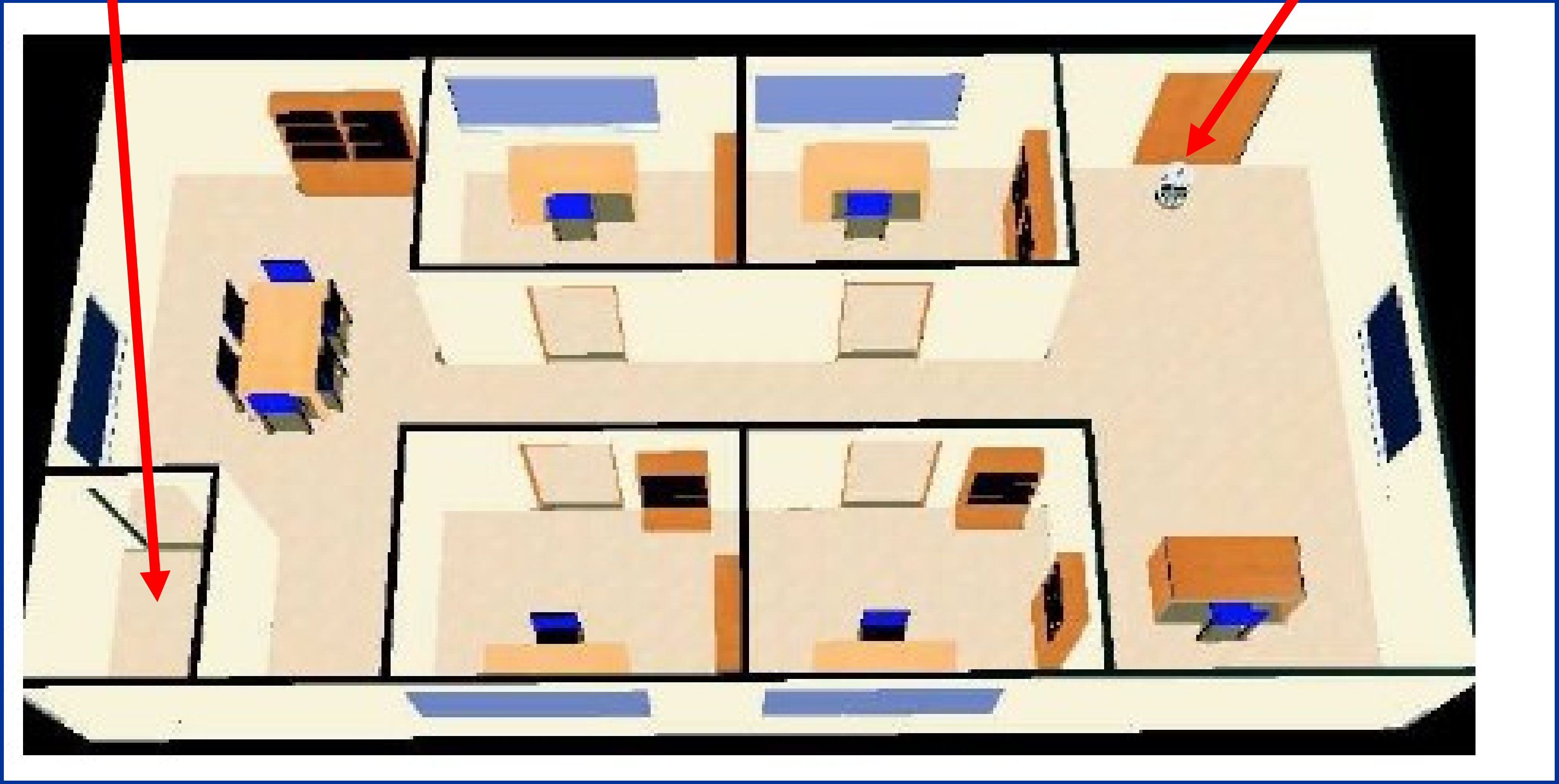
- Develop autonomous robots for guarding tasks, or internal delivery of goods.
  - Intended for use, without human supervision, in *unstructured environments* : Hospitals, offices, industries, etc:
- Evaluate and further develop the *Utility function* (UF) method for behavior selection in autonomous robots.

Application 4

Example: A delivery task

Starting position

Target position



Application 4

Behavioral selection using the utility function method: A case study involving a simple guard robot

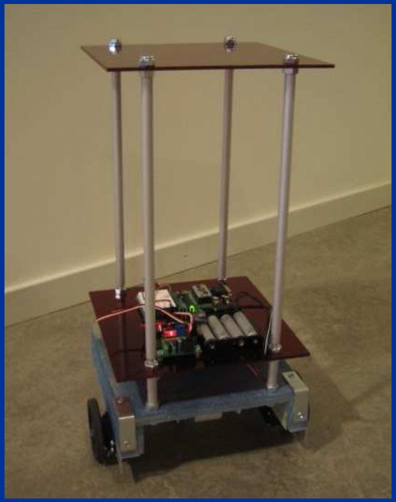
- M. Wahde, J. Pettersson, H. Sandholt, K. Wolff

Adaptive Systems Research Group,
Chalmers University of Technology,
Göteborg, Sweden

Application 4

The robot

- Conceptual model, 1st, and 2nd prototype



Application 4

Main challenge

- Behavior selection system
  - Behavioral organization system must be sufficiently general.
  - Trade-off between efficiency, and safety and self-preservation
- UF method foundation:
  - Robotic brains (control systems) are built in a bottom-up fashion, from simple behaviors.
  - Behavior-based robotics (BBR) approach.

Application 4

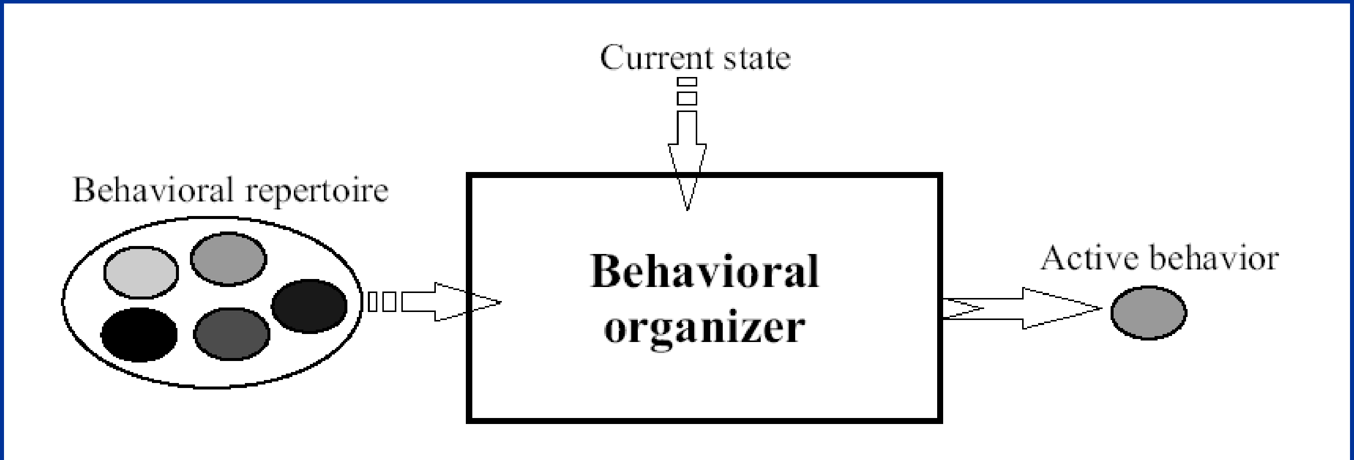
Background

- Derived from economic theory and game theory [1].
- Utility: common currency for comparing the outcome of possible situations.
- Arbitration method.

Application 4

Utility function (UF) method [3]

- Each behavior is associated with a *utility function* that depends on the state variables.
- Behavior selection: the behavior with highest utility is active.



Utility function (UF) method [3]

- Utility functions (normally polynomials in the state variables) are optimized in simulations, using an EA.

$$
\begin{aligned}
U_1 &= U_1(s_1, s_2, s_3, s_4, s_5, p_1) \\
U_2 &= U_2(s_1, s_2, s_3, s_4, s_5, p_1, x_1) \\
U_3 &= U_3(p_1, x_2)
\end{aligned}
$$

Application 4

Utility function (UF) method

- Polynomial ansatz:

$$U(s_1, x_1) = a_{00} + a_{10}s_1 + a_{01}x_1 + a_{20}s_1^2 + a_{11}s_1x_1 + a_{02}x_1^2.$$

- external var:s (sensor readings): s_i
- internal physical var:s (battery level) p_i
- internal abstract var:s (hormones) x_i
- coefficients: a_{ij}

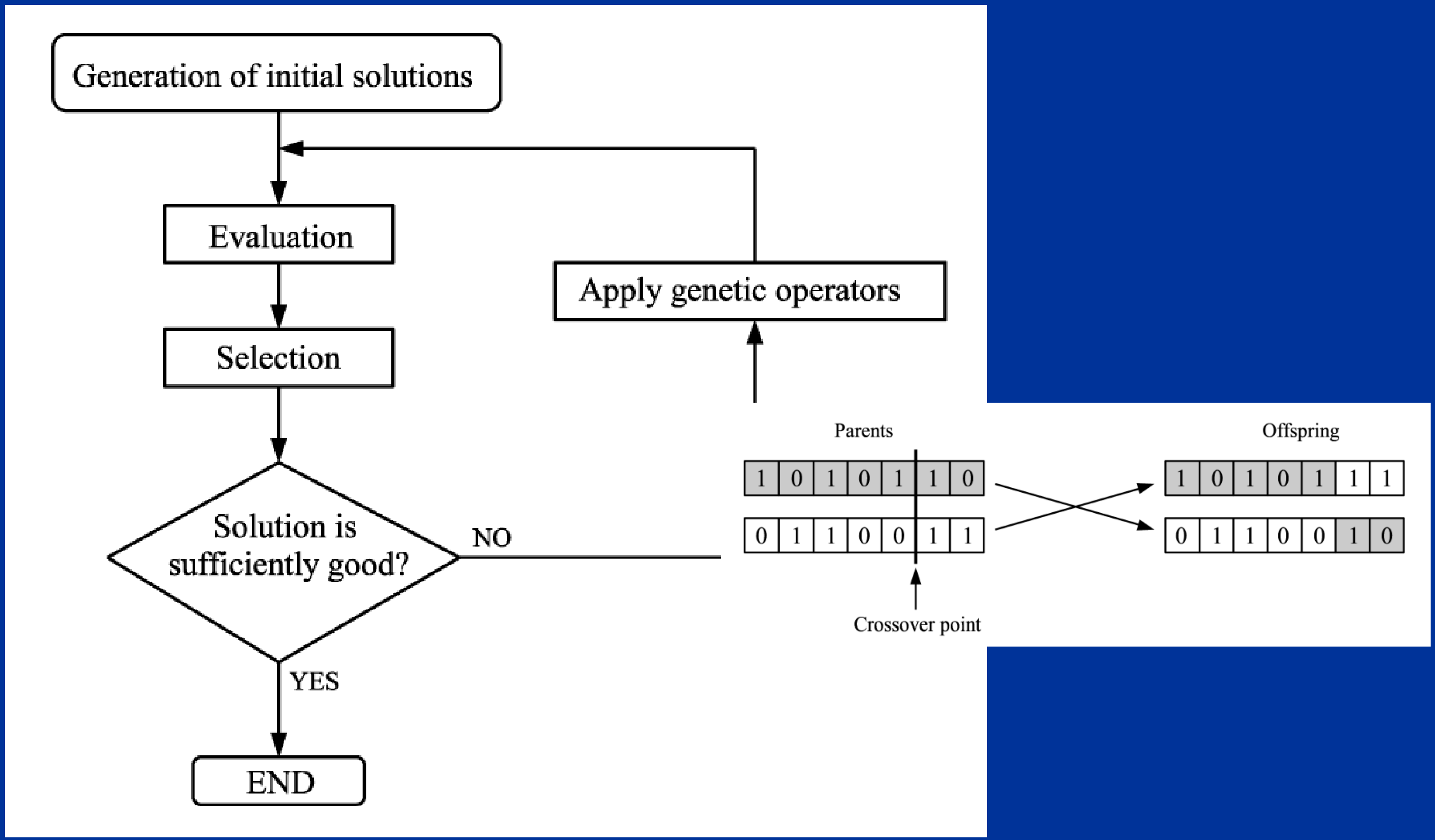
Application 4

The problem

- How to set the coefficients a_{ij}?
- In the UF method, this is done using an evolutionary algorithm (EA).
- Fitness is often associated with a given *task behavior*.

Application 4

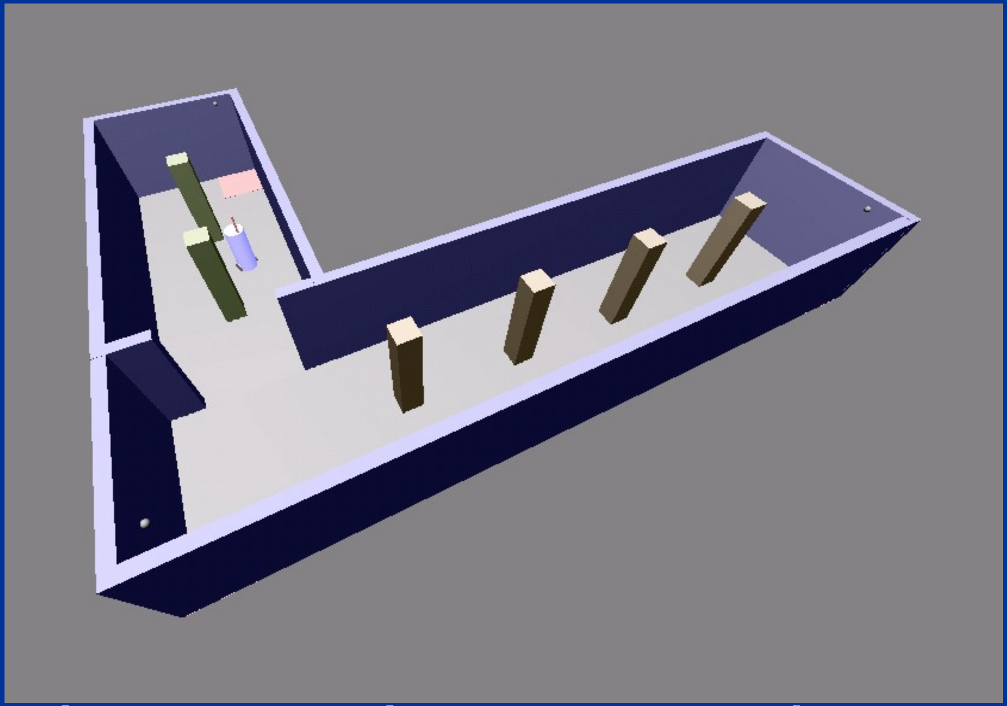
A standard EA

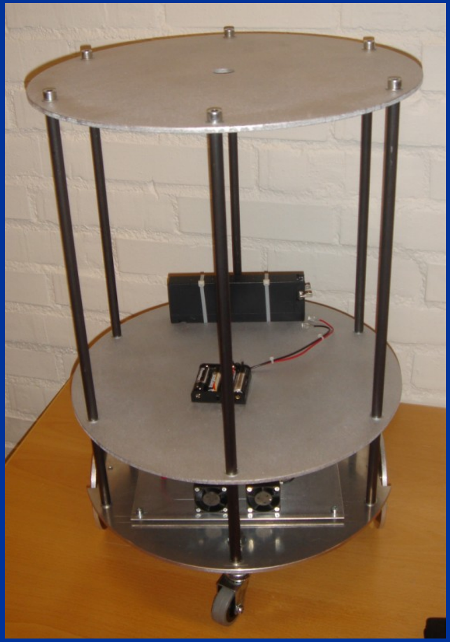


Application 4

Simple guard robot (example)

- Arena patrolled by robot:



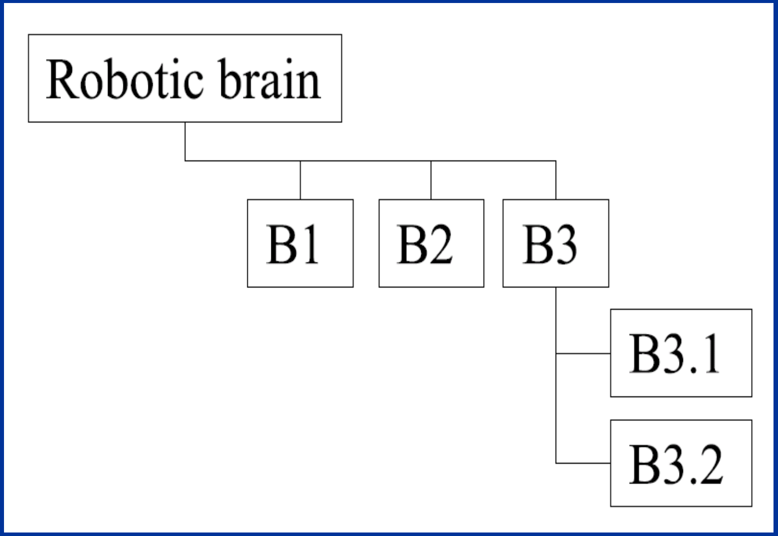


- Fitness: Time spent in *navigation* behavior.

Application 4

Behaviors

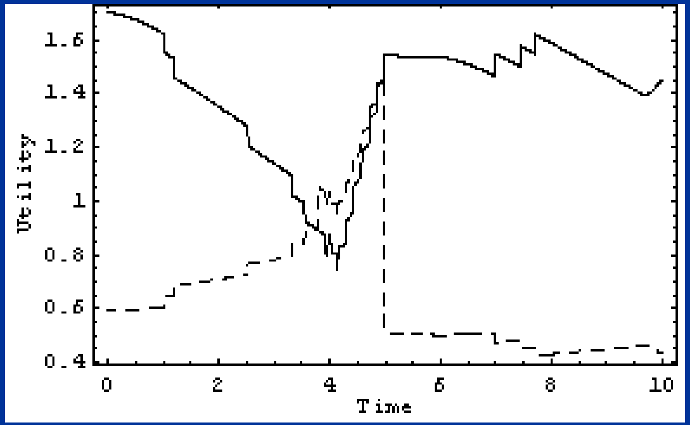
- Straight-line navigation (B1) (*task behavior*)
- Obstacle avoidance (B2)
- Energy maintenance (B3)
  - Corner seeking (B3.1)
  - Battery charging (B3.2)
- B2 and B3 are *auxiliary behaviors*

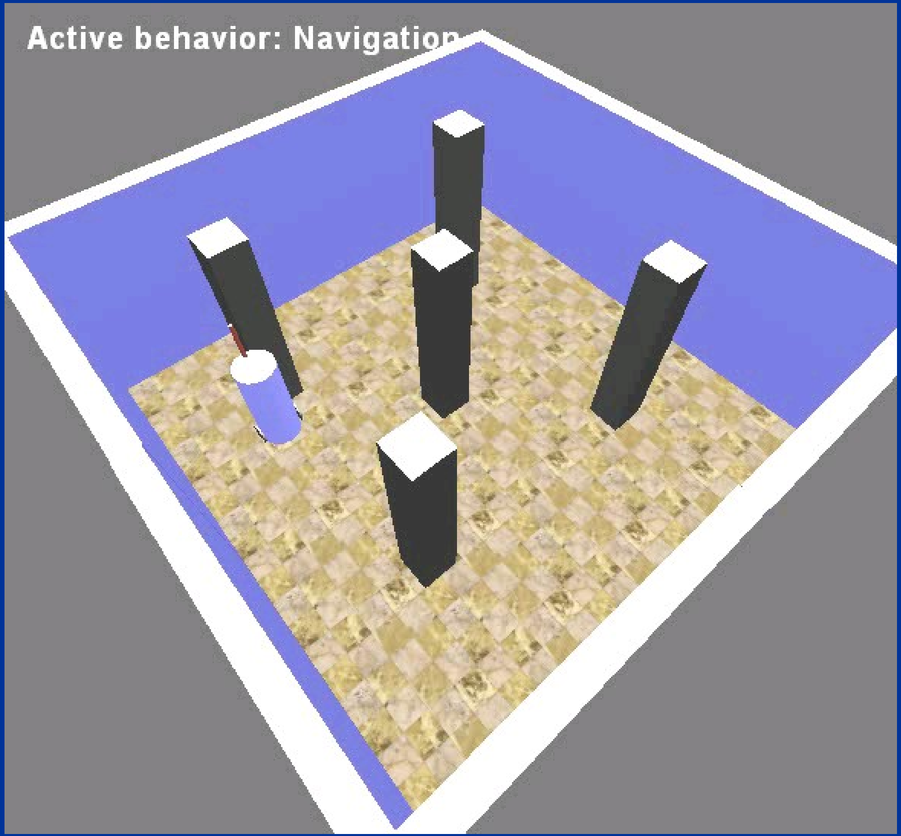


Application 4

Utility function (UF) method

- Simple, illustrative example (*movie*):





Application 4

Conclusions from simulation example

- Systems for appropriate behavior selection were found, typically after the evaluation of a few thousand individuals.
- The best solutions could be found using a higher (3 or 4) polynomial degree.
- For lower polynomial degrees (1-2), solutions can be found more rapidly.

Summary and conclusions

- The UF method have been rigorously tested in several simulation studys.
- UFLib software package has been developed.
  - Availible for free, for academic use.
- Several transportation robot prototypes have been developed.

Application 4

References

- [1] Morgenstern, O. and von Neumann, J. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

- [2] Pettersson, J., Sandberg, D., Wolff, K., and Wahde, M. *Behavioral selection in domestic assistance robots: A comparison of different methods for optimization of utility functions*. Proceedings of SMC 2006.

- [3] Wahde, M. *A Method for Behavioral Organization for Autonomous Robots Based on Evolutionary Optimization of Utility Functions*. Journal of Systems and Control Engineering (IMechI), 217, 249-258, 2003.

- [4] Wahde, M., Pettersson, J., Sandholt, H. and Wolff, K. *Behavioral Selection Using the Utility Function Method: A Case Study Involving a Simple Guard Robot*. Proceedings of AMiRE 2005.

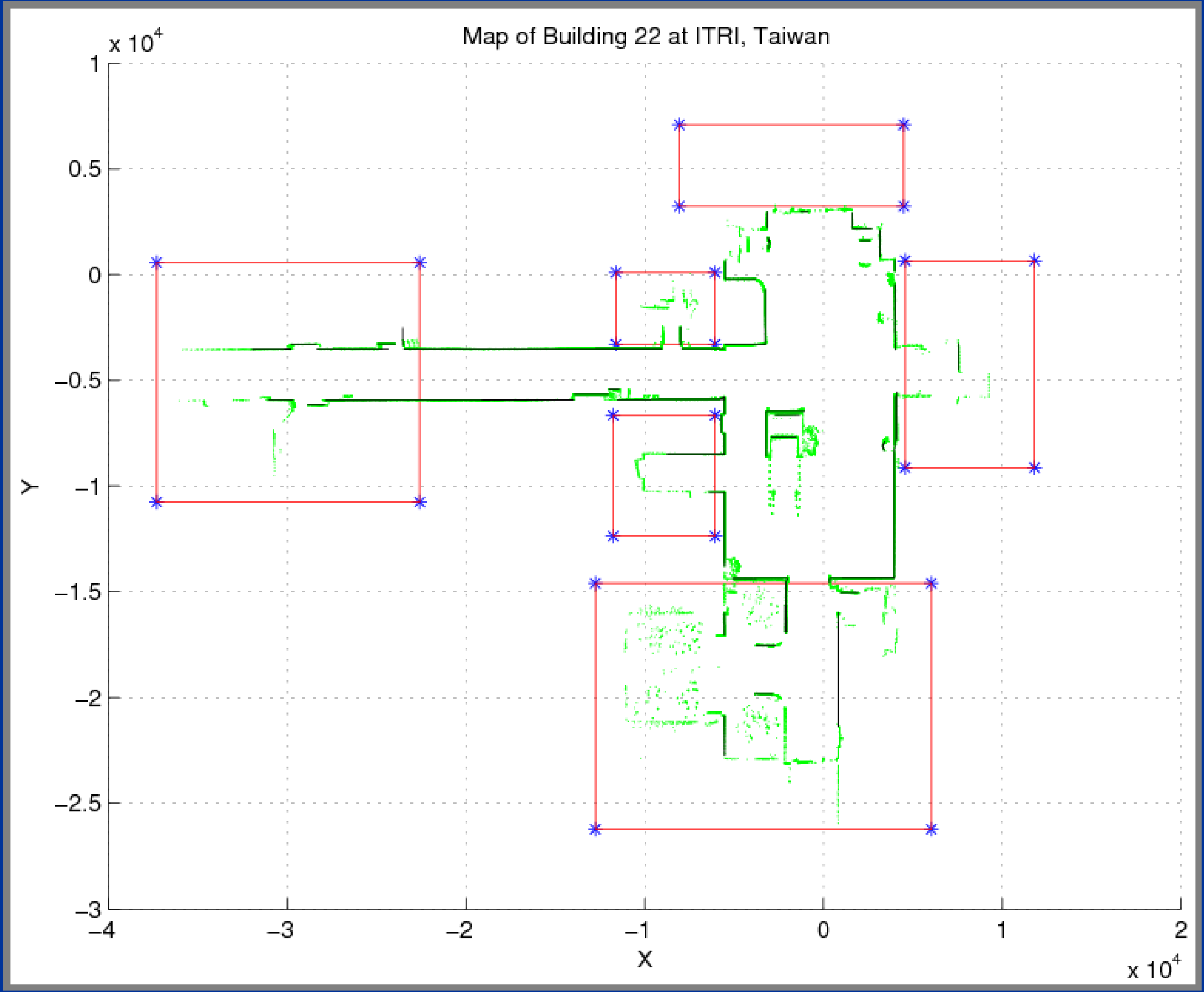
Application 4

Current work: Extended UF

- Application at ITRI, Taiwan: EUF method

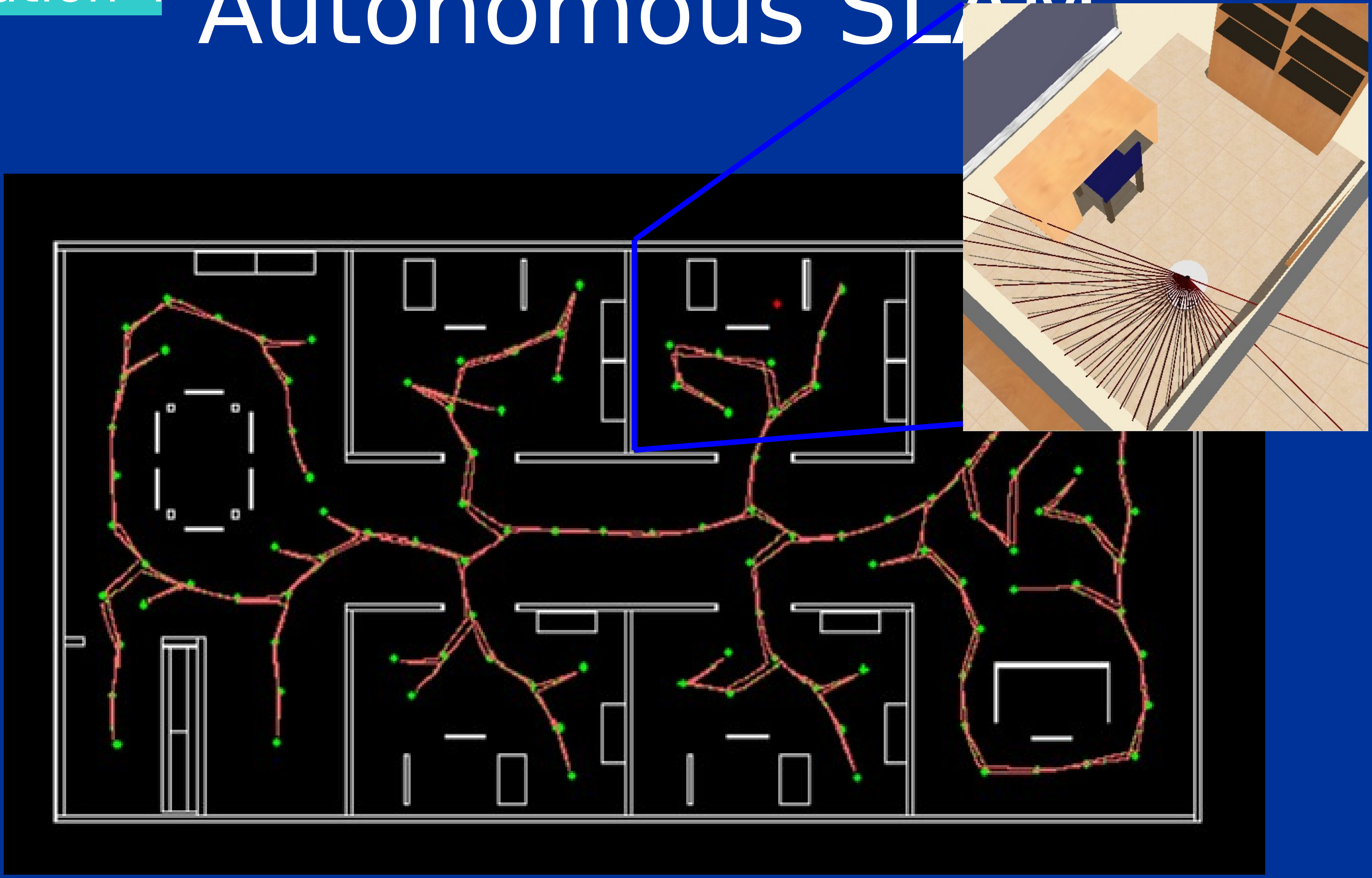
Application 4

Current status: Extended UF



Application 4

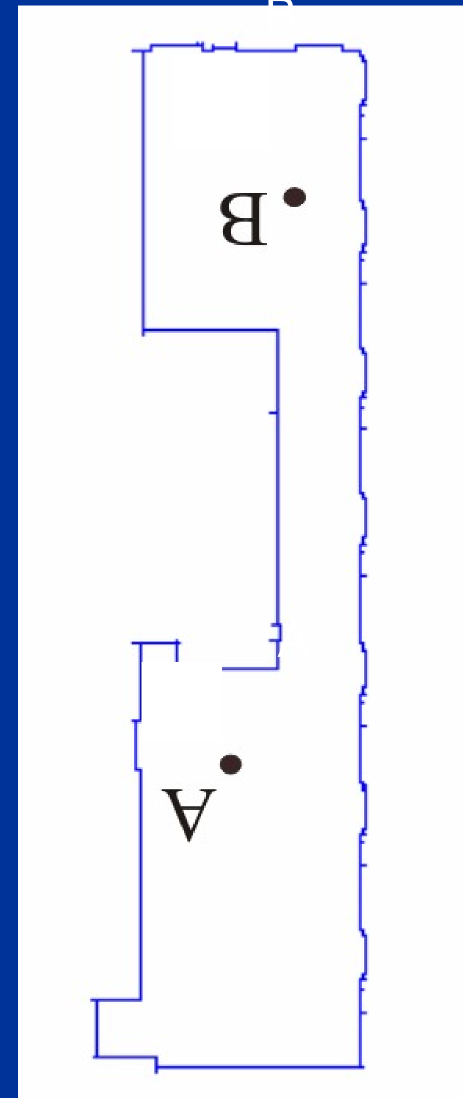
Autonomous SLAM



Distance: 159.2 m, time: 939.1 s.

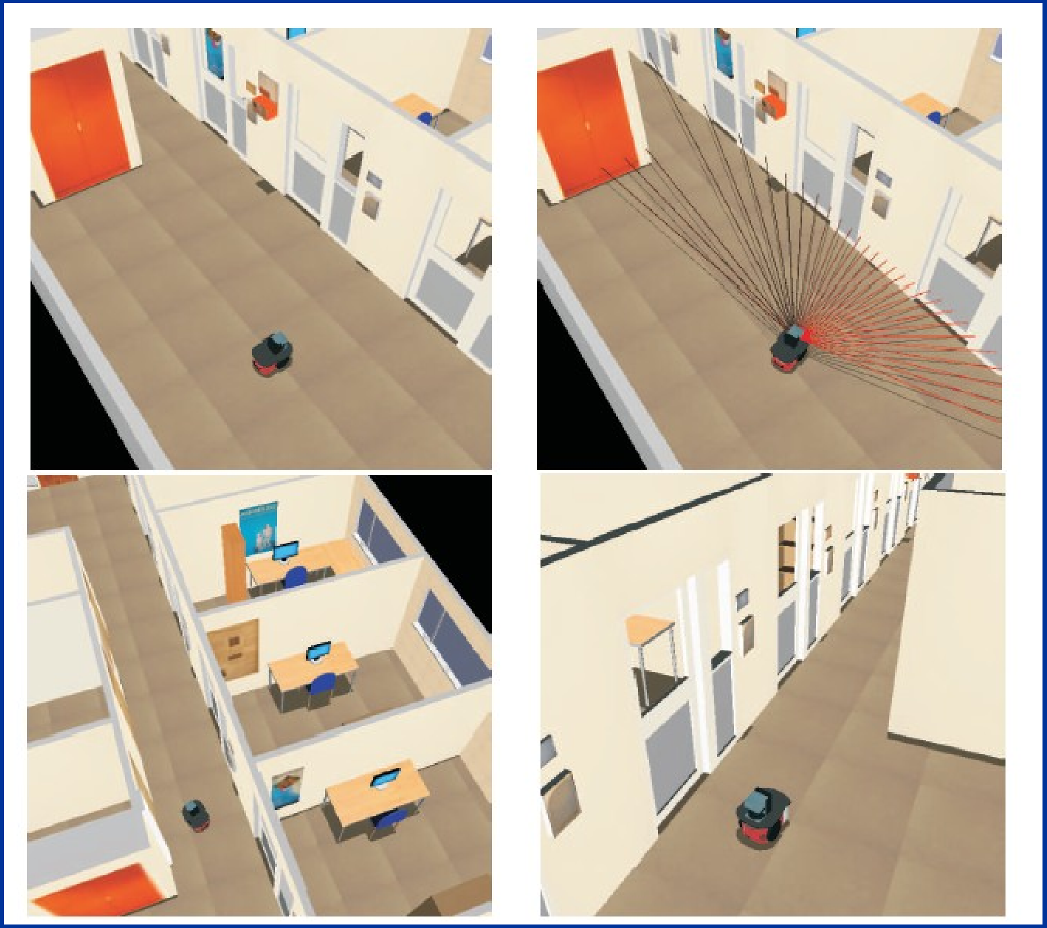
Application 4

Autonomous SLAM



Application 4

Autonomous SLAM



Application 4

Autonomous SLAM

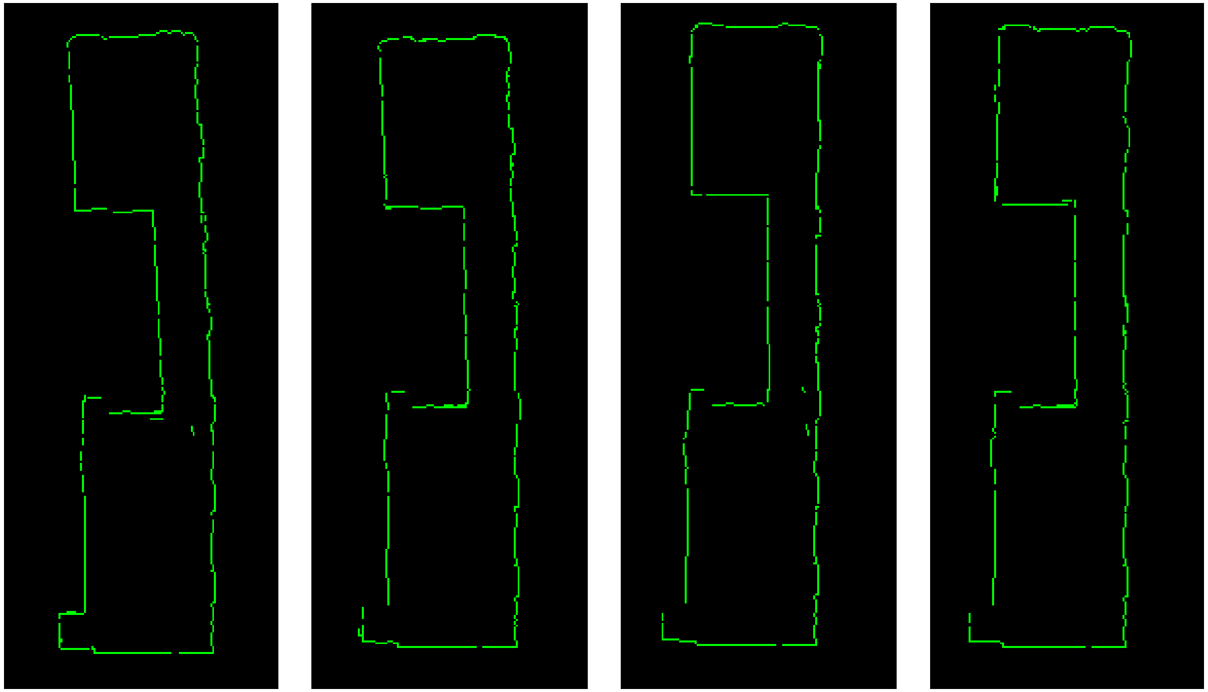


Figure 6.18: *From left to right, the panels show individuals with fitness values 4.08, 6.17, 7.10, and 9.57, respectively. See the main text for further details regarding the individuals that generated the maps.*

Application 4

Autonomous SLAM

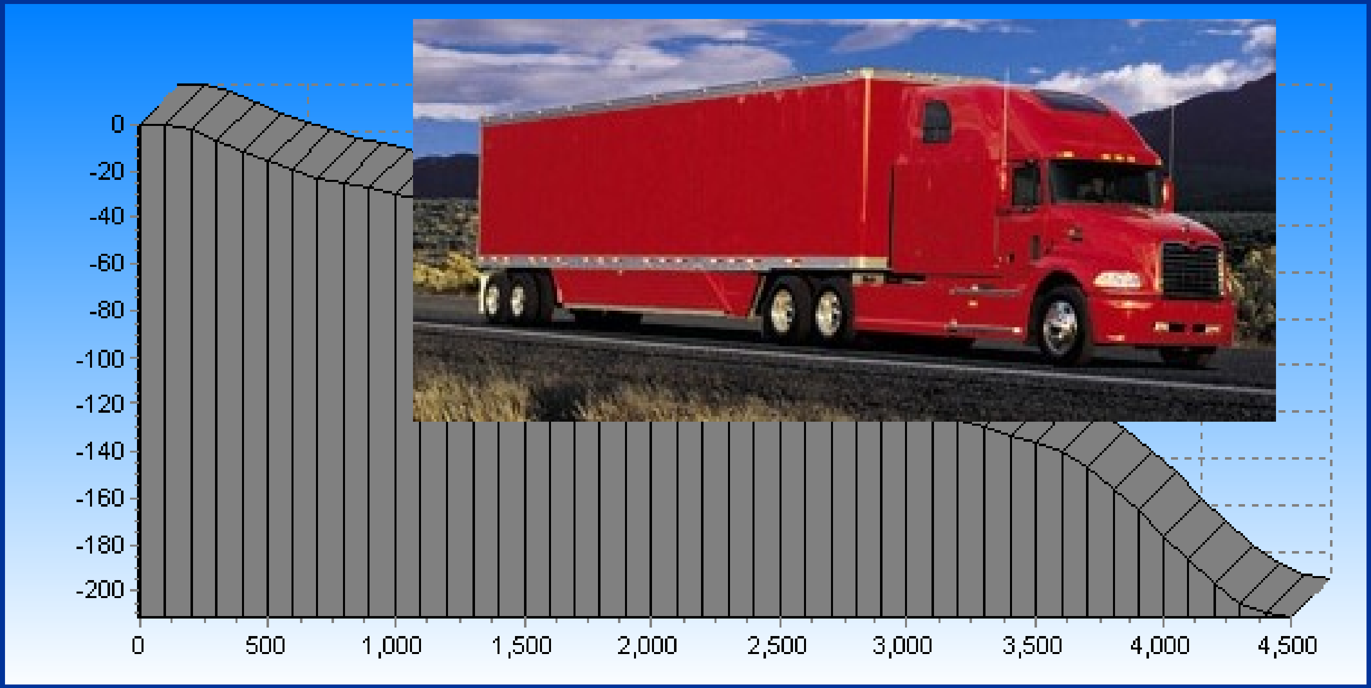
Application 5

3D interpretation of images

- Evolving 3D model interpretation of images using graphics hardware
  - F. Lindblad, P. Nordin, and K. Wolff
  - In Proceedings of the 2002 IEEE Congress on Evolutionary Computation, CEC2002. Honolulu, Hawaii, USA.

Application 6

Optimization of brake utilization for heavy-duty trucks



General reference:
www.me.chalmers.se/~mwahde/AdaptiveSystems/Publications/LingmanWahdeAVEC2002.pdf

Application 7

Driver sleepiness detection

See this link for David Sanberg's Licentiate thesis:
http://vtiextweb.vti.se/11936.epibrw

Final remarks

- Why would one use EAs for optimization?

  -EAs can handle **non-differentiable objective functions**. Classical optimization methods usually requires information about 1:st or 2:nd order derivatives.

  -EAs are suitable for optimization problems that lack a complete mathematical model: EAs can be used with objective functions whose values can only be obtained as a result of a (costly) **simulation**, or using **hardware-in-the loop** (HIL).

  -EAs can handle complex problems with **many local optima** and a **varying number of variables** (as in optimization of neural networks).

- The power of EAs stems mainly from its **parallel search**. Note, however, convergence towards a suboptimal result can occur!

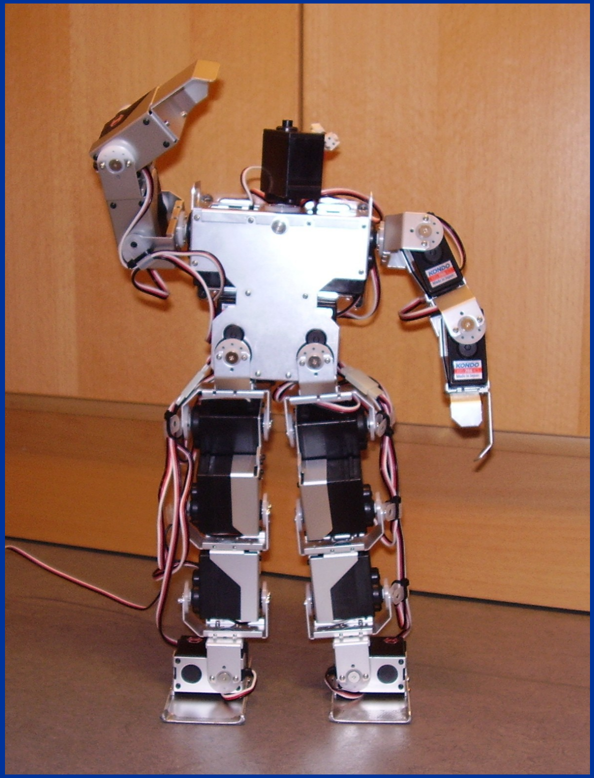
- However, deep knowledge about the problem domain in question is required!

- The **application domain** for EAs is huge->

Application domains GA/GP

- Numerical optimization,Function fitting, data mining, classification, biotechnology, financial market, robotics control, etc., etc.
- A comprehensive overview can be found in tables 12.2-12.6 in:
  - Banzhaf, W. et al (1998). "*Genetic Programming - An Introduction: On the Automatic Evolution of Computer Programs and Its Applications.*" Morgan Kaufmann.

Thanks for your attention!



See the web page for these lecture slides