Applications of stochastic optimization algorithms in the fields of autonomous robots and vehicle engineering

Keynote presentation at INTELLI2012

Mattias Wahde

2012-04-30

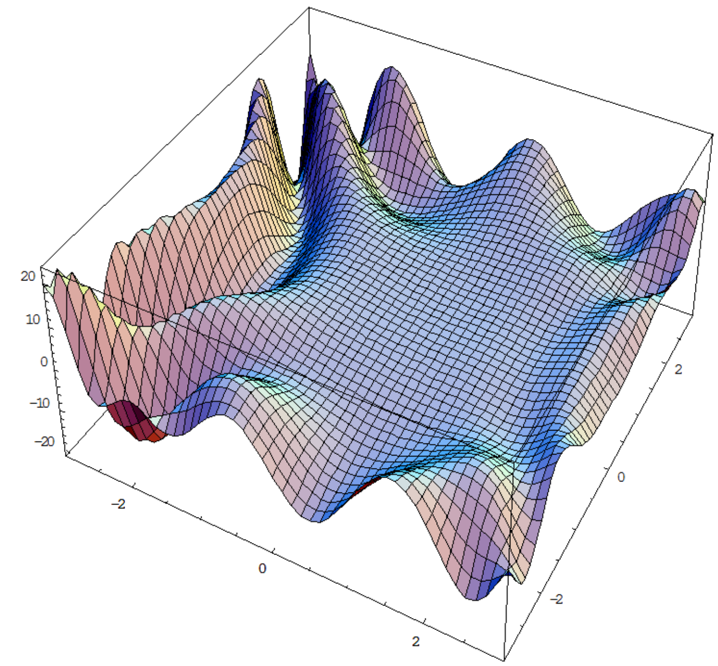
Outline

- Optimization: Brief introduction
  - Classical optimization methods
  - Stochastic (biologically inspired) optimization methods
- Application 1: Brake blending in heavy-duty trucks
- Application 2: Optimization of bipedal (robot) gaits
- Application 3: Sleepiness detection in car drivers.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Optimization: General introduction

- Optimization: The problem of finding the global optimum (maximum or minimum) of an objective function.
- Sometimes, but *not* always, the objective function can be written as a mathematical function in closed form: $f = f(x_1, x_2, \ldots, x_n)$.



Classical optimization methods

- Optimization methods can (loosely) be categorized into classical methods and stochastic methods (often, but not always, biologically inspired).
- Examples of classical methods:
  - Gradient descent
  - Newton's method
  - The method of Lagrange multipliers
  - Simplex method (linear programming)
  - Frank-Wolfe algorithm (quadratic programming)
  - etc. etc.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Limitations of classical optimization

- Classical methods, when applicable, are typically the best choice.
- However, classical methods have limitations.
  - Many (if not most) require *differentiable* objective functions.
  - Some require the objective function to have certain (unlikely, in realistic problems) properties, such as convexity.
- Furthermore, in many optimization problems, either
  - The objective function cannot be specified as an explicit mathematical function. Instead, the evaluation result can only be obtained after, say, a lengthy simulation.
  - The number of *variables* is not fixed (e.g. structural optimization of artificial neural networks).

Biologically inspired opt. methods

- To overcome the limitations of classical optimization, many stochastic optimization methods have been developed.
- Some, but not all, are inspired by biological phenomena.
- There are many examples of biologically inspired optimization methods (henceforth abbreviated BIOMs):
  - Evolutionary algorithms (EAs),
  - Particle swarm optimization (PSO),
  - Ant colony optimization (ACO)
  - etc. etc., and also many versions of each class of method.
- Here, only a few specific examples will be given, focusing on *applications* of the methods, rather than theory.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

BIOMs

- These methods have many advantages, namely
  - They do not require differentiable (or even continuous) objective functions.
  - They do not demand that one should simplify the problem.
  - They are particularly effective in complex search spaces with many local optima.
  - They can handle structural optimization (i.e. problems in which not only the parameters but also the structure of a system (for example, a neural network) is being optimized.
  - They are easy to implement (at least the basic versions).
  - They are almost fully parallelizable.

Why BIOMs?

- Adaptation is a form of optimization.
- Example 1 (one among many): The evolution of shark skin.
- Ridges provide drag reduction, thus allowing very fast swimming.
- The same principles can be applied to reduce drag around aircraft.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

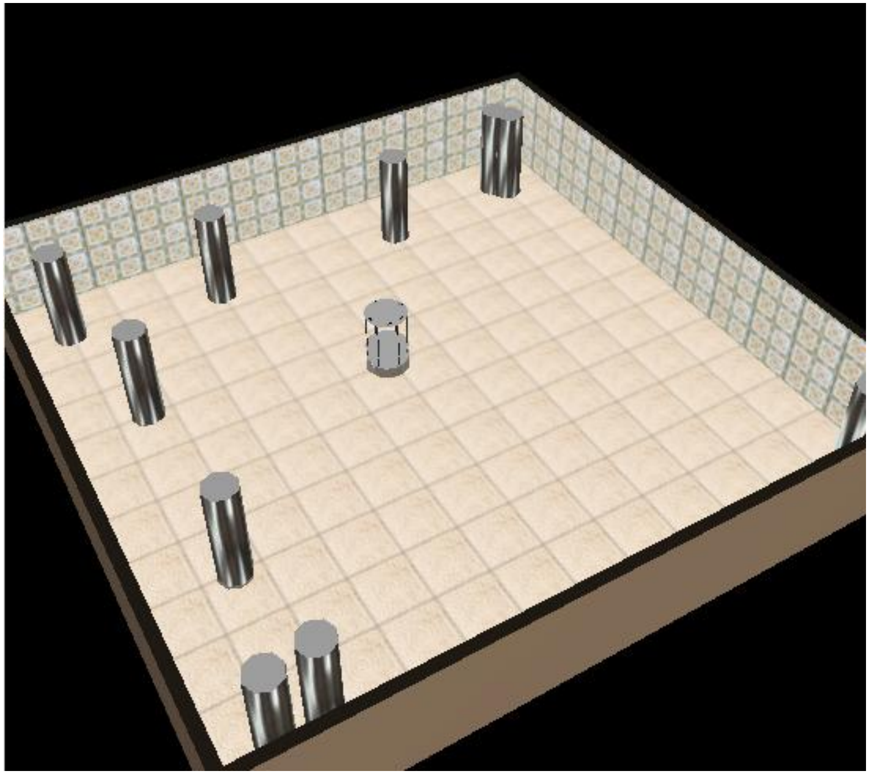
Why BIOMs?

- Example 2: Some species (particularly ants, bees and termites) display very advanced forms of cooperation.
- Specific example: Weaver ants (Oecophylla)

BIOMs: Illustrative example

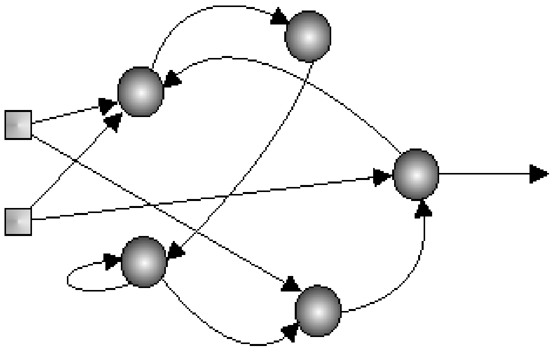
- As an introductory example, consider the (toy) robotics problem defined as follows:

  A robot (equipped with distance (IR or sonar) sensors) is required to move as far as possible without colliding with any movable obstacle (cylinders).



BIOMs: Illustrative example

- Evaluation measure (objective function): Distance moved until collision or maximum time reached.
- The maximum allowed time is gradually increased.
- Optimization method: An evolutionary algorithm
- The robotic brain is structured as a continuous-time (recurrent) artificial neural network:



Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

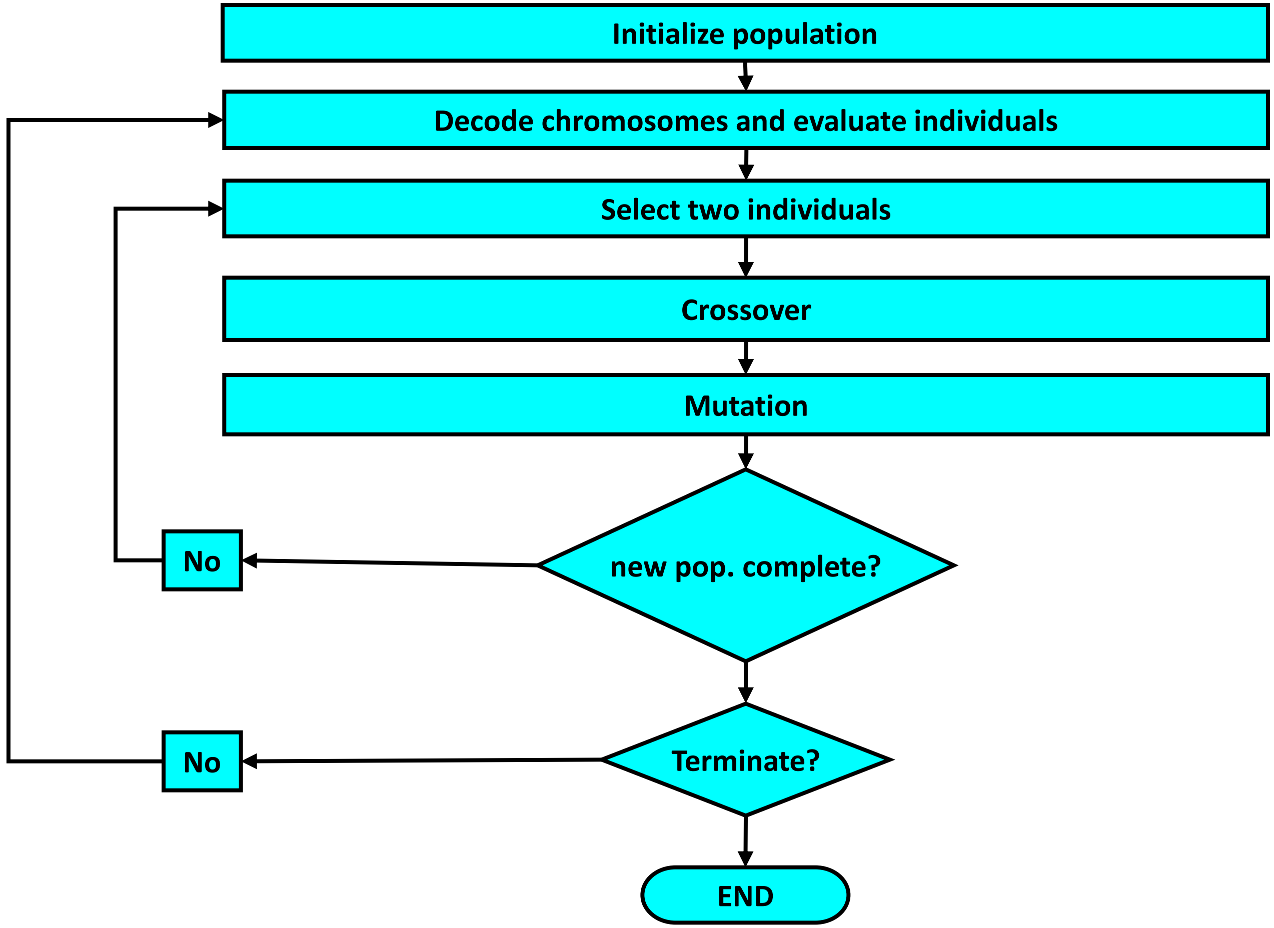
BIOMs: Illustrative example

- Note:
  - It is difficult to assign credit to individual actions – only an overall performance measure can be generated, at the end of an evaluation.
  - Both the structure (number of neurons, connectivity) and the parameters (connection weights) of the network were optimized.

Evolutionary algorithms

- Evolutionary algorithms (EAs) are (loosely) inspired by darwinian evolution – a process of gradual, hereditary change, over long periods of time.
- Population-based optimization method: A set (population) of candidate solutions (individuals) are maintained.
- The population is subjected to evolutionary operators
  - Selection (in proportion to performance (fitness))
  - Crossover (combination of genetic material from different individuals)
  - Mutation (small random changes to an individual's genetic material).

Evolutionary algorithms



Particle swarm optimization

- Based on swarming behavior.
- Many organisms (e.g. some bird and fish species) display swarming behavior.
  - Protection against predators,
  - Efficient food gathering,
  - Mate selection
  - etc.

Particle swarm optimization

1. Initialize positions and velocities of the particles p_i:

   1.1 $x_{ij} = x_{\min} + r(x_{\max} - x_{\min})$, $i = 1, \ldots, N$, $j = 1, \ldots, n$
   1.2 $v_{ij} = \frac{\alpha}{\Delta t}\left(-\frac{x_{\max} - x_{\min}}{2} + r(x_{\max} - x_{\min})\right)$, $i = 1, \ldots, N$, $j = 1, \ldots, n$

2. Evaluate each particle in the swarm, i.e. compute $f(\mathbf{x}_i)$, $i = 1, \ldots, N$.
3. Update the best position of each particle, and the global best position. Thus, for all particles p_i, $i = 1, \ldots, N$:

   3.1 if $f(\mathbf{x}_i) < f(\mathbf{x}_i^{\text{pb}})$ then $\mathbf{x}_i^{\text{pb}} \leftarrow \mathbf{x}_i$.
   3.2 if $f(\mathbf{x}_i) < f(\mathbf{x}^{\text{sb}})$ then $\mathbf{x}^{\text{sb}} \leftarrow \mathbf{x}_i$.

4. Update particle velocities and positions:

   4.1 $v_{ij} \leftarrow v_{ij} + c_1 q \left(\frac{x_{ij}^{\text{pb}} - x_{ij}}{\Delta t}\right) + c_2 r \left(\frac{x_j^{\text{sb}} - x_{ij}}{\Delta t}\right)$, $i = 1, \ldots, N$, $j = 1, \ldots, n$
   4.2 Restrict velocities, such that $|v_{ij}| < v_{\max}$.
   4.3 $x_{ij} \leftarrow x_{ij} + v_{ij}\Delta t$, $i = 1, \ldots, N$, $j = 1, \ldots, n$.

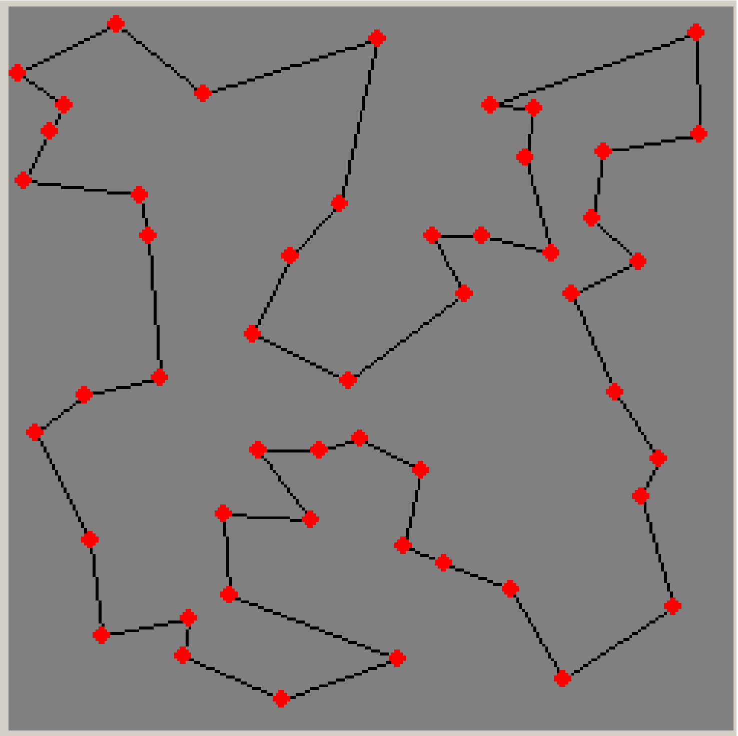
5. Return to step 2, unless the termination criterion has been reached.

Ant colony optimization

- Ants are capable of remarkably efficient discovery of short paths during foraging. How do they achieve this?
- Note that ants
  - ... are (almost) blind,
  - ...have no explicit leaders
- Method: Ants deposit a trail of (volatile) pheromones (a chemical substance) as they move. When choosing a path, ants tend to move in the direction of highest pheromone concentration.

Ant colony optimization

- Ant colony optimization (ACO) is built on the same principles.
- Artificial ants are released on a construction graph, and then move according to probabilistic rules based on artificial pheromone levels, in order to find the shortest path.
- Straightforward example: The travelling salesman problem (TSP).



Outline

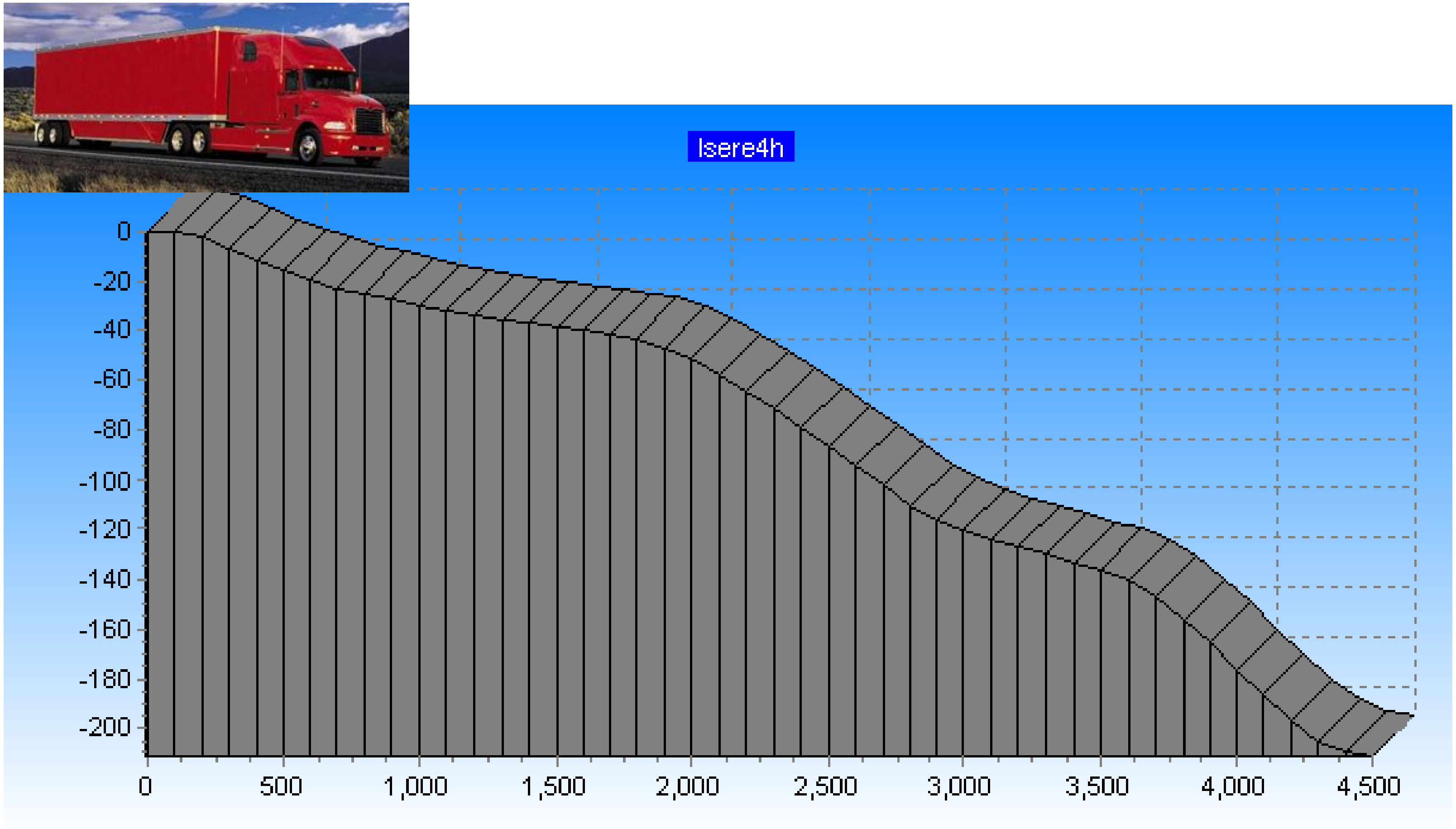
- Optimization: Brief introduction
  - Classical optimization methods
  - Stochastic (biologically inspired) optimization methods
- **Application 1: Brake blending in heavy-duty trucks**
- Application 2: Optimization of bipedal (robot) gaits
- Application 3: Sleepiness detection in car drivers

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Optimization of braking systems

- A common problem (in hilly areas, e.g. parts of France!) is that truck drivers overuse the disc brakes, with overheating and loss of braking capability as a result.
- PhD project (Peter Lingman): Devise an automatic system for optimizing brakes usage to avoid accidents due to brake overheating.
- Cooperation with Volvo Trucks.
- One part of this project was approached using BIOMs.

Optimization of braking systems



Optimization of braking systems

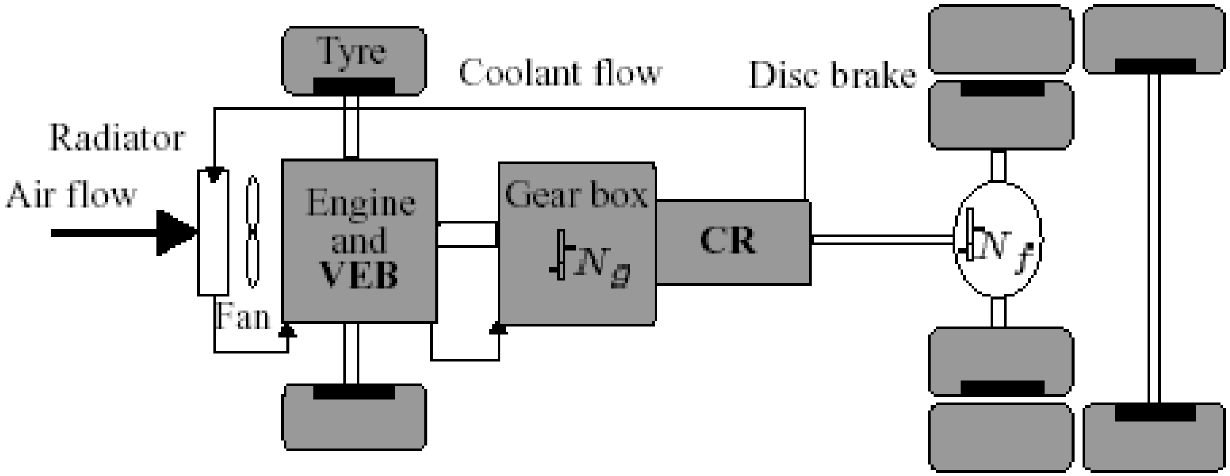


Fig. 1. Volvo retardation system

Optimization of braking systems

- Downhill cruising: select a set speed and maintain it.
- Requires gear changes and activation of pedal brakes (foundation brakes) and auxiliary brakes (e.g. engine brake and compact retarder).
- Problem: too much use of the foundation brakes $\Rightarrow$ overheating $\Rightarrow$ fading $\Rightarrow$ no braking force $\Rightarrow$ accident.
- Overheating of disc brakes: fading
- Overheating of auxiliary brakes: cooling system saturation

Optimization of braking systems

- Possible solutions:
  - Use a lower set speed
    Problem: Slower transportation, road congestion etc.
  - Use auxiliary brakes to save brake pads and discs.
    Problem: high drive tyre wear, leading to high maintenance cost.
- Instead, the correct problem formulation is:

How does one find an optimal strategy for brake blending (usage of different brakes), taking into account constraints, such as brake temperature, speed, engine speed etc.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Optimization of braking systems

- The problem was studied using (initially) a simplified vehicle model. Longitudinal motion:

$$m\dot{v} = F_{\mathrm{drive}} - F_{\mathrm{air}} - F_{\mathrm{roll}} - F_{\mathrm{grade}} - F_{\mathrm{aux}} - F_{\mathrm{found}}$$

- $F_{\mathrm{drive}} = 0$ (for downhill cruising)
- F_{air} ($=c_{\mathrm{air}}v^2$) and F_{roll} ($=c_{\mathrm{roll}}m$) and denote the air and roll resistance, respectively, and $F_{\mathrm{grade}} = mg\sin\alpha$ represents gravity ($\alpha < 0$ for downhill cruising).
- F_{aux} and F_{found} are the forces from the auxiliary and foundation brakes, respectively.
- (+eqs. for brake system and cooling system dynamics, disc pad and tyre wear etc.)

Optimization of braking systems

- The brake blending system used, as input:
  - Vehicle speed (*v*)
  - Current road slope (α)
  - Disc brake temperature (T_1)
  - Coolant temperature ($T_{coolant}$)
  - Engine speed (v_E)
- The required outputs were:
  - Total retardation force request
  - Gear choice (increase, decrease, unchanged)
  - Fraction of braking force taken from foundation brakes
  - Split of braking force from VEB (engine brake) and (CR) compact retarder

Optimization of braking systems

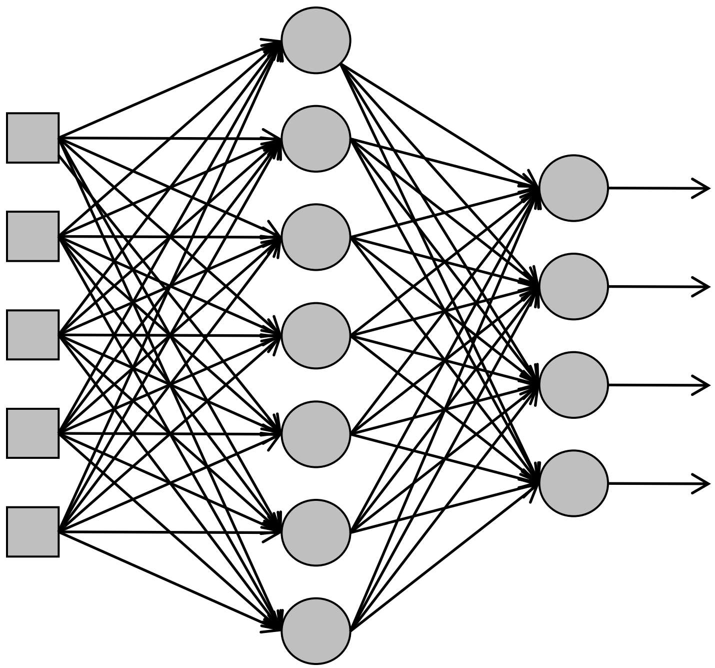
- The constraints were:

  $T_1 < 500$ °C, $v > 5$ m/s, $v < 25$ m/s, $v_E < 2300$ rpm,

  $v_E > 600$ rpm, time $< t_{max}$ (depending on the length of the slope)
- It is difficult to formulate a suitable mathematical model combining the inputs and taking all constraints into account.
- Instead, feedforward neural networks (FFNNs) were used.
- FFNNs, and artificial neural networks (ANNs) in general, are mathematical constructs loosely inspired by the properties of the human brain
- A two-layer FFNN can represent any continuous function.

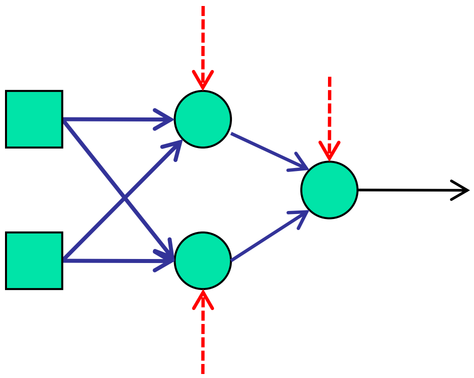
Optimization of braking systems

- Brake blending was implemented using feedforward neural networks (FFNNs)



Optimization of braking systems

- Using a GA to optimize an FFNN. Example: 2-2-1 network

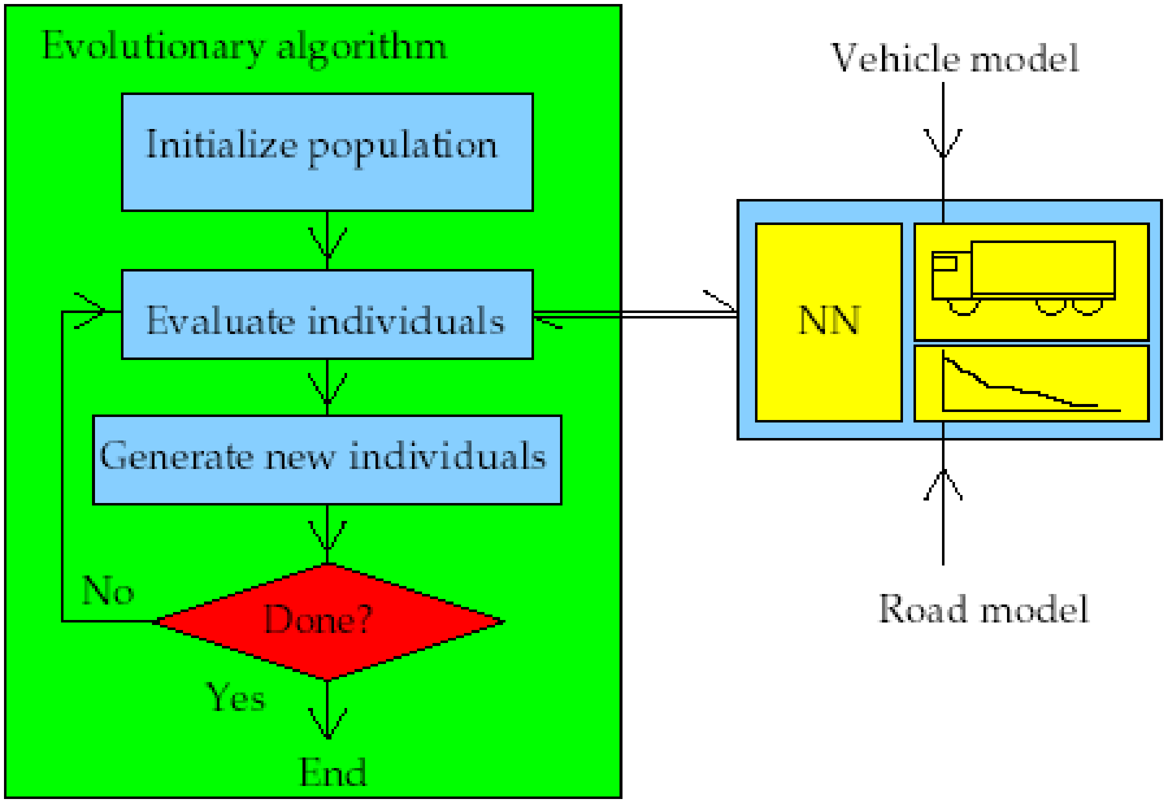


- Encode in chromosomes with 9 real-valued genes:

- Evaluate and assign fitness values. Then apply selection, crossover, and mutation etc.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Optimization of braking systems



Optimization of braking systems

- Two different cases were considered.
  - Achieving highest possible average speed (without violating the constraints)
    - Fitness measure: distance travelled (until the constraints are violated or the truck reaches the end of the slope).
  - Achieving the lowest possible wear cost (without violating the constraints)
    - Fitness measure: $1/C_{tot}$ if the truck runs for the maximum allowed time (t_{max}) and $e^{k(t_0 - t_{max})}/C_{tot}$ if the truck violates the constraints at time $t_0 < t_{max}$, where k is a positive constant and C_{tot} is the total wear cost (pad and tyres)

Optimization of braking systems

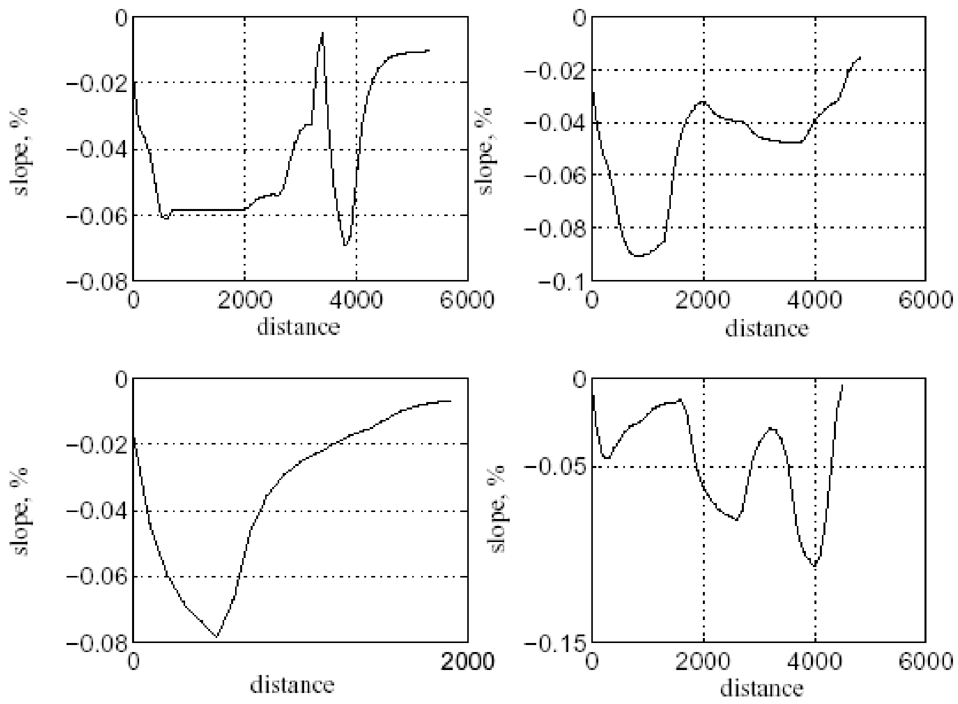
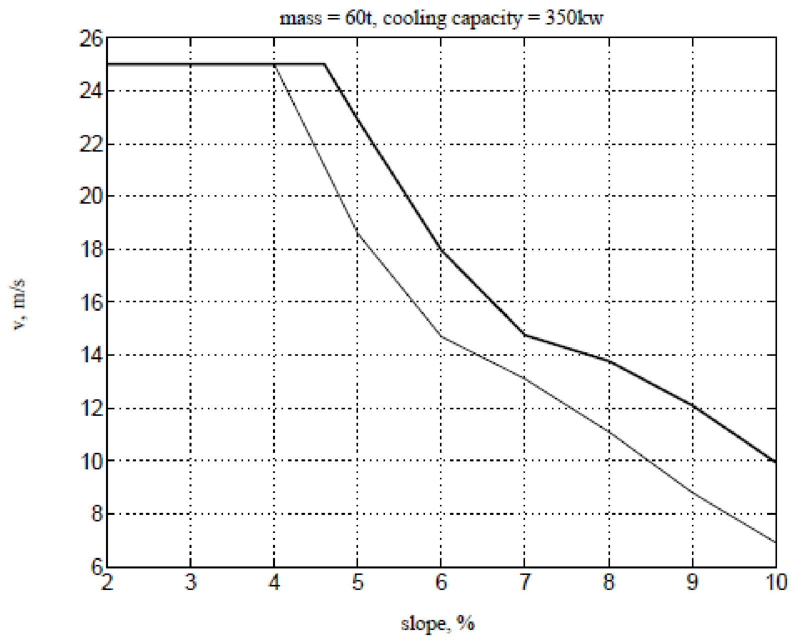


Fig. 5. Example of measured road profiles used. French alps, Isère1-4

Optimization of braking systems

- Results, case 1: Constant slope



Thin line: maximum stationary speed achieved by a skilled driver, selecting optimal gear and (constant) level of auxiliary brake usage. Thick line: maximum stationary speed achieved by the FFNN (varying brake usage optimally).

Optimization of braking systems

- Results, case 1: Varying slope

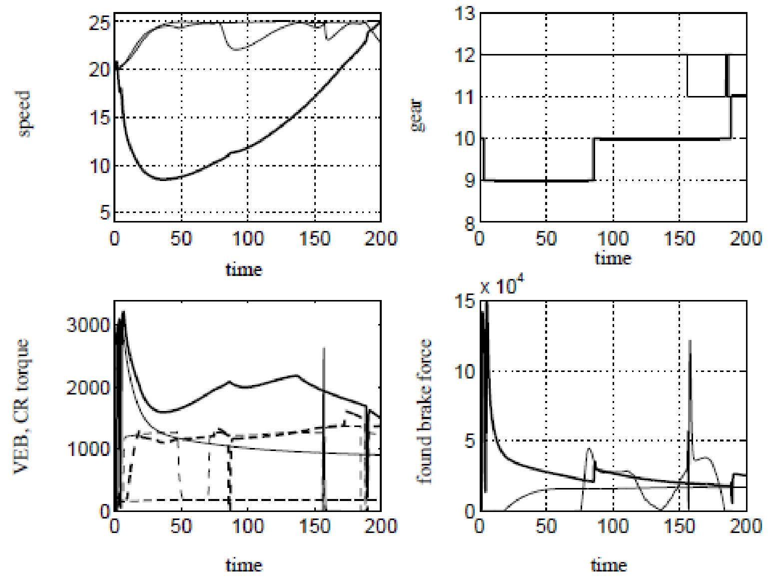
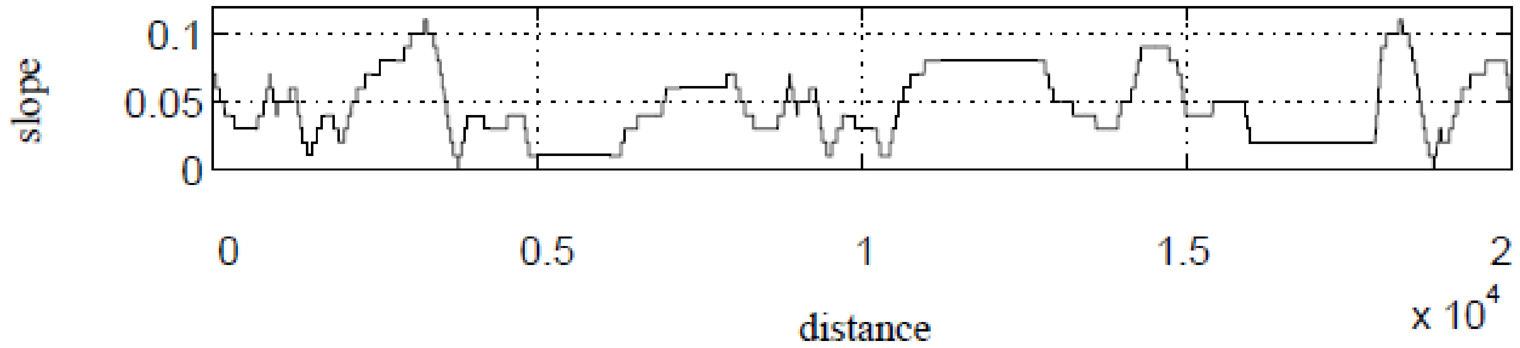


Fig. 3. Optimal blending for high mean speed on 3 roads. Thick solid line: 10% constant slope, Medium solid line: Isère 4 road, Thin solid line: 5% constant slope

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Optimization of braking systems

- Problem: overfitting (adaptation to specific conditions of the training road).
- Solution: Construct a long, artificial road, containing as many relevant aspects as possible (using parts from the French alps (Isère), and Kassel hills in Germany).



Optimization of braking systems

- After training, the best networks found in each generation were applied to 14 test slopes.
- Two generalization measures were defined:

$$G = \frac{1}{N_\mathrm{r}} \sum_{i=1}^{N_\mathrm{r}} \frac{d_i}{L_i}; \; V = \frac{1}{N_\mathrm{r} v_\mathrm{max}} \sum_{i=1}^{N_\mathrm{r}} \overline{v}_i$$

- d_i = distance travelled on slope *i*, L_i = length of slope *i*.

Optimization of braking systems

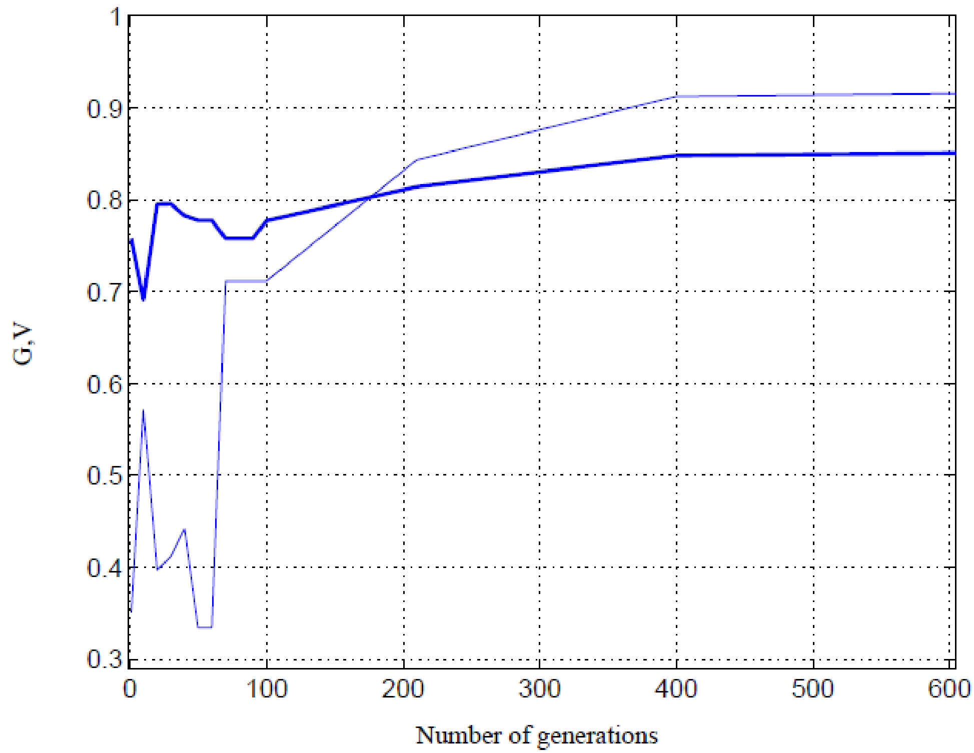


Fig. 6. Generalization measures, G and V. Thick solid line: V, Thin solid line: G

Optimization of braking systems

- Results, case 2: minimizing wear cost
- From the figure one can, for example, note that a *very* experienced driver can handle a 5% slope using auxiliary brakes only, at a wear cost of 1.1€.
- The FFNN controller, by contrast, can handle the same slope at a wear cost of only 0.38€, through optimal brake blending.

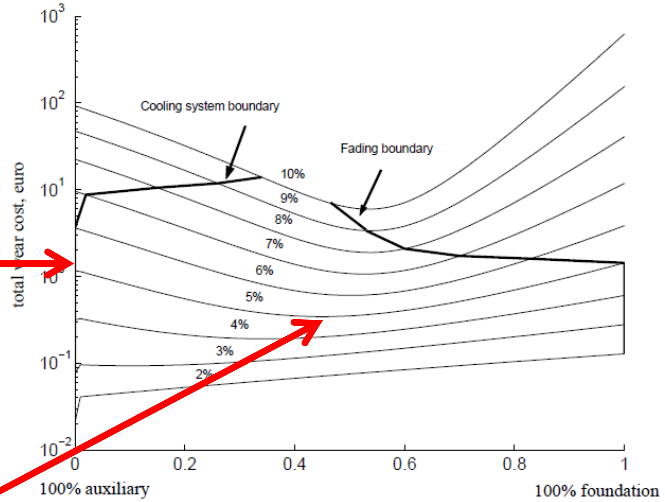


Fig. 9. Vehicle of 60 t and 6 axles travelling at constant speed 15 m/s down a 3000 m descent for 9 different slopes

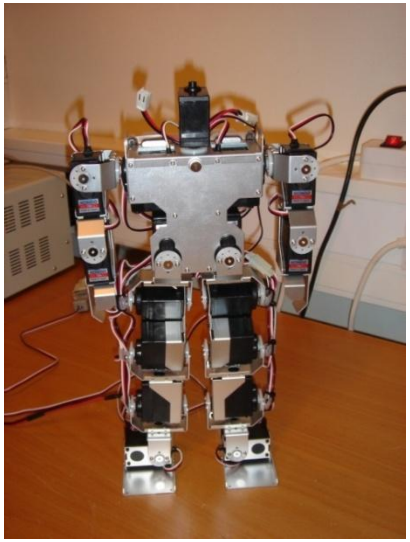
Reference: Lingman, P. And Wahde, M. *Transport and maintenance effective retardation control using neural networks with genetic algorithms*, Vehicle System Dynamics, 42, 89-107, 2004

Outline

- Optimization: Brief introduction
  - Classical optimization methods
  - Stochastic (biologically inspired) optimization methods
- Application 1: Brake blending in heavy-duty trucks
- **Application 2: Optimization of bipedal (robot) gaits**
- Application 3: Sleepiness detection in car drivers

Optimization of bipedal gaits

- **Humanoid robots**
  - A special case of autonomous robots
  - Humanoid shape⇒
    - Easy adaptation to environments designed for people.
    - Capability of walking in stairs etc.
    - Easier to accept, on a social level (less intimidating etc.)
    - ...but more difficult to control (not statically balanced etc.)

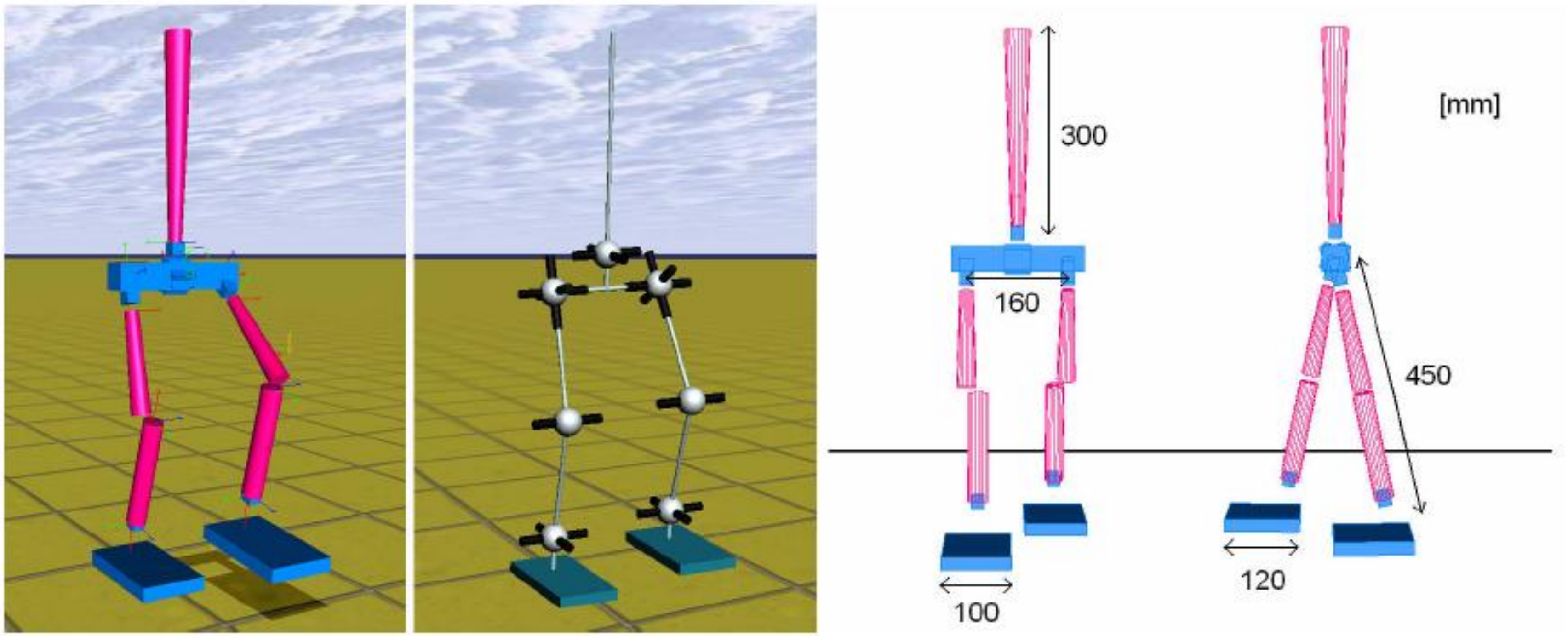


Optimization of bipedal gaits

- Central pattern generators (CPGs)
  - Oscillatory neural circuits, found in the brain of many animal species, including mammals.
  - Involved in oscillatory phenomena such as crawling, walking etc.
  - Observations support the notion of CPGs in humans: treadmill training of patients with spinal cord lesion.

- Objective: Use a CPG network to generate a reliable gait (walking pattern) for a bipedal robot.

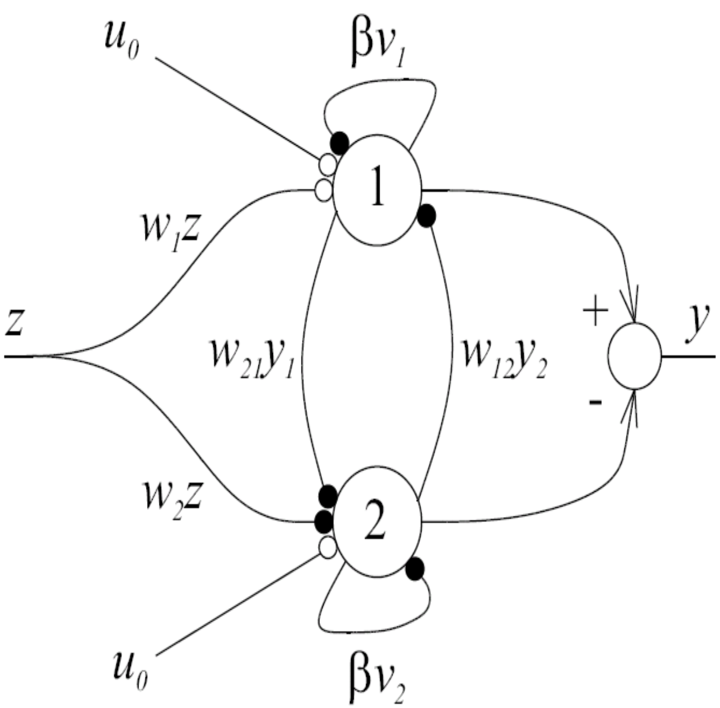
Optimization of bipedal gaits



Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

The Matsuoka oscillator

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

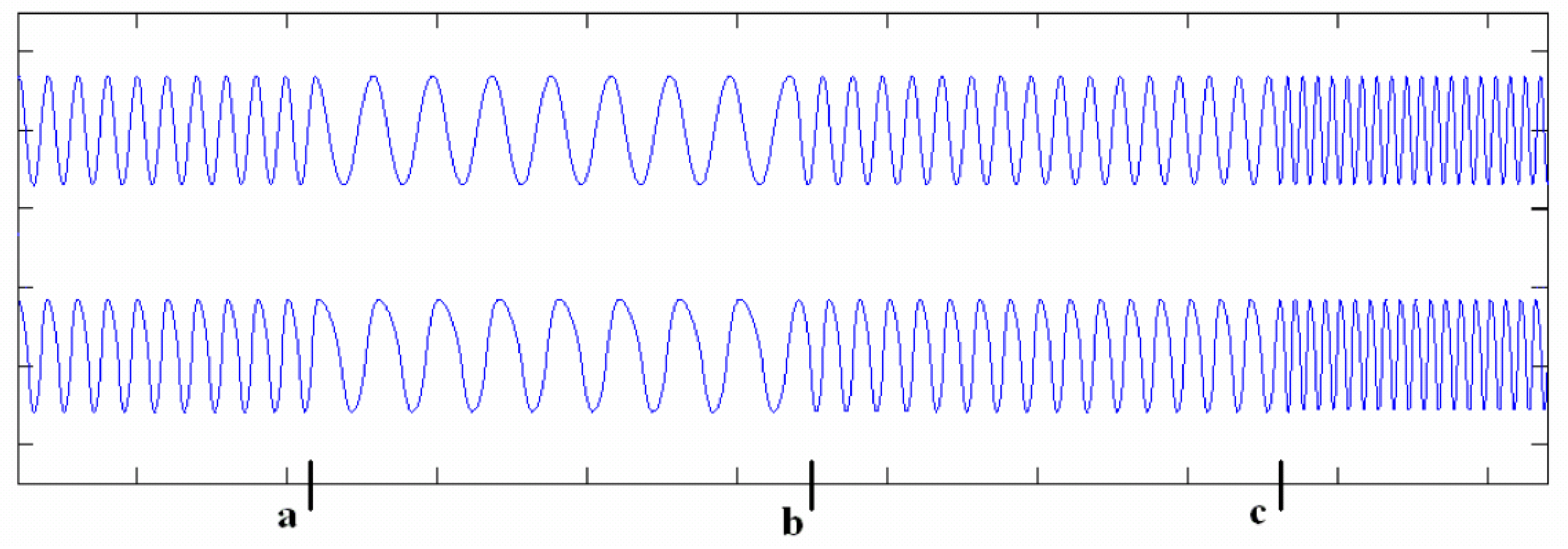


u_i = inner state
v_i = degree of self-inhibition
τ_u and τ_v = time constants
u_0 = bias (tonic input)
w_{ij} = connection weights
y_i = output

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

The Matsuoka oscillator

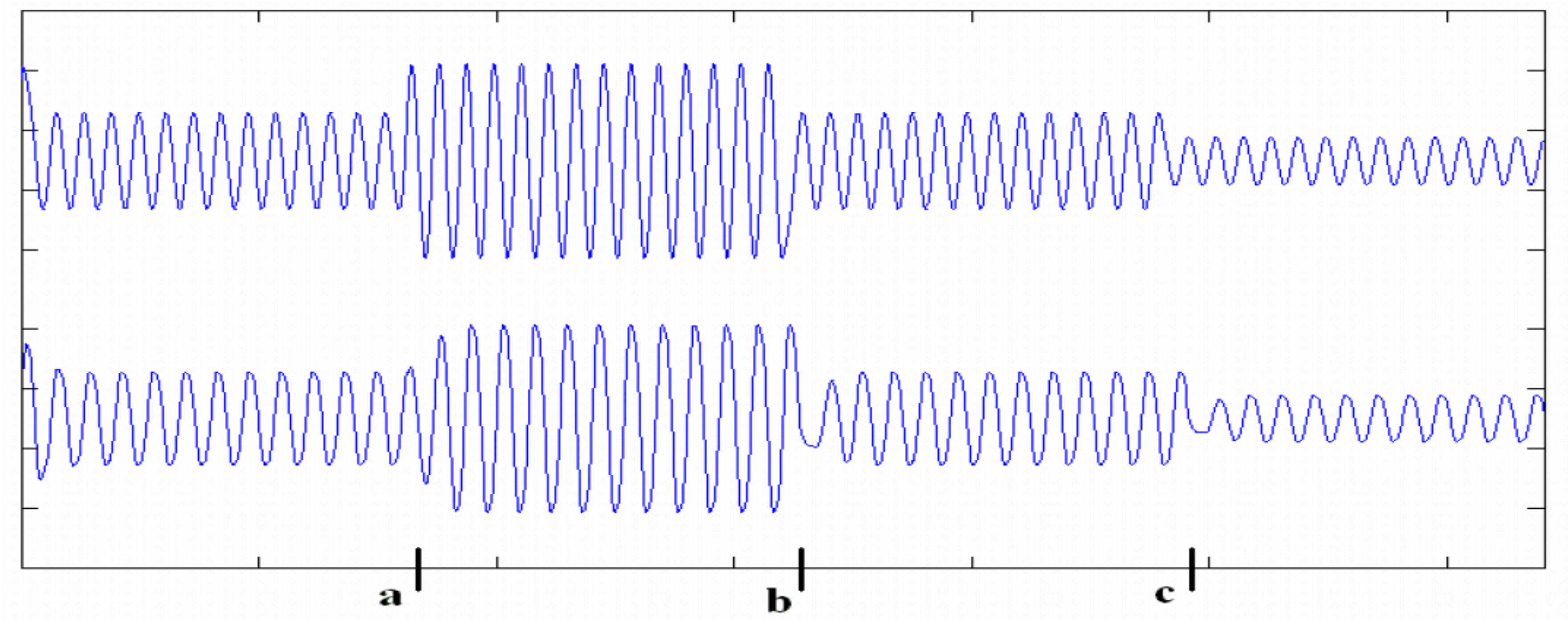
- Frequency variation when the time constants τ_u and τ_v are varied:



Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

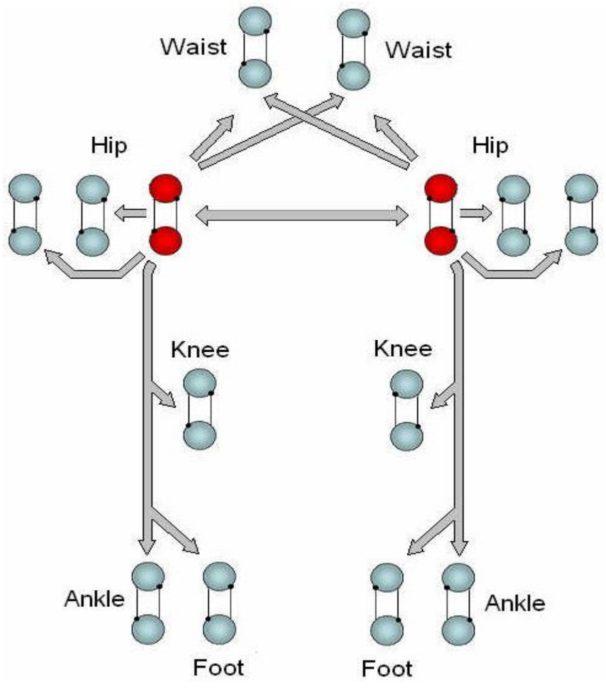
The Matsuoka oscillator

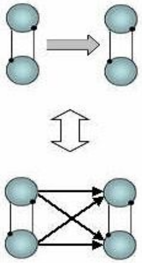
- Amplitude variation when the bias u_0 is varied:

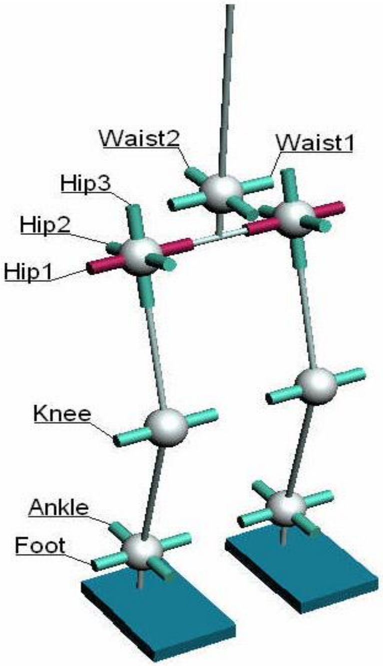


Optimization of bipedal gaits

- Arrows indicate a *possible* connection



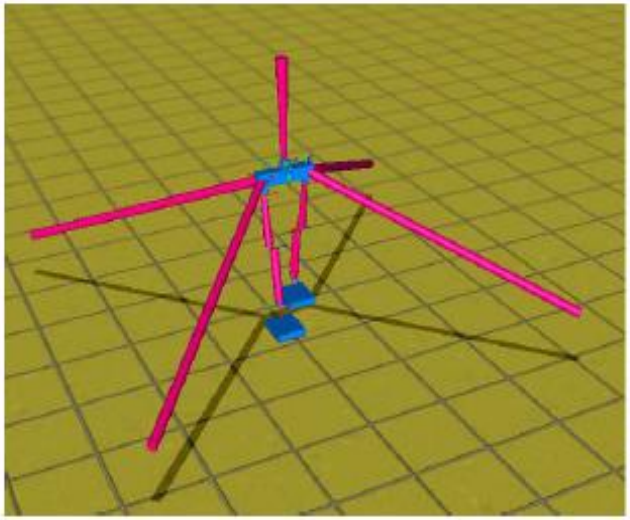


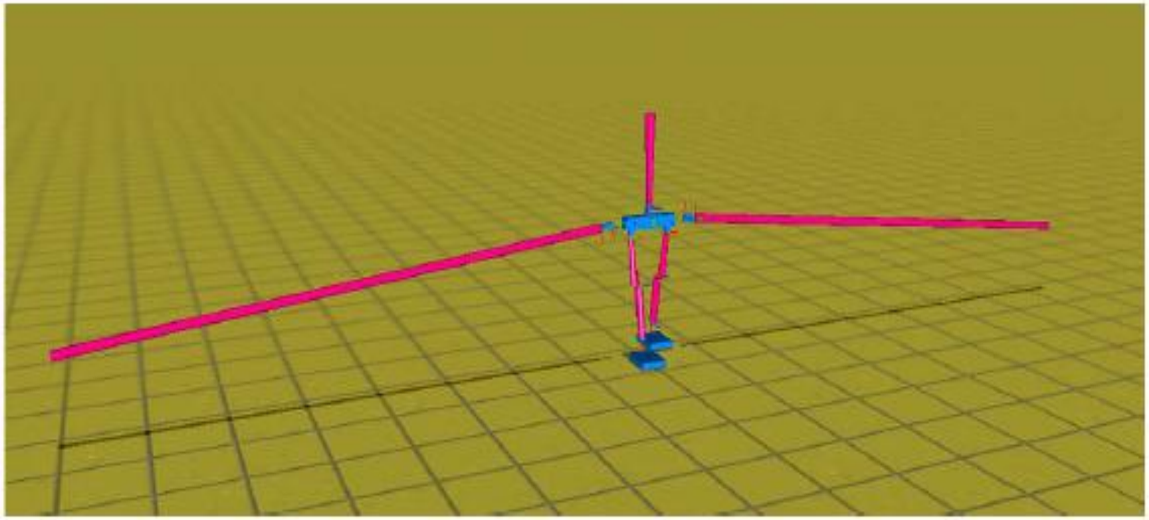


Optimization of bipedal gaits

- The CPG network (structure and parameters) were optimized using a genetic algorithm.
- Difficult problem: Choosing a fitness measure.
- If the naïve measure (horizontal center-of-mass position at the end of the evaluation) is used, the GA will quickly find a local optimum in which the robot simply throws its body forward.
- This (useless) local optimum is hard to escape from!
- Solution: Use a baby-walker!

Optimization of bipedal gaits

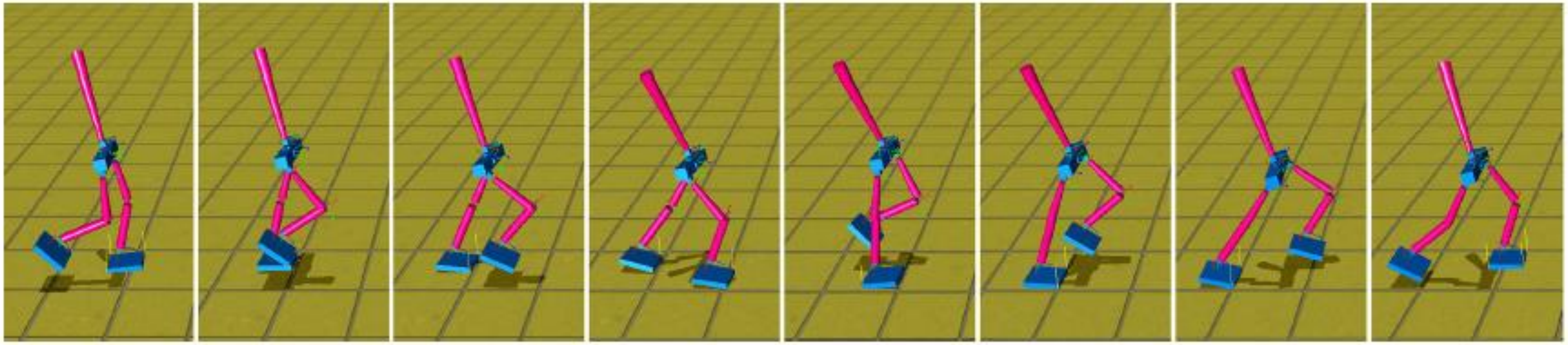




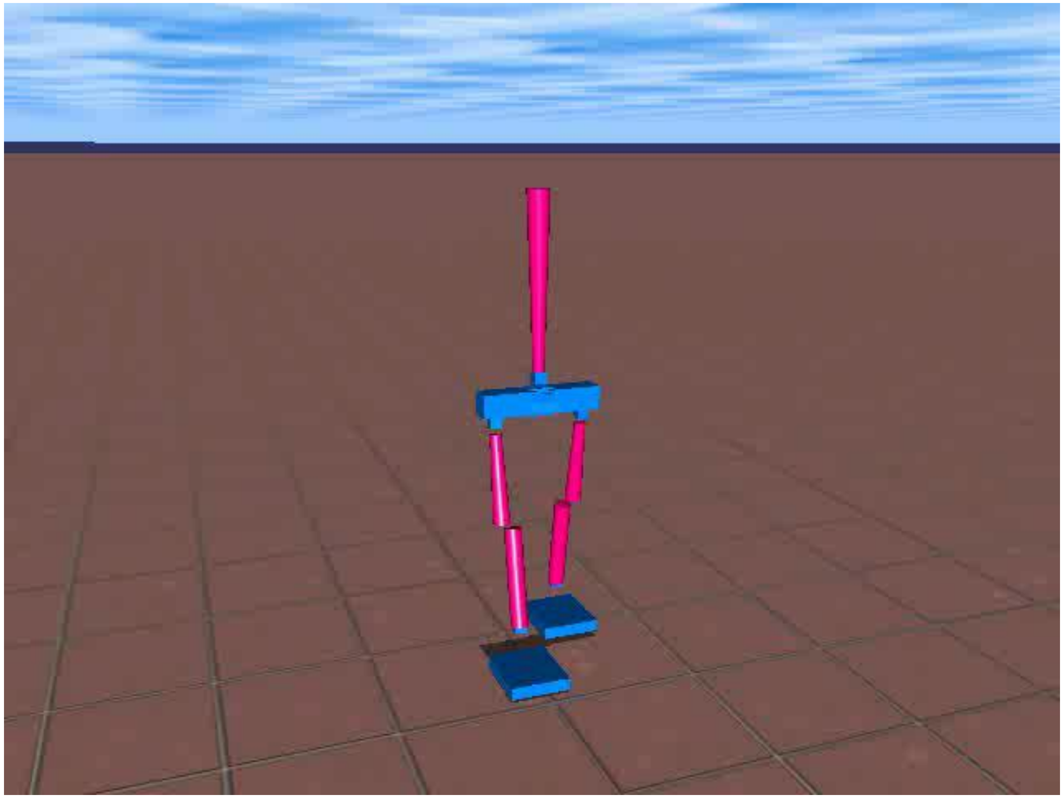
- The (massless) support structure provides a correcting force if the robot begins to fall.
- It is gradually removed during evolution.

Optimization of bipedal gaits

- The optimization was carried out in a high-fidelity simulator developed in my research group.
- Results: A very robust bipedal gait was found, allowing the (simulated) robot to walk essentially indefinitely:

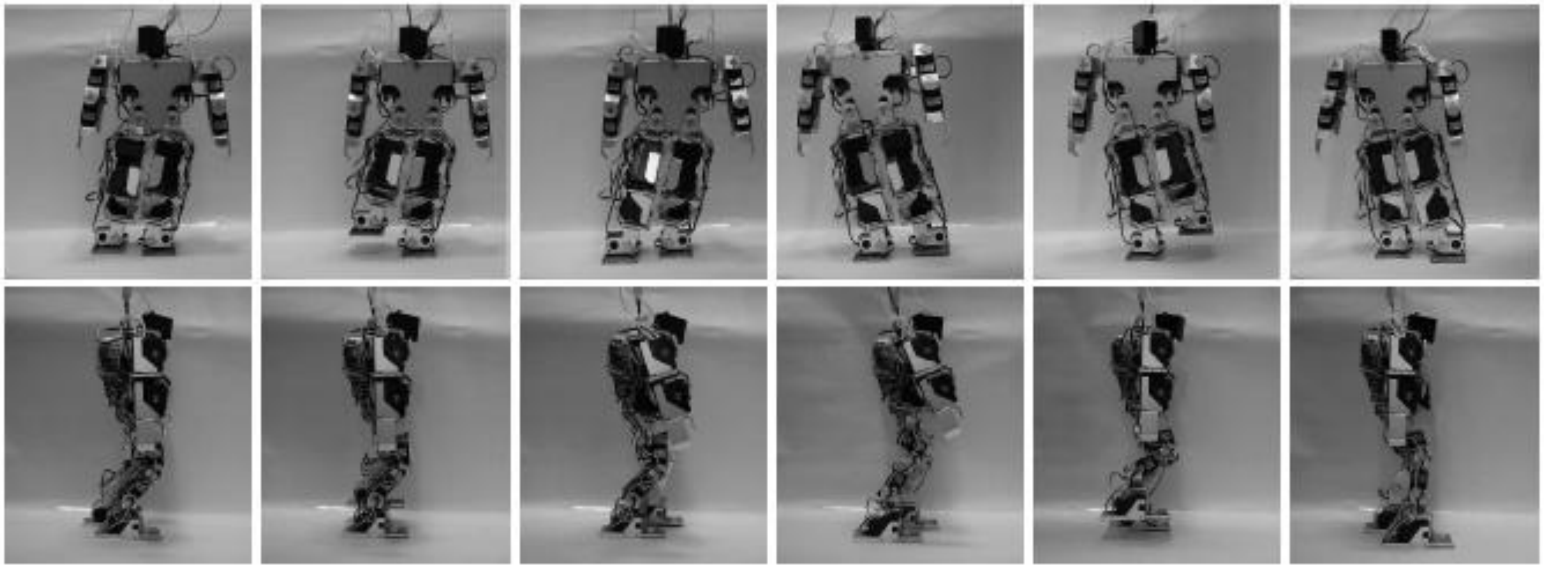


Optimization of bipedal gaits



Optimization of bipedal gaits

- The results were then transferred to a real robot (after some adaptation to the servo motors used in the real robot)



Reference: Heralic, A., Wolff, K., and Wahde, M. *Central pattern generators for gait generation in bipedal robots,* invited book chapter in *Humanoid Robots*, Ed. Hackel, M., Advanced Robotic Systems, Vienna, pp. 285-304, 2007

Outline

- Optimization: Brief introduction
  - Classical optimization methods
  - Stochastic (biologically inspired) optimization methods
- Application 1: Brake blending in heavy-duty trucks
- Application 2: Optimization of bipedal (robot) gaits
- **Application : Sleepiness detection in car drivers.**

Opt. of driver sleepiness detection

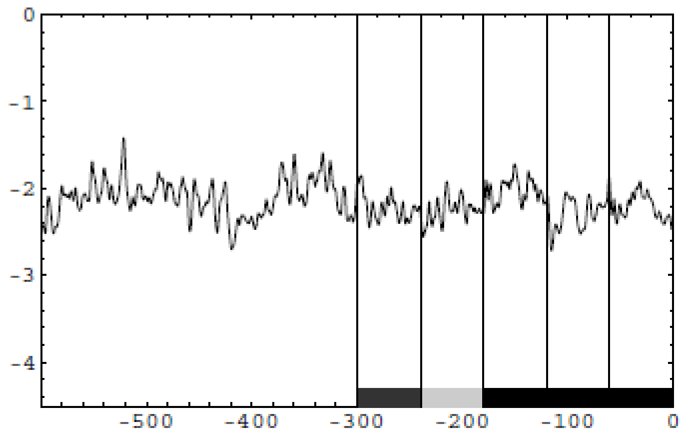
- Driver sleepiness accounts for a large fraction (up to 15-20%) of road accidents.
- Driving while sleepy can be compared with driving under the influence of alcohol.
- Thus, finding ways of detecting driver sleepiness (and, ultimately, also giving timely warnings to the driver) it thus a highly relevant topic.

Opt. of driver sleepiness detection

- In general, it has been observed that sleepy drivers tend to oscillate more in the driving lane, than alert drivers:

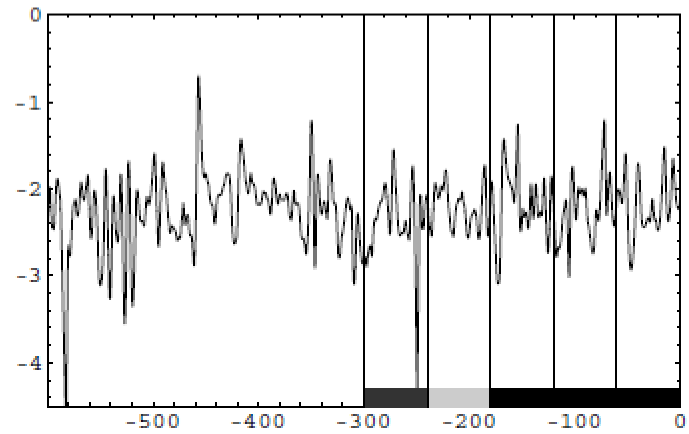
Lateral position

Alert driver (example)



Time

Sleepy driver (example)



Time

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Opt. of driver sleepiness detection

- Thus, one may use the **standard deviation of lateral position**, over a suitable interval of time, as an *indicator* of driver sleepiness:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}},$$

  where x_i is the lateral position of the vehicle for sample *i*.
- Unfortunately, the difference between alert and sleepy drivers (in terms of σ) is rarely as clear cut as in the example on the previous page.
- Furthermore, there are great variations between individuals.

Opt. of driver sleepiness detection

- Other sleepiness indicators have been introduced as well, e.g. the **percentage of lane exits** (Lanex):

$$L = \frac{\sum_{i=1}^{N} \theta(x_i)}{N},$$

where

$$\theta(x_i) = \begin{cases} 1 & \text{if } x_i > x_{\mathrm{L}}, \\ 1 & \text{if } x_i < x_{\mathrm{R}}, \\ 0 & \text{otherwise} \end{cases}$$

x_L and x_R denote the left and right limits of the lane, respectively.

Opt. of driver sleepiness detection

- Other indicators include
  - (i) time to lane crossing (TLC),
  - (ii) steering wheel reverals,
  - (iii) rapid steering wheel movements
  - etc.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Opt. of driver sleepiness detection

- In one of the studies, we wanted to combine various indicators to form a sleepiness detection system (SDS), using the indicators as inputs.
- In order to optimize the SDS, a reference (objective function) is needed.
- Several references of sleepiness can be used, e.g. physiological signals and subjective estimations of sleepiness.
- Here, subjective estimations were used, according to the Karolinska sleepiness scale (KSS), which ranges from 1 (very alert) to 9 (very sleepy, fighting sleep).

Opt. of driver sleepiness indicators

- In this particular study, a high-fidelity driving simulator was used.

Opt. of driver sleepiness indicators

- The study involved 12 test subjects who drove either (i) after a normal night of sleep or (ii) after having slept only four hours.
- The test subjects were asked to estimate their level of sleepiness (using KSS) every 10 min, taking into account how they had felt the last 5 min.
- A total of 3,335 data points were obtained (where a data point consists of a KSS estimate and the corresponding time series data, e.g. Lateral position).
- 2,001 data points were used for training, 667 for validation, and 667 for test.

The problem of overfitting

- In almost all real-world problems, particularly those involving experiments, the amount of data is limited.
- The naïve approach of using all the available data for training (optimization) normally results in systems with low degree of predictability since, eventually, the optimization algorithm will start fitting the noise in the data.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

The problem of overfitting

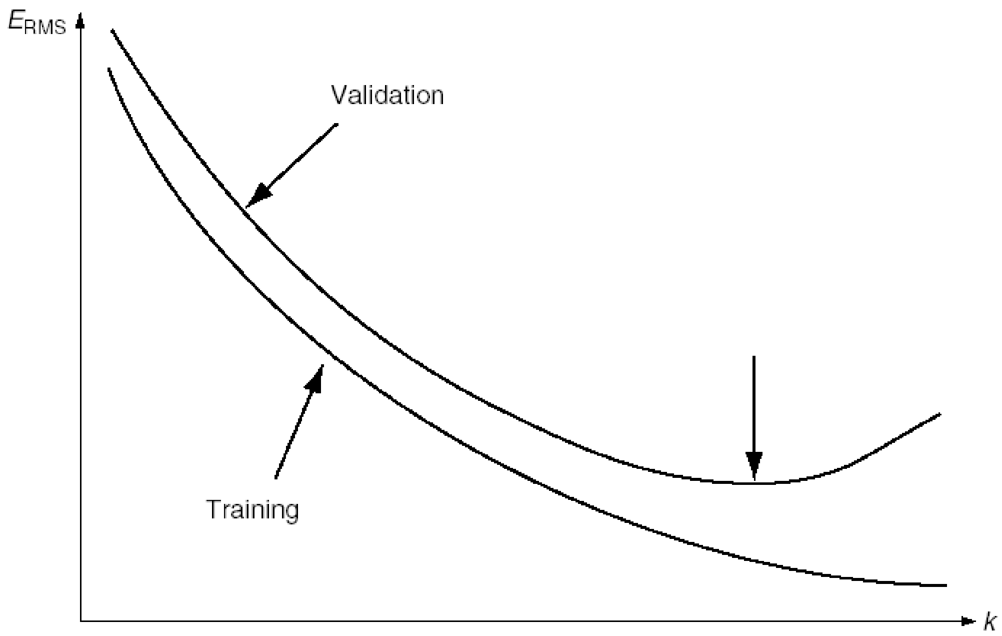
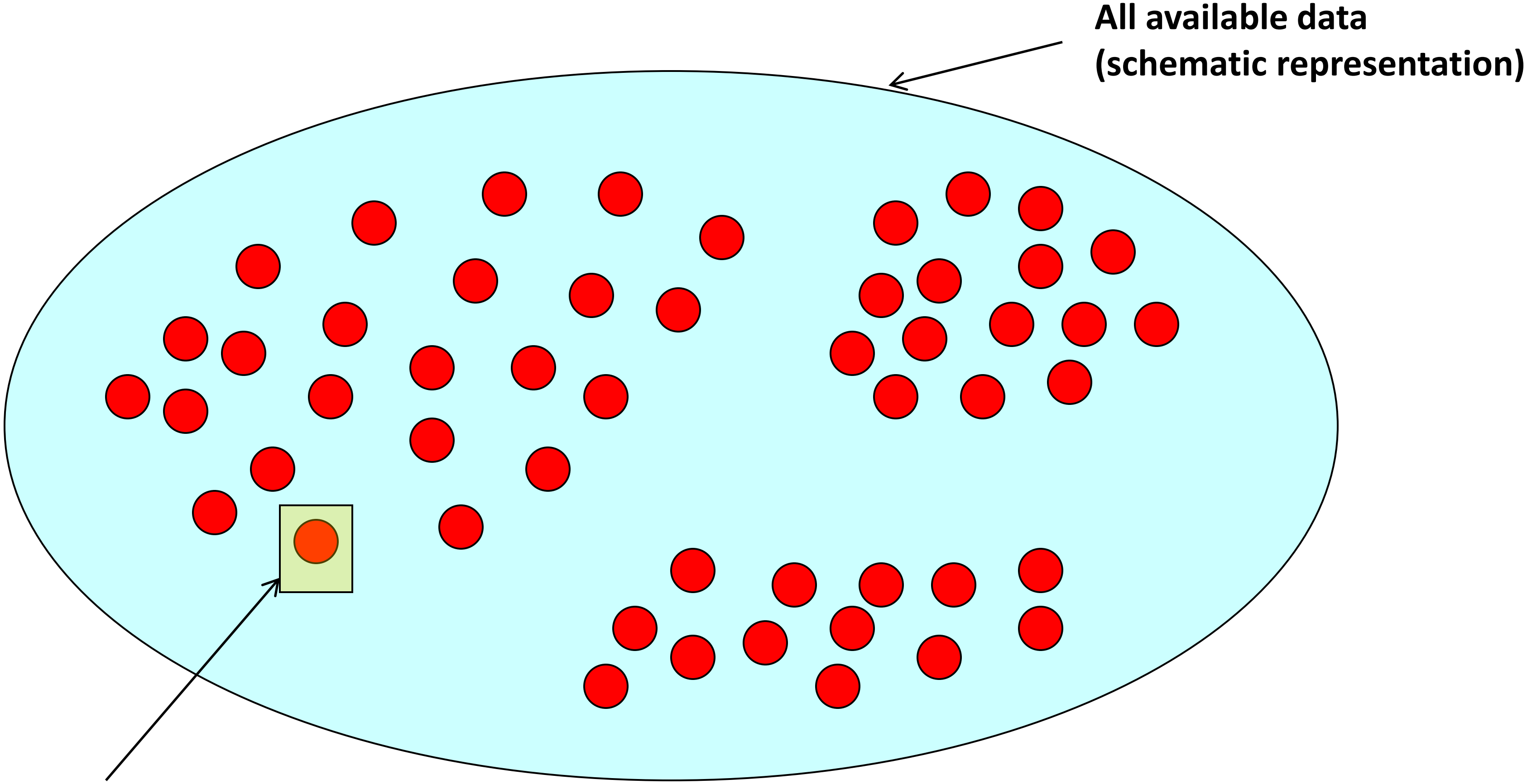


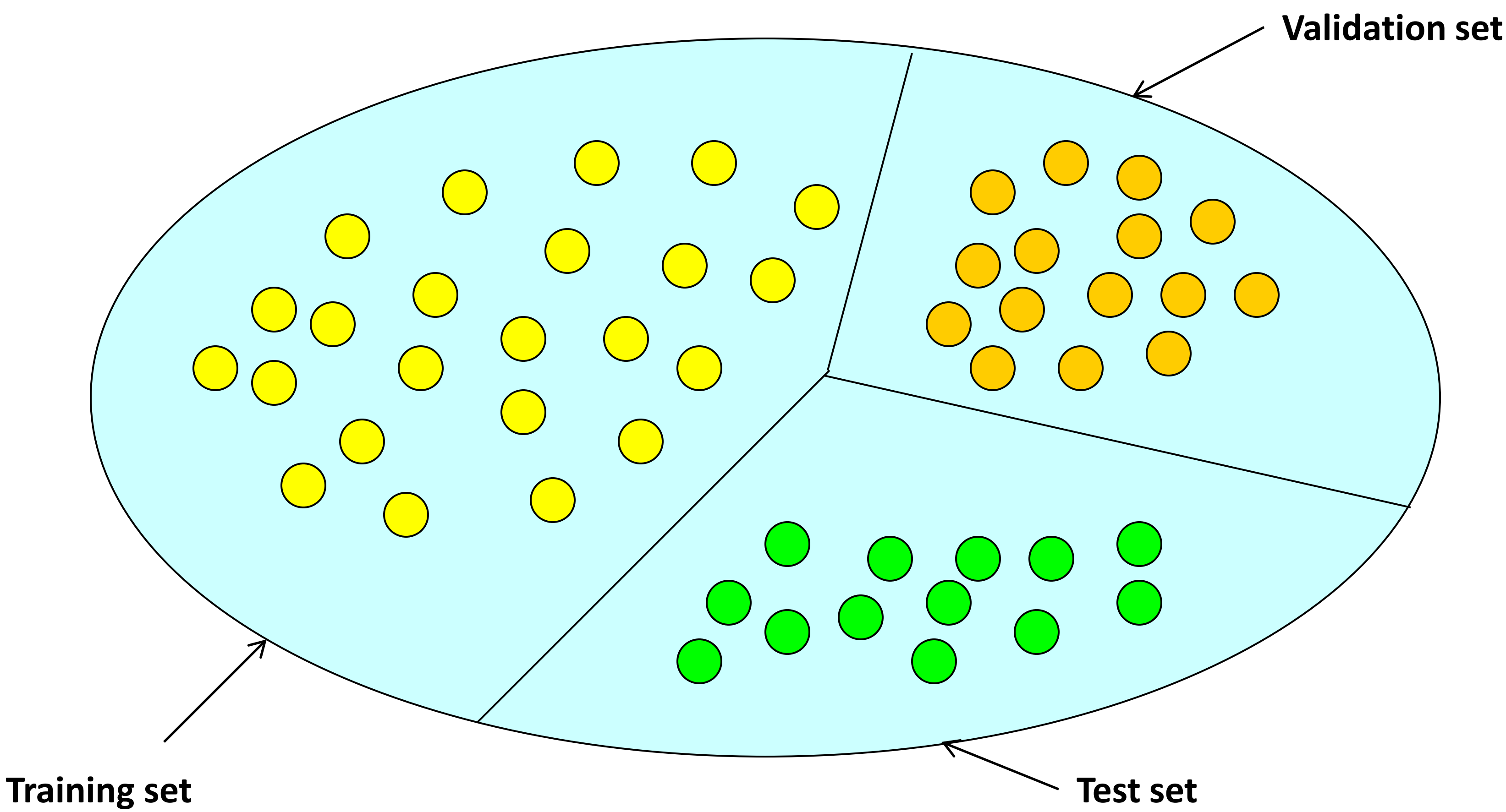
Figure C.4: *An illustration of overfitting. The horizontal axis measures the number of training epochs k. As the training progresses, the training error* E_{RMS}^{tr} *is gradually reduced. The validation error* E_{RMS}^{val} *also drops, in the beginning. However, in later epochs, the validation error begins to rise, as the network starts fitting the noise in the data. The best network is taken as the one having minimal validation error, as indicated by a vertical arrow in the figure. Note that the figure is idealized: in reality, both the training and validation errors typically display a certain amount of noise, making it more difficult to pinpoint the best network.*

The problem of overfitting



Each disc represents an input-output pair.
Specific example. Input: rainfall levels in years T-1,T-2,T-3,T-4,T-5. Output: rainfall in year T.

The problem of overfitting



The problem of overfitting

- Procedure for *avoiding* overfitting:
  - Use the **training data** to give feedback to the training algorithm.
  - Use the **validation data** to determine when to stop the training – measure the error (or fitness) over the validation set, but do *not* make the result of the measurement available to the training algorithm.
  - Use the (previously unused) **test data** (after completing the training) to test the results.
- A typical division would be to use 60% of the data for training, 20% for validation and 20% for testing.

The problem of overfitting

- Example:

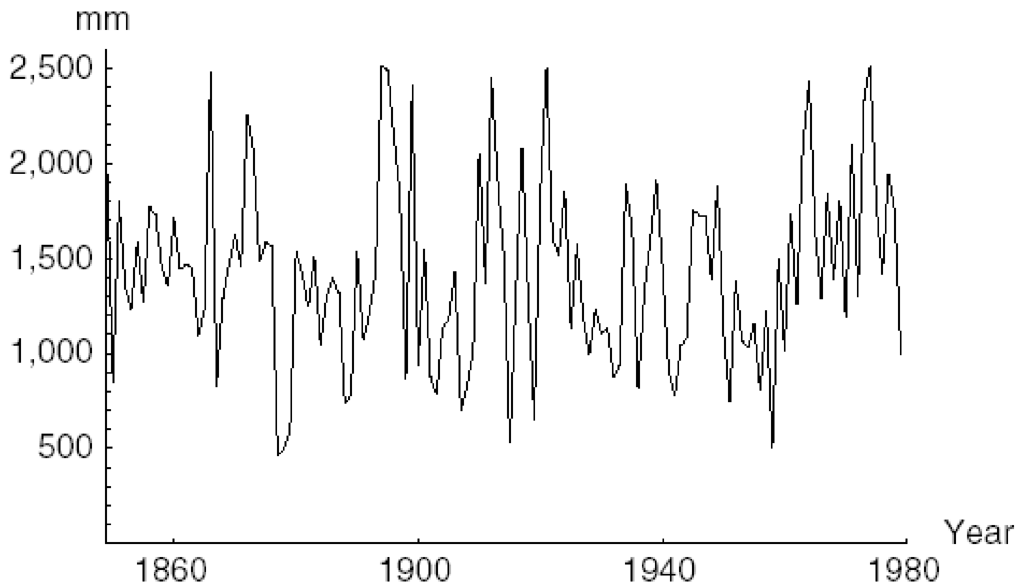


Figure C.5: *The annual rainfall in Fortaleza, Brazil, from 1849 to 1979. See Example C.4.*

The problem of overfitting

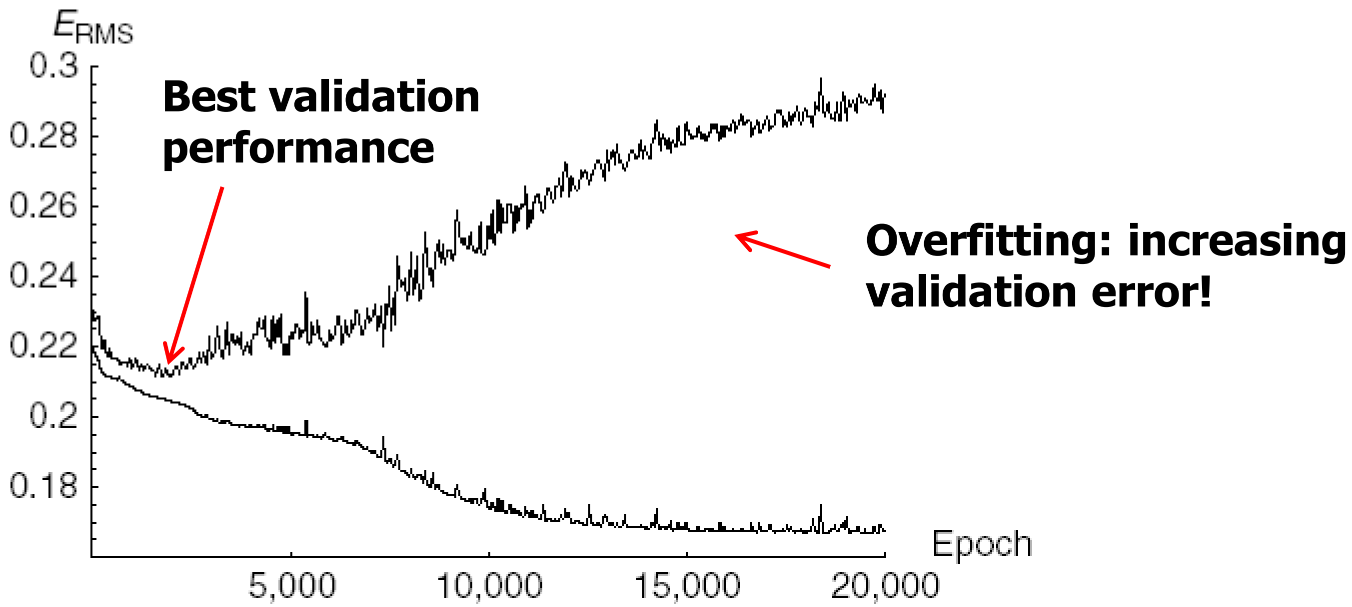


Figure C.6: *An example of overfitting. The lower curve shows the training error for an FFNN being trained to predict the rainfall data in Example C.4, whereas the upper curve shows the validation error.*

Opt. of driver sleepiness indicators

- The optimization problem was cast as a binary classification problem, for which the two classes
  - C_1 **sleepy** (KSS = 8 or 9) and
  - C_2 **alert** (KSS < 7)

  were defined.
- Data points with KSS = 7 were excluded, in order to obtain a clear separation of the classes.

Opt. of driver sleepiness indicators

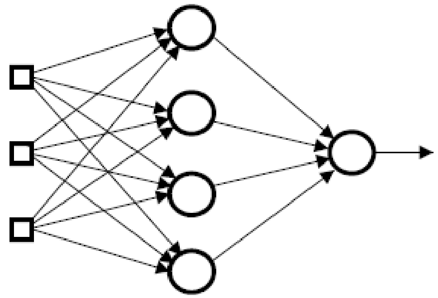
- The objective function (range: [0,1]) was taken as the average, over all test subjects of, of the average of detection sensitivity and detection specificity:

$$f = \frac{1}{M}\sum_{i=1}^{M} f_i \equiv \frac{1}{M}\sum_{i=1}^{M}\left(\frac{\Gamma_i^{\mathrm{sens}} + \Gamma_i^{\mathrm{spec}}}{2}\right),$$

- Where
  - Γ_i^{sens} is equal to the number of correctly classified data points (from test subject *i*) in the sleepy class, divided by the total number of data points in that class.
  - Γ_i^{spec} is equal to the number of correctly classified data points in the alert class, divided by the total number of data points in that class.

Opt. of driver sleepiness indicators

- We chose to use an FFNN to combine the various indicator values.



- The FFNN was optimized using particle swarm optimization.
- Several different combinations of input signals (indicator values) were tried.
- For each combination of input signals, several different network sizes were tested.

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Opt. of driver sleepiness indicators

- Results

Indicator	Training	Validation	Test
Vehicle speed			
I_1	58.8%	57.1%	60.4%
I_2	55.0%	55.1%	55.0%
Lateral position			
I_3	58.9%	58.8%	57.8%
I_4	61.4%	59.3%	61.3%
I_5	68.6%	68.7%	70.7%
I_6	70.5%	69.2%	71.3%
I_7	62.8%	61.2%	62.9%
I_8	65.4%	65.2%	66.3%
Steering wheel angle			
I_9	54.9%	60.6%	55.3%
I_{10}	63.7%	61.4%	63.2%
I_{11}	56.5%	56.3%	55.8%
I_{12}	63.4%	64.1%	63.1%
I_{13}	63.7%	61.4%	63.2%
Yaw angle			
I_{14}	71.2%	68.9%	71.9%
I_{15}	66.4%	64.9%	66.1%

Network name	Training	Validation	Test
N_1	72.2%	71.6%	72.4%
N_2	67.0%	66.2%	63.1%
N_3	73.2%	71.1%	71.1%
N_4	74.1%	74.1%	74.6%
N_5	75.8%	74.5%	74.6%
N_6	77.5%	74.5%	75.5%

Neural networks (combining several indicators)

Individual indicators

Standard deviation of yaw angle

Mattias Wahde, PhD, professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Opt. of driver sleepiness indicators

- The study presented above was carried out a few years ago.
- Since then (after obtaining the necessary permissions!), two ground-breaking studies have also been carried out, using real vehicles (cars and trucks) on real roads. Using both simulator and real road data, we have also considered:
  - Optimizing the individual indicators (not just their combination)
  - Adding a model of sleepiness (the Sleep/Wake predictor, which takes prior sleep into account).
  - Adding also blink behavior time series (in addition to the driving behavior time series).
  - Combining, again using neural networks, these indicators.

Opt. of driver sleepiness indicators

- Our current state-of-the art detection system combines
  - (i) a general variability indicator (developed in my group)
  - (ii) an optimized steering wheel reversal indicator
  - (iii) a blink indicator (Perclos)
  - (iv) the sleep/wake predictor model
- This system achieves a performance score of 0.88 (88%) on previously unseen test data.

Thanks for your attention!

References (papers from my group)

[1] Lingman, P. And Wahde, M. Transport and maintenance effective retardation control using neural networks with genetic algorithms, Vehicle System Dynamics, 42, 89-107, 2004

[2] Wahde, M. A general-purpose method for decision-making in autonomous robots, In: Proceedings of IEA-AIE 2009, LNCS (LNAI), 5579, pp. 1-10, 2009 (Available on my web page, under publications)

[3] Wahde, M. and Sandberg, D. An algorithm for sensory area coverage by mobile robots operating in complex arenas, In: Proceedings of FIRA2009, LNCS **5744**, pp. 179-186, 2009

[4] Heralic, Almir; Wolff, Krister; Wahde, Mattias: *Central pattern generators for gait generation in bipedal robots, invited book chapter in Humanoid Robots*, Ed. Hackel, M., Advanced Robotic Systems, Vienna, pp. 285-304, 2007

[5] Sandberg, D., Wolff, K., and Wahde, M. A robot localization method based on laser scan matching, In: Proceedings of FIRA2009, LNCS **5744**, pp. 171-178,2009

[6] Sandberg, David; Anund, A.; Fors, C.; Kecklund, G.; Karlsson, J. G.; Wahde, Mattias; Åkerstedt, T.: The Characteristics of Sleepiness During Real Driving at Night - A Study of Driving Performance, Physiology and Subjective Experience. Sleep, 34 (10) pp. 1317-1325, 2010

[7] Sandberg, David; Åkerstedt, Torbjörn; Kecklund, Göran; Anund, Anna; Wahde, Mattias: Detecting driver sleeepiness using optimized non-linear combinations of sleepiness indicators . IEEE Transactions on Intelligent Transportation Systems, 12 (1) pp. 97-108, 2010

[8] Wahde, M. et al. Reliable long-term navigation in indoor environments, In: Topalov, A.V. (Ed.) Recent advances in mobile robots, pp. 261-286, 2011 - Available for free download at InTech Publishers, see http://www.intechopen.com/books/recent-advances-in-mobile-robotics/reliable-long-term-navigation-in-indoor-environments

(or Google ”Wahde Navigation InTech”).