

# Real World Optimization of Energy Efficient Vehicle Control

Bastian Lang

Bonn-Rhein-Sieg University of Applied Science

July 21, 2016

- 1 Project Description
- 2 The Simple Model
- 3 NEAT with Simple Model
- 4 Control Program for Velomobile
- 5 Open Tasks

What is the project about?

Creating Energy Efficient Vehicle Controller

# Project Description

Project Description

What is the project about?

Creating Energy Efficient Vehicle Controller

What ML technologies are being used?

ANNs evolved using NEAT

# Project Description

Project Description

What is the project about?

Creating Energy Efficient Vehicle Controller

What ML technologies are being used?

ANNs evolved using NEAT

What is the project based on?

Paper showing ANNs can compete with state-of-the-art approaches ([Gaier and Asteroth, 2014])

## Minimum

- Evolve Energy Efficient Controller with Simple Model

## Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality

## Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality



## Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality

### Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality

### Expected

- Create Data Driven Model

### Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality

### Expected

- Create Data Driven Model
- Evolve Energy Efficient Controller with DD Model

### Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality

### Expected

- Create Data Driven Model
- Evolve Energy Efficient Controller with DD Model
- Evaluate in Reality

### Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality

### Expected

- Create Data Driven Model
- Evolve Energy Efficient Controller with DD Model
- Evaluate in Reality
- Compare Solutions

### Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality

### Expected

- Create Data Driven Model
- Evolve Energy Efficient Controller with DD Model
- Evaluate in Reality
- Compare Solutions

### Minimum

- Evolve Energy Efficient Controller with Simple Model
- Evaluate in Reality
- Compare Simulation vs Reality

### Expected

- Create Data Driven Model
- Evolve Energy Efficient Controller with DD Model
- Evaluate in Reality
- Compare Solutions

### Maximum

- Use Multi-Objective Approach (i.e. Surrogate Modelling)

- 1 Project Description
- 2 The Simple Model**
- 3 NEAT with Simple Model
- 4 Control Program for Velomobile
- 5 Open Tasks



## Time Based Model

$$\frac{ds}{dt} = \begin{pmatrix} t' \\ x' \\ v' \\ W' \end{pmatrix} = \begin{pmatrix} 1 \\ v \\ \frac{F(x,v)}{m} \\ F_u * v \end{pmatrix}$$

Where

- $F_u$ : Force at wheel due to control command
- $F(x, v)$ :  $F_u$  - some drag

TODO: Visualizations of Simulations

- 1 Project Description
- 2 The Simple Model
- 3 NEAT with Simple Model**
- 4 Control Program for Velomobile
- 5 Open Tasks

## Parameters

- Population size: 60
- Maximum Generations: 40
- Speciation algorithm: k-means
- Number of Species: 3
- Drop-off rate: 25
- Dataset of 30/5 tracks (Training/Test)

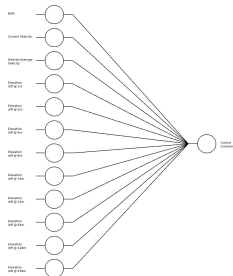


Figure : Initial Network Topology

## On Set of Tracks

- Weighted Sum of Single Track Fitnesses

### On Set of Tracks

- Weighted Sum of Single Track Fitnesses

### On Single Track

- Fitness: Saved Energy - Time Penalty
- Saved Energy: Maximum Energy Consumption - Actual Energy Consumption
- Time Penalty:

$$\begin{cases} 0 & \text{if } neededTime \leq desiredTime \\ (neededTime - desiredTime)^2 & \text{else} \end{cases}$$

- Average Best Fitness
- Average Nr Generations

# NEAT with the Simple Model

Simulations



- ① Project Description
- ② The Simple Model
- ③ NEAT with Simple Model
- ④ Control Program for Velomobile
  - Given Hardware/Software
  - Problems
  - Control Program(s)
- ⑤ Open Tasks

## Hardware

- Velomobile
- Electric Motor (Vivax-Assist)
- Speed Controller (MasterSPIN 75 Pro OPTO)
- Brake Sensor
- Hall Sensor
- Power Sensor
- Simple Button

## Hardware

- Velomobile
- Electric Motor (Vivax-Assist)
- Speed Controller (MasterSPIN 75 Pro OPTO)
- Brake Sensor
- Hall Sensor
- Power Sensor
- Simple Button

## Software

- Run motor on constant speed
- Read brake sensor
- Read hall sensor
- Shuts off above 25km/h
- Shuts off on brake activation
- Shuts off on button press

## Why a Problem?

- Needed for velocity data
- Python-code to read sensor
- C-code to control motor
- Not implemented

## Why a Problem?

- Needed for velocity data
- Python-code to read sensor
- C-code to control motor
- Not implemented

## Solution

- Write/read file in python/C → Synchronization

## Why a Problem?

- Needed for velocity data
- Python-code to read sensor
- C-code to control motor
- Not implemented

## Solution

- Write/read file in python/C → Synchronization
- Write/Read output stream → Python script needs to call C script and resets state

## Why a Problem?

- Needed for velocity data
- Python-code to read sensor
- C-code to control motor
- Not implemented

## Solution

- Write/read file in python/C → Synchronization
- Write/Read output stream → Python script needs to call C script and resets state
- Use socket communication

## Why a Problem?

- Needed for collecting data
- Not implemented

## Solution

- Increase speed on button click
- Shut motor off on brake activation only



## Why a Problem?

- Cannot drive vehicle

## Solution

- (Hardware-)Debug with working initial code
- Only send signal on change
- Range [7,19] instead of [0,100]

## Why a Problem?

- Obviously wrong data gets logged
- No synchronization during data access

# Problems

## Huge Numbers in Log

### Why a Problem?

- Obviously wrong data gets logged
- No synchronization during data access

### Solution

- Synchronize using mutexes → Still huge values

# Problems

## Huge Numbers in Log

### Why a Problem?

- Obviously wrong data gets logged
- No synchronization during data access

### Solution

- Synchronize using mutexes → Still huge values
- Remove all possible multi-threading → Hall Sensor

# Problems

## Huge Numbers in Log

### Why a Problem?

- Obviously wrong data gets logged
- No synchronization during data access

### Solution

- Synchronize using mutexes → Still huge values
- Remove all possible multi-threading → Hall Sensor
- Rewrite in Python → Files empty, occasional restarts

# Problems

## Huge Numbers in Log

### Why a Problem?

- Obviously wrong data gets logged
- No synchronization during data access

### Solution

- Synchronize using mutexes → Still huge values
- Remove all possible multi-threading → Hall Sensor
- Rewrite in Python → Files empty, occasional restarts

# Problems

Huge Numbers in Log

## Why a Problem?

- Obviously wrong data gets logged
- No synchronization during data access

## Solution

- Synchronize using mutexes → Still huge values
- Remove all possible multi-threading → Hall Sensor
- Rewrite in Python → Files empty, occasional restarts

## Open Approaches

- Use C-code with wiringPi synchronization mechanism

TODO: Diagram



- 1 Project Description
- 2 The Simple Model
- 3 NEAT with Simple Model
- 4 Control Program for Velomobile
- 5 Open Tasks**

- Fix Logging
- Evaluate Solutions Simple Model
- Collect Data
- Learn Model
- NEAT on DD Model
- Evaluate Solutions DD Model



Gaier, A. and Asteroth, A. (2014).

Evolving look ahead controllers for energy optimal driving and path planning.

*In Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium on*, pages 138–145. IEEE.