An Automated TORCS Engine Using MLP and Differential Evolution

Erwin Huijzer Jan-Willem Feilzer Mathijs Mul

Computational Intelligence
December 15, 2016

Outline

1. Implementation
2. Design
   - Neural Network
   - Evolution
   - Steering
   - Radius Calculation
   - Recovery
3. Race
4. Improvement

Implementation

- Literature review

Implementation

- Literature review

Driving as a human: a track learning based adaptable architecture for a car racing controller

Jan Quadflieg · Mike Preuss · Günter Rudolph

Implementation

- Literature review

Driving as a human: a track learning based adaptable archite

Jan Qua

Learning the Track and Planning Ahead in a Car Racing Controller

Jan Quadflieg, Mike Preuss, Oliver Kramer, Günter Rudolph

***Abstract*— We propose a robust approach for learning car racing track models from sensory data for the car racing simulator TORCS. Our track recognition system is based on the combination of an advanced preprocessing step of the sensory data and a simple classifier that delivers six types of track shapes similar to the ones a human would recognize. Out of these, establishing a complete track model is straightforward. This model provides an information advantage to controller strategies, as it generally enables *planning*. We demonstrate how such a planning controller can be derived by a mixture of expert knowledge and a simple evolutionary learning approach and give experimental evidence that knowing not only the current conditions but also the big picture of the track is beneficial, as may be expected.**

- Or, they preprocess the sensor data to discretize them into categories that would also be used by human (e.g. rally) drivers when describing the track, and then use a rule-based system to react to the current conditions.

Taking the second approach surely needs more conceptual work at first, but has several advantages. Data summarization to a human understandable level and controlling itself are separate tasks and may also be interchanged between controllers. Besides this implementation advantage, it is also easier to detect where something is wrong if any problem occurs. This would be almost impossible e.g. for an *artificial neural network* (ANN) controller. We conjecture that parti-

Implementation

- Literature review

Driving as a human: a track learning based adaptable archite

Jan Qua

Learning the Track and Planning Ahead in a Car Racing Controller

nter Rudolph

process the sensor data to discretize them
es that would also be used by human (e.g.
when describing the track, and then use a
stem to react to the current conditions.

nd approach surely needs more conceptual
has several advantages. Data summariza-
understandable level and controlling itself
s and may also be interchanged between
es this implementation advantage, it is also
here something is wrong if any problem
d be almost impossible e.g. for an *artificial*
ANN) controller. We conjecture that parti-

Evolutionary Optimization of a Neural Network Controller for Car Racing Simulation

Damianos Galanopoulos, Christos Athanasiadis, and Anastasios Tefas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, 54124, Greece
tefas@aiia.csd.auth.gr

Implementation

- Literature review

Driving as a human: a track learning based adaptable archite

Jan Quad

Learning the Track and Planning Ahead in a Car Racing Controller

nter Rudolph

cess the sensor data to discretize them that would also be used by human (e.g. hen describing the track, and then use a m to react to the current conditions.

approach surely needs more conceptual s several advantages. Data summariza- derstandable level and controlling itself nd may also be interchanged between this implementation advantage, it is also re something is wrong if any problem e almost impossible e.g. for an *artificial* N) controller. We conjecture that parti-

Evo

D

Evolving a rule system controller for automatic driving in a car racing competition

Diego Perez, Yago Saez *Member, IEEE, Gustavo Recio, Pedro Isasi*

***Abstract*— The techniques and the technologies supporting Automatic Vehicle Guidance are important issues. Automobile manufacturers view automatic driving as a very interesting product with motivating key features which allow improvement of the car safety, reduction in emission or fuel consumption or optimization of driver comfort during long journeys. Car racing is an active research field where new advances in aerodynamics, consumption and engine power are critical each season. Our proposal is to research how evolutionary computation techniques can help in this field. For this work we have designed an automatic controller that learns rules with a genetic algorithm. This paper is a report of the results obtained by this controller during the car racing competition held in Hong Kong during the IEEE World Congress on Computational Intelligence (WCCI 2008).**

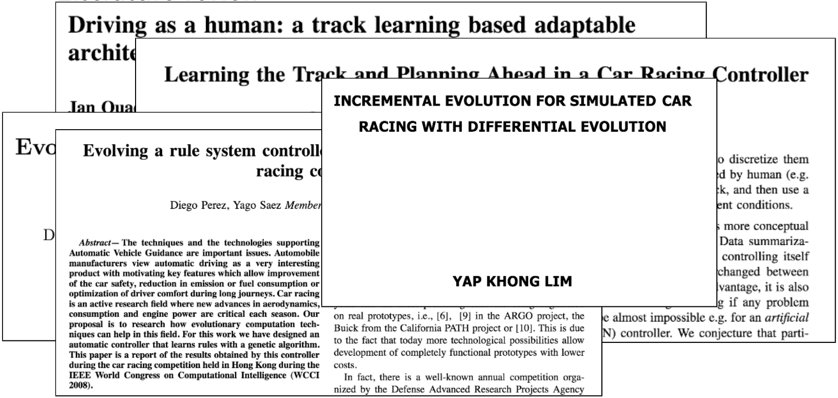
others, robotics, artificial intelligence, computer engineering, telecommunications, signal and image processing, or control and automation techniques in order to develop efficient systems for automatic driving.

Many different approaches have been studied in recent years and the most promising ones are being engineered on real prototypes, i.e., [6], [9] in the ARGO project, the Buick from the California PATH project or [10]. This is due to the fact that today more technological possibilities allow development of completely functional prototypes with lower costs.

In fact, there is a well-known annual competition organized by the Defense Advanced Research Projects Agency

Implementation

- Literature review



Driving as a human: a track learning based adaptable
archite

Jan Quad

Learning the Track and Planning Ahead in a Car Racing Controller

o discretize them
d by human (e.g.
k, and then use a
ent conditions.

more conceptual
Data summariza-
controlling itself
changed between
vantage, it is also
g if any problem
e almost impossible e.g. for an *artificial*
N) controller. We conjecture that parti-

Evo

D

Evolving a rule system controll
racing co

Diego Perez, Yago Saez *Member*

Abstract— The techniques and the technologies supporting Automatic Vehicle Guidance are important issues. Automobile manufacturers view automatic driving as a very interesting product with motivating key features which allow improvement of the car safety, reduction in emission or fuel consumption or optimization of driver comfort during long journeys. Car racing is an active research field where new advances in aerodynamics, consumption and engine power are critical each season. Our proposal is to research how evolutionary computation techniques can help in this field. For this work we have designed an automatic controller that learns rules with a genetic algorithm. This paper is a report of the results obtained by this controller during the car racing competition held in Hong Kong during the IEEE World Congress on Computational Intelligence (WCCI 2008).

on real prototypes, i.e., [6], [9] in the ARGO project, the Buick from the California PATH project or [10]. This is due to the fact that today more technological possibilities allow development of completely functional prototypes with lower costs.

In fact, there is a well-known annual competition organized by the Defense Advanced Research Projects Agency

INCREMENTAL EVOLUTION FOR SIMULATED CAR
RACING WITH DIFFERENTIAL EVOLUTION

YAP KHONG LIM

Implementation

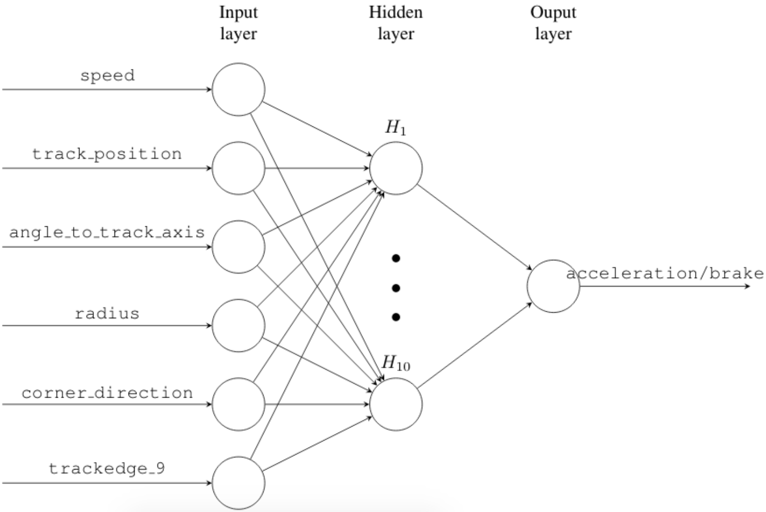
- Build Rule Based Controller
- Train Neural Network offline
  - LSTM
  - MLP
- Evolve weights
  - Differential evolution
- Code components
- Race

Outline

- Test multiple configurations (LSTM/MLP)
- Feed-forward MLP
  - input layer with 6 nodes
  - hidden layer with 10 nodes
  - 1 output variable
- Trained with DL4J on data from rule based controller
- Backpropagation

Design
Neural Network

Visualisation of the architecture:



Outline

Weights are evolved using differential evolution:

Representation	Real-valued vectors (81 weights)
Recombination	Uniform crossover (CR 0.7)
Mutation	Differential mutation (F=0.5)
Parent selection	Uniform random selection of the 3 necessary vectors
Survival selection	Deterministic elitist replacement (parent vs. child)

Outline

Design
Steering

Coded heuristic:

- Detect drifting of the car
  - Align steering with axis

Coded heuristic:

- Detect drifting of the car
  - Align steering with axis
- Check if car is near the edge of the track
  - Drive back to the middle

Coded heuristic:

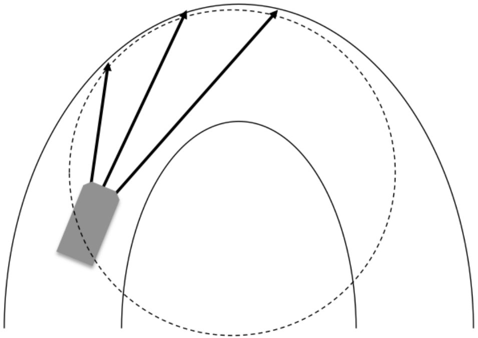
- Detect drifting of the car
  - Align steering with axis
- Check if car is near the edge of the track
  - Drive back to the middle
- Normal situation
  - Steer towards longest point of view

Coded heuristic:

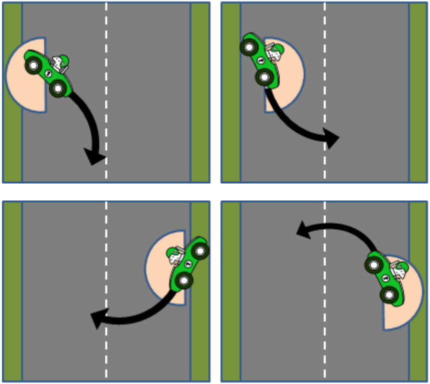
- Detect drifting of the car
  - Align steering with axis
- Check if car is near the edge of the track
  - Drive back to the middle
- Normal situation
  - Steer towards longest point of view
- Handle other situations
  - Correct to align with axis

Outline

Use the distances from the sensor along the axis and its immediate neighbours to estimate the radius of curvature

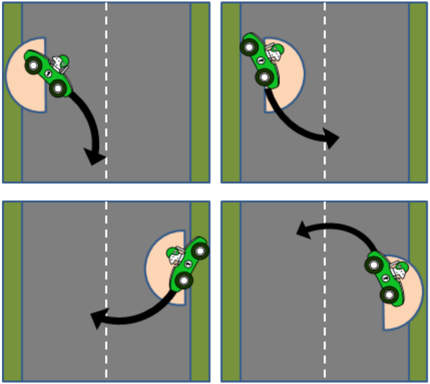


Outline



Simple turn in road principle

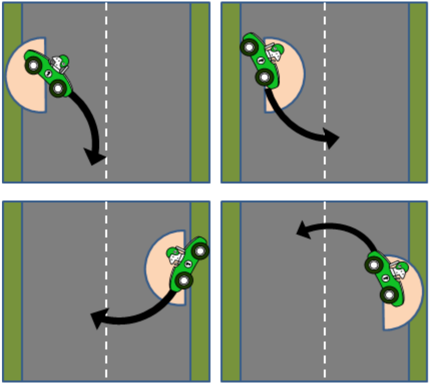
- Detect starting position of the car



Simple turn in road principle

- Detect starting position of the car
- Detect if car is on other side of road
  - New 'start' position

Design
Recovery



Simple turn in road principle

- Detect starting position of the car
- Detect if car is on other side of road
  - New 'start' position
- Detect if car is aligned with track axis
  - Full throttle

Race

- Controller selection
  - Cautious controller
  - Aggressive controller
  - Cautious first, aggressive next. Use best.

Race

- Controller selection
  - Cautious controller
  - Aggressive controller
  - Cautious first, aggressive next. Use best.
- Acceleration
  - If neural prediction function outputs $<$-0.2: full brake
  - If neural prediction function outputs $>$0.2: full acceleration
  - No brake/acceleration for output between -0.2 and 0.2 for smoother driving

Room for improvement

- Neural network for steering
- Co-evolution steering/acceleration
- Make use of track map
- More intelligent choice of controllers
- Use mutation instead of differential evolution
- Make use of opponent sensors
- Better recovery