Developments for ‘Racecar’ Project

Tim Stengelin, 001455200

These agents are deterministic, rule/learning-based controllers that use five forward-facing LIDAR range readings (-45°, -10°, 0°, +10°, +45°) and the current scalar velocity to choose two discrete actions each step: a steering action (left / right / straight) and a velocity action (accelerate / coast / brake). The agent's objectives are: (1) stay on the road, (2) maximize safe velocity.

Rule-based agents

Remark: If the font color is gray (e.g., Feature construction), this indicates that the part has not changed/remained the same from the previous version.

1. **Version: Simple Rule Agent**
   - Key building blocks
     1. Direction Control:
        Turn towards the side with more open space based on the 45° lidar.

$$dir\_control = \begin{cases} left & ; dist\_l45 - dist\_r45 > 0 \\ straight & ; dist\_l45 - dist\_r45 = 0 \\ right & ; dist\_l45 - dist\_r45 < 0 \end{cases}$$

     2. Speed Control:
        Adjust speed to keep constant velocity.

$$vel\_control = \begin{cases} accelerate & ; vel < 0.15 \\ coast & ; vel = 0.15 \\ break & ; vel > 0.15 \end{cases}$$

   - Results:

Average Score	Fast	Spiral	Small Circle	Large Circle	Oval
99.103	90.099	93.030	103.476	103.861	105.051

   - Observations:
     - The car reaches the finish on every track in all tests.
     - The car’s speed is not adapted to the individual track segments. In tight corners the car drives at a speed close to the maximum. On straights or in wide curves the car drives at a speed that is far below the maximum.
     - The car continuously tries to stay in the middle of the track. Only the 45° distance sensors are used to determine steering commands. In addition, it does not follow the ideal racing line, e.g. near the inside of a curve.
   - Plan:
     Creation of a completely new agent with rules that do not build on the Simple Rule Agent.
2. **Version: Enhanced Rule Agent**
   - Assumptions:
     For the calculations inside the agent several constants of the environment are assumed: the track is two units wide, the car can decelerate by up to 0.05 speed-units per step, and the car can rotate by up to 6° per step when cornering.
   - Key building blocks
     1. Feature construction:
        To distinguish between straights and turns, we form a weighted average of the -/+45° (40%) and -/+10° (60%) distance readings, then subtract the right-side average from the left-side average to produce a single direction value. If that value lies within ±0.05 the track is considered straight, a value greater than +0.05 indicates a left turn, and a value less than −0.05 indicates a right turn.

$$weighSum\_dist\_l = 0.6 \cdot dist\_l10 \; + \; 0.4 \cdot dist\_l45$$
$$weighSum\_dist\_r = 0.6 \cdot dist\_r10 \; + \; 0.4 \cdot dist\_r45$$

$$balance\_dist = weighSum\_dist\_l - weighSum\_dist\_r$$

$$dir\_track = \begin{cases} straight\_ahead & ; |balance\_dist| < 0.05 \\ left\_turn & ; balance\_dist > 0.05 \\ right\ turn & ; balance\_dist \; < \; 0.05 \end{cases}$$

2. Turn / curvature estimation (for turns only):
   For detected turns, the agent fits a circle through three points formed by the lidar readings and their bearing angles (using get_outer_circle). The function converts polar (r,θ) to Cartesian and computes the center of the circle (Ux, Uy), relative to the car position, and radius R. This gives an estimated curvature / radius of the road (ahead).

$$\begin{bmatrix} Ux \\ Uy \\ radius\_outer \end{bmatrix} = f(dist\_l45, dist\_l10, dist\_s, dist\_r10, dist\_r45)$$

3. Path-position reasoning (for turns only):
   For detected turns, the outer circle radius is rounded to the nearest 0.5 (round-half), and the inner radius is set to that rounded outer radius minus the track width. The car's distance to the center of the previously calculated circle is determined. With that, the relative across-road position is computed as a normalized value between 0 and 1.

$$radius\_inner = radius\_outer - TRACK\_WIDTH$$

$$actual\_radius\_path = \sqrt{(Ux)^2 \cdot (Uy)^2}$$

$$actual\_rel\_path\_pos = \frac{actual\_radius\_path - radius\_inner}{radius\_outer - radius\_inner}$$

4. Velocity setpoint (for straights and turns):
   For detected turns, the maximum allowed speed is limited between 0 and 1 and scaled according to the turn radius and the maximal change in direction per step. On straight sections the maximum allowed speed is always 1.

$$\text{desir\_max\_vel} = \begin{cases} 1 & ;\ \text{dir\_track} = \text{straight\_ahead} \\ \text{actual\_radius\_path}\cdot\text{MAX\_DELTA\_DIR\_PER\_STEP} & ;\ \text{dir\_track} = \text{left\_turn OR right\_turn} \end{cases}$$

5. Action selection rule for steering:
   On detected straight sections, steering is based on the weighted average of the distance readings.

$$dir\_control = \begin{cases} straight & ; |balance\_dist| < 0.05 \\ left & ; balance\_dist > 0.05 \\ right & ; balance\_dist < 0.05 \end{cases}$$

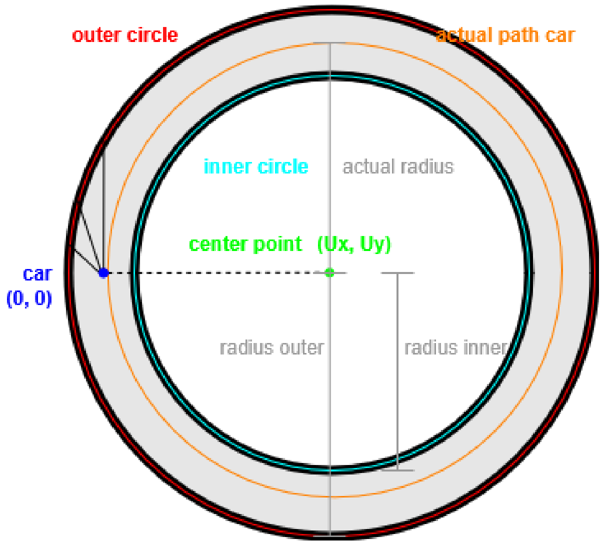
   For detected turns, if the vehicle's relative position shows a meaningful offset it actively steers into the curve (e.g., left for a left turn) and if the offset is negligible it continues straight through.

$$dir\_control = \begin{cases} left & ; (dir\_track = left\_turn) \wedge (actual\_rel\_path\_pos > 0.05) \\ straight & ; (dir\_track = left\_turn) \wedge (actual\_rel\_path\_pos \leq 0.05) \\ right & ; (dir\_track = right\_turn) \wedge (actual\_rel\_path\_pos > 0.05) \\ straight & ; (dir\_track = right\_turn) \wedge (actual\_rel\_path\_pos \leq 0.05) \end{cases}$$

6. Action selection rule for velocity:
   If the car's speed is within a small tolerance of the desired speed, coast; if it is above the desired maximum, brake; if it is below the desired maximum, accelerate.

$$vel\_control = \begin{cases} coast & ; |vel - desir\_max\_vel| < 0.02 \\ accelerate & ; vel - desir\_max\_vel > 0.02 \\ brake & ; vel - desir\_max\_vel < 0.02 \end{cases}$$

- Sketch:



- Results:

Average Score	Fast	Spiral	Small Circle	Large Circle	Oval
94.389	141.407	43.638	83.510	135.546	67.846

- Observations:
  One of the agent's major problems becomes apparent during testing: the transition from wide curves/straights to tight curves. The transition problem can be explained by the agent adjusting the radius-dependent target speed of the car too late. The calculated target speed is valid from the current time until a point in the near future.
- Plan:
  The agent should be able to make its (target) speed dependent not only on the current track segment but also on future track segments.

3. Version: Enhanced Rule Agent with emergency stopping

- Assumptions
- Key building blocks
  1. Feature construction
  2. Turn / curvature estimation (for turns only)
  3. Path-position reasoning (for turns only)
  4. Velocity setpoint (for straights and turns)
  5. Action selection rule for steering
  6. Action selection rule for velocity
  7. Emergency stopping:
     If the car is moving forwards and an obstacle straight ahead is closer than a velocity-dependent stopping distance, the agents set brake and steers away from the obstacle.

$$active = \begin{cases} 0 & ; dist\_s > \dfrac{(vel)^2}{2 \cdot MAX\_BRAKE\_PER\_STEP} \cdot (1 + 0.2) \\ 1 & ; dist\_s \leq \dfrac{(vel)^2}{2 \cdot MAX\_BRAKE\_PER\_STEP} \cdot (1 + 0.2) \end{cases}$$

$$vel\_control = brake$$

$$dir\_control = \begin{cases} left & ; balance\_dist > 0 \\ right & ; balance\_dist \leq 0 \end{cases}$$

- Results:

Average Score	Fast	Spiral	Small Circle	Large Circle	Oval
106.305	141.407	78.039	83.510	135.546	93.025

- Observations:
  - The problem with the transition from wide curves/straights into tight curves is solved. This is especially evident on Track 8 and results in an increase of the reward on that track (Score Track 8: 43.638 → 78.039).
  - The second major problem is that at the very beginning of the track the car often crashes into the wall if the first track element is a sharp turn. This can be explained by the calculation of the car's target speed for curves. The calculation assumes that the car's heading/orientation is at right angles to the line through the car's center and the center of the curve. This is not always true at the start, because the car is placed with a random heading/orientation.
- Plan:
  The agent should use different rules to determine speed/steering angle during the start phase.

4. Version (MyAgent.py): Enhanced Rule Agent with emergency stopping and startup behaviour

- Assumptions
- Key building blocks
  1. Feature construction
  2. Turn / curvature estimation (for turns only)
  3. Path-position reasoning (for turns only)
  4. Velocity setpoint (for straights and turns)
  5. Action selection rule for steering
  6. Action selection rule for velocity
  7. Emergency stopping
  8. Startup behaviour:
     For the first 10 steps the agent follows a startup routine: it accelerates on every fifth step and otherwise coasts, while steering is based on the weighted average of the distance readings.

$$active = \begin{cases} 0 & ;\ step \geq 10 \\ 1 & ;\ step < 10 \end{cases}$$

$$vel\_control = \begin{cases} accelerate & ;\ step \,\%\, 5 = 0 \\ coast & ;\ step \,\%\, 5 \neq 0 \end{cases}$$

$$dir\_control = \begin{cases} straight & ;\ |balance\_dist| < 0.05 \\ left & ;\ balance\_dist > 0.05 \\ right & ;\ balance\_dist < 0.05 \end{cases}$$

- Results:

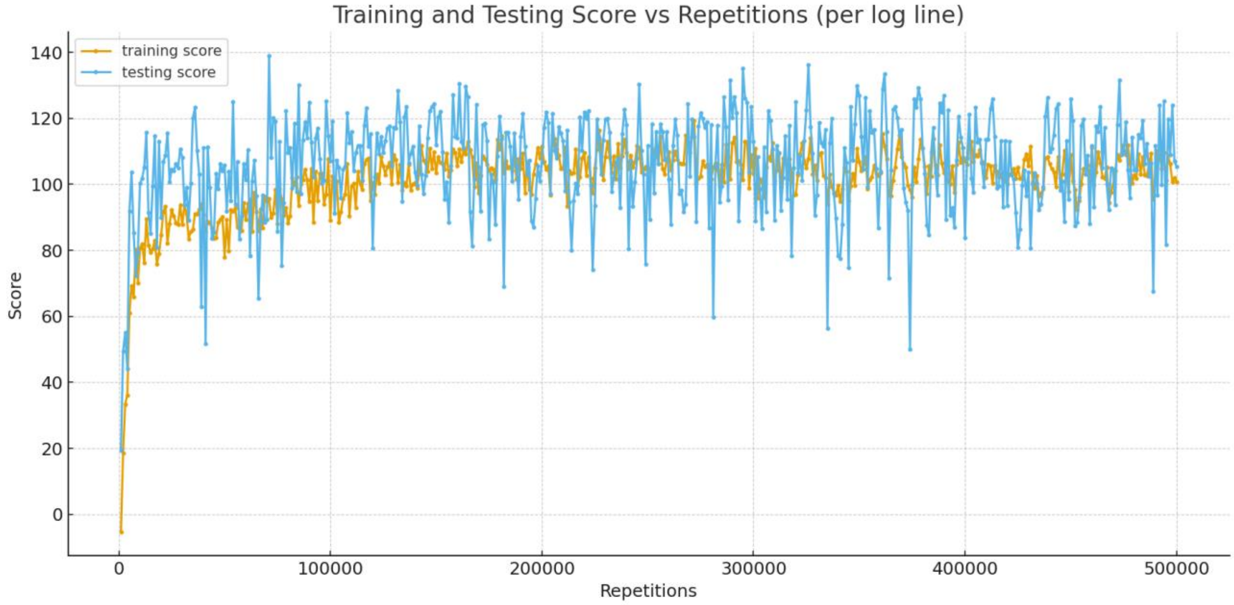
Average Score	Fast	Spiral	Small Circle	Large Circle	Oval
152.879	197.604	113.671	108.873	191.031	153.217

- Observations:
  - The problem with the start phase is solved. This is evident on all tracks (Average Score: 106.305 → 152.879).
  - In this version, 'emergency stopping' is also enabled/disabled depending on the timestep. This further increases the average score.
- Plan:
  A possible improvement is to refine/modify the algorithm for determining $dir\_track \in \{straight\_ahead, left\_turn, right\_turn\}$. This information is the basis for all further agent decisions and is not always reliable.

Learning-based Agent

5. Version: Q-learning Agent

- Method: Simple Q-learning with learning rate of 0.1.
- Results:



- Observations:
  - The training and testing scores converge during learning toward an average score of about 105.
  - In addition, both curves show large fluctuations, which indicates a learning rate that is too high.
- Plan:
  Use a more suitable (smaller) model; increase exploration; use a decaying learning rate. (For time reasons, the learning approach for the agent was not pursued further.)

Summary of Results

Ver-sion	Description	Rule (R) / Learning (L)	Average Score	Fast	Spiral	Small Circle	Large Circle	Oval
1	Simple Rule Agent	R	99.103	90.099	93.030	103.476	103.861	105.051
2	Enhanced Rule Agent	R	94.389	141.407	43.638	83.510	135.546	67.846
3	E. R. A. with emergency stopping	R	106.305	141.407	78.039	83.510	135.546	93.025
4	E. R. A with emergency stopping and startup behave.	R	152.879	197.604	113.671	108.873	191.031	153.217