**Abstract**

# Abstract

The following functional goals are set:

* Record 1 human within normal, but diverse traffic conditions and learn the behaviour in undisturbed scenarios (goal 1)
* Record various scenarios to identify comfort, nominal and dynamic lane change behaviour (not connected to specific human drivers) (goal 2)
* Enable dynamic parametrization of the algorithm to prove the resulting trajectory is adjusted. (goal 3)

# Issues (situation, motivation and tasks)

This activity aims to provide technical solutions for personalizing the vehicle behaviour in case of automated lane changes. The motivation is Toyota TSS4 PSN (Personalized Behaviour Patterns) feature request. The project is in acquisition phase, where the general engineering purpose is to find simple solutions for limited use cases. These solutions must act as proves of the concept.

## Scenario coverage

The following scenarios should be considered during the development:

* Highway
* Speed range: 75kph – 160kph
* 2 or 3 parallel lanes
* Traffic density: with no object around to mid-density (tbd)
* Lanes with good marking

The scenario definition will act as a basis for measurement data segmentation.

## Problem formulation

In the scenarios defined in point 1.1 the characteristic of the driver must be measured and identified. As the customer explicitely requested certain indicators to identify, we consider this first. The first measure of the lane change is the *time* needed to complete the lane change maneuver. Our aim is to provide a solution, which under the circumstances provided in point 1.1 is able to learn the time horizon preference of the given driver, and it is able to reproduce the lane change characteristic through VMC lateral control chain. For this, the following considerations must be made:

* What are the quantities (either ego vehicle or environment) on which the human’s selected time horizon depends on? These will be our predictor variables.
* Can we, and if yes, how can we influence the automated chain behaviour to reproduce the behaviour?
* How do we learn the time horizon (in terms of methodology, real time or offline…etc).
* How do we exclude outlying situations?
* How do we classify lane changes?

These questions will be answered in the following chapters.

## Development platform

The development platform is divided into two things:

* Data acquisition system (including measurement and segmentation)
* Algorithm development

The learning data is gathered from real world measurement using a test vehicle. The measurements are conducted in real traffic. A reference route is used, whose details can be found in appendix 1. The drivers are certified Bosch drivers (with CAT11 training available), with the conscience of the topic, and the goals of the measurement. The only deviation from normal driving is the need of pressing lane change approval button once the lane change is done according to the drivers.

The test vehicle is the VMC Platform vehicle of XC-DX/ESM5-Bp group. Its features are the following:

* Volskwagen Golf VII Variant
* Engine: 1.4 TSI with 7-geared DSG gearbox system
* Front radar: Gen4 Bosch; Front camera: MPC2.5 Bosch camera system
* Tires according to weather conditions, pressure checked

As an extra (non-vehicle built-in) measurement device a high accuracy GPS system is used. It is an ADMA Gen 3 device, including a built-in IMU and estimator for vehicle kinematic quantities. 2 pcs of Novatel GPS antennas are used, which are located at the top of the vehicle. Their position is calibrated in the ADMA. All quantities are transformed to the rear-axis midpoint (base link). Before all measurement starts the settling of the Kalman filter is checked. The GPS availability is measured. RTK-Fixed status is preferred and cut later in the segmentation phase.

Table : measured quantities

|  |  |  |
| --- | --- | --- |
| **Signal name** | **Description** | **Unit** |
| GPS\_status | GPS availability (1: not available, 2: normal GPS, 3: RTK course, 4: RTK fixed) | - |
| KF\_status | 0 if KF is not settled yet, 1 if it is settled | - |
| Left\_/Right\_Index | Indicator status operated by the driver | - |
| LaneChange\_Approved | Lane change manually approved by the driver | - |
| GPS\_time | Global GPS time | ms |
| Lat/LongPos\_abs | Global position values in the earth CS | degrees |
| theta\_abs | Orientation of the vehicle based on the GPS | degrees |
| yawRateGPS | Yaw rate of the vehicle fused from the GPS data and the IMU data | degree/sec |
| c01\_left/right | Ego lane edge distances | m |
| c02\_left/right | Neighbouring lane edge distance | m |
| c1 / c2 | Common orientation and curvature of the lanes | -/ 1/m |
| SteeringAngle | Road wheel angle measured value | rad |
| AccelerationX\_ESP | Longitudinal acceleration in ego frame | m/s^2 |
| AccelerationY\_ESP | Lateral acceleration in ego frame | m/s^2 |
| yawRateESP | Yawrate of the vehicle in ego frame | rad/s |
| VelocityX\_ESP | Longitudinal velocity in the ego frame | m/s |
| dxPP | Object distance in front | m |
| vxPP | Object velocity in front | m/s |
| axPP | Object acceleration in front | m/s^2 |

Table 1 contains the recorded measurement signals. There are usually more signals measured for later purpose, but those are filtered out when converting the raw data to mat files.

The development starts in a prototypical environment. The entire post-measurement data conditioning happens in a self-developed evaluation framework. This is called kdp\_hlb\_evalframework. The framework is divided into three main steps:

* Data conversion.
* Segmentation
* Evaluation.

The segmenting and evaluation scripts are written in MATLAB. The conversion happens in a Python framework. The entire MATLAB code is embedded into the Python framework. The user must open an executable UI, through which the measurement files, segmentation profiles and the evaluation profiles must be given.

In the data conversion step the raw data (usually mf4 format) is converted into mat files. The mat files only contain the configured channels.

The segmenting makes data quality checks, filtering and calculation of some derived quantities which are directly not measured. The classification of the measurement segments also takes place here. For instance lane changes are cut here, and they are also classified to groups.

The evaluation phase include indicator calculations (e.g. time horizon of the lane change), generate plots or runs the learning algorithm for the lane change model.

The outputs are the following:

* Generated data for model validation
* Optimization plots
* Performance plots (comparison between human drives, model and VMC model).

The VMC relevant components in open-loop configuration can be added to the evaluation which will provide opportunities of direct testing with the vehicle code.

## Segmentation

Segmentation stands for all calculations which condition the raw data in terms of quality and relevance, and also calculates extra information. It also classifies the data segments.

In the following different functions and their main information are detailed.

### New channel calculations (mainly the pose info)

The new channels are calculated for the segments before cutting. These are the followings:

(1)

(2)

(3)

These signals will be used as the global coordinates and orientation of the vehicle in the end.

### Localization

Localization means defining the global position of the vehicle (based on global coordinates), and filter out locations which are out of the predefined relevant positions of the map. There are constant lane sections which are defined as stable roads, and roundabouts, exit/entries are filtered out. Any measurement points lying outside of the relevant locations are neglected.

### Lane change detection and lane position (incl. correction)

The lane change detection happens in two steps. The first step includes a rough estimation of the lane change, where the main source of information is the driver’s input from the indicator (left or right) and an approval button, which is pressed at the end of the lane change maneuver. The first logic checks the rising edges of these signals. Once the indicator is set to any direction, the lane change maneuver begins, and once the approval button is set, the maneuver is considered finished. There are however some remarks:

* This way the lane change is usually longer than the actual maneuver (where the vehicle has significant lateral velocity to the lane edges).
* It must be ensured, that the approval button is pressed before the indicator is set to the other direction, otherwise the lane change is invalid.
* If no approval button is set after a time out period, the lane change is considered invalid, the vehicle is assumed to stay in its previous lane.

Good and bad examples can be found in Figure 1. Top-down the lane change is:

* Valid
* Invalid
* Invalid
* Invalid

The lane changes detected according to step 1 are indicated in newly added channel to the measurement. These channels separately indicates the left and right changes. When lane change maneuver is ongoing, these channels are set to value 1, otherwise to 0.

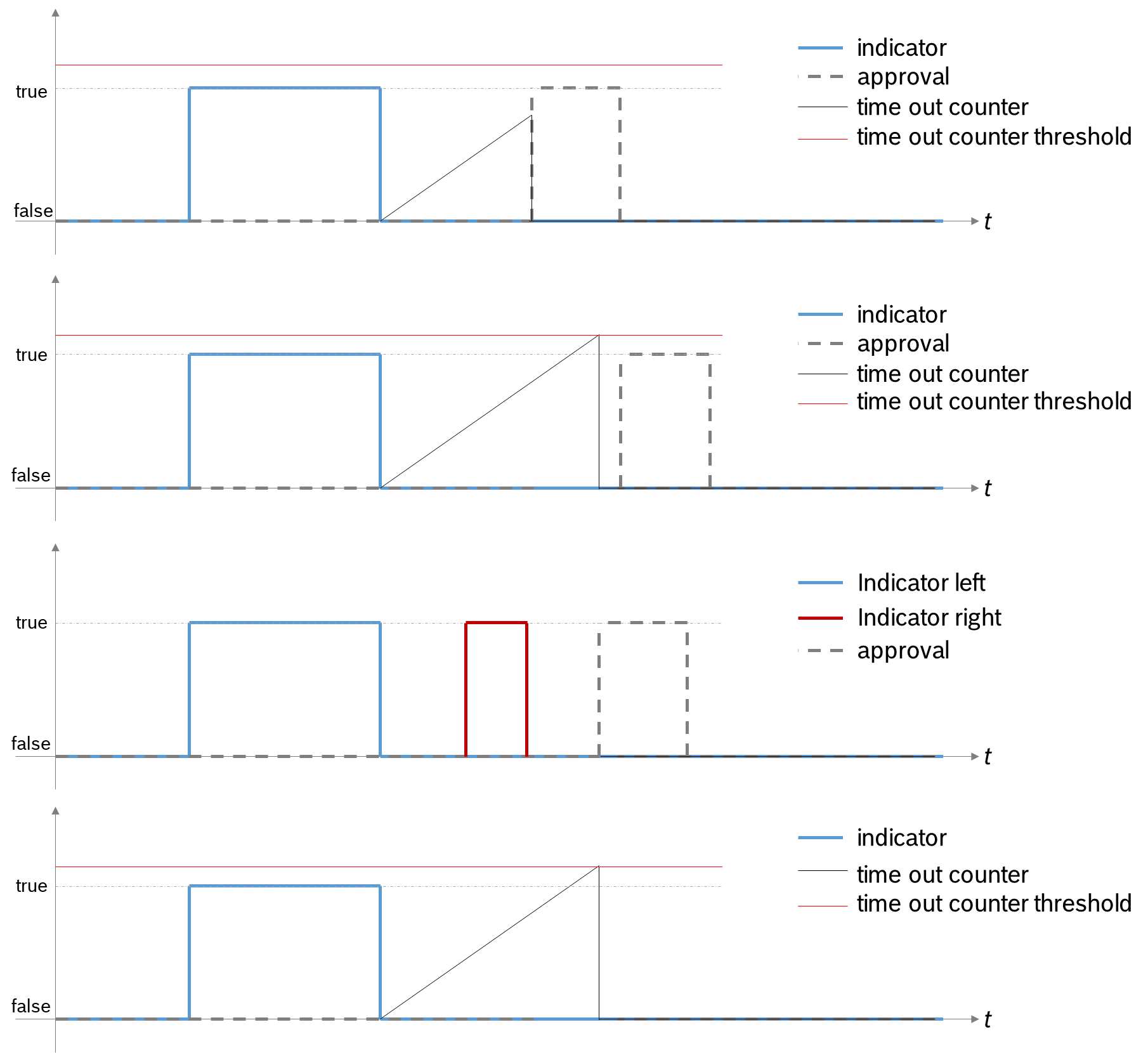


Figure 1: different scenarios for lane change detection

In step 2 of the lane change detection the previously detected lane changes are slightly modified. The goal is to have lane change start and end indeces according to the actual vehicle movement. The movement is conserved in the relative yawrate of the vehicle compared to the reference line (lane). There is a constant yawrare threshold and a debounce timer to start and finish the maneuver (once the relative yawrate decreases below the threshold and remains there until at least the debounce time, the lane change is considered to be finished). The threshold and the debounce time is calibratable. In these two steps a high accuracy lane change detection is available.

After the lane change is detected, the position information is calculated.

### Lane arrangement

Based on the lane position described in point 1.4.3 the lane edges are shifted. The original naming is kept, but the channel understanding changes. The camera provides the lane information always relative to the ego lane. As such, the understanding of the lane edges are the following:

* c01\_left / right: lane edges of the ego lane
* c02\_left: the left edge of the lane which is left to us (e.g. inner lane in the highway, oppsite lane on a 2x1 lane country road
* c02\_right: the right edge of the lane which is right to us (e.g. outer lane in the highway)

However, with this arrangement there are jumps in the lane edge signals when the lane change occurs (in case of a left lane change c02\_left becomes c01\_left, c01\_left becomes c01\_right..etc). This way during the evaluation the reference line can only be calculated with difficulties, also the Frenet coordinates may jump. Therefore based on the position, the lanes are re-arranged. When the vehicle crosses the given lane edge, the lane position steps (+1 or -1) and then the lane edges are arranged to have the following meaning:

* c01\_left: outer or middle lane on highway, ego lane on 2x1 country road, left edge
* c01\_right: outer or middle lane on highway, ego lane on 2x1 country road, right edge
* c02\_left: inner lane on highway, opposite lane on 2x1 country road, left edge
* c02\_right: outer-outer lane on highway, right edge

With this rearrangement the corridor edges will be continuous.

*Note: as there is a slight asynchronous behaviour of the lane edge changes in the camera, the position is recalculated for all lane edges.*

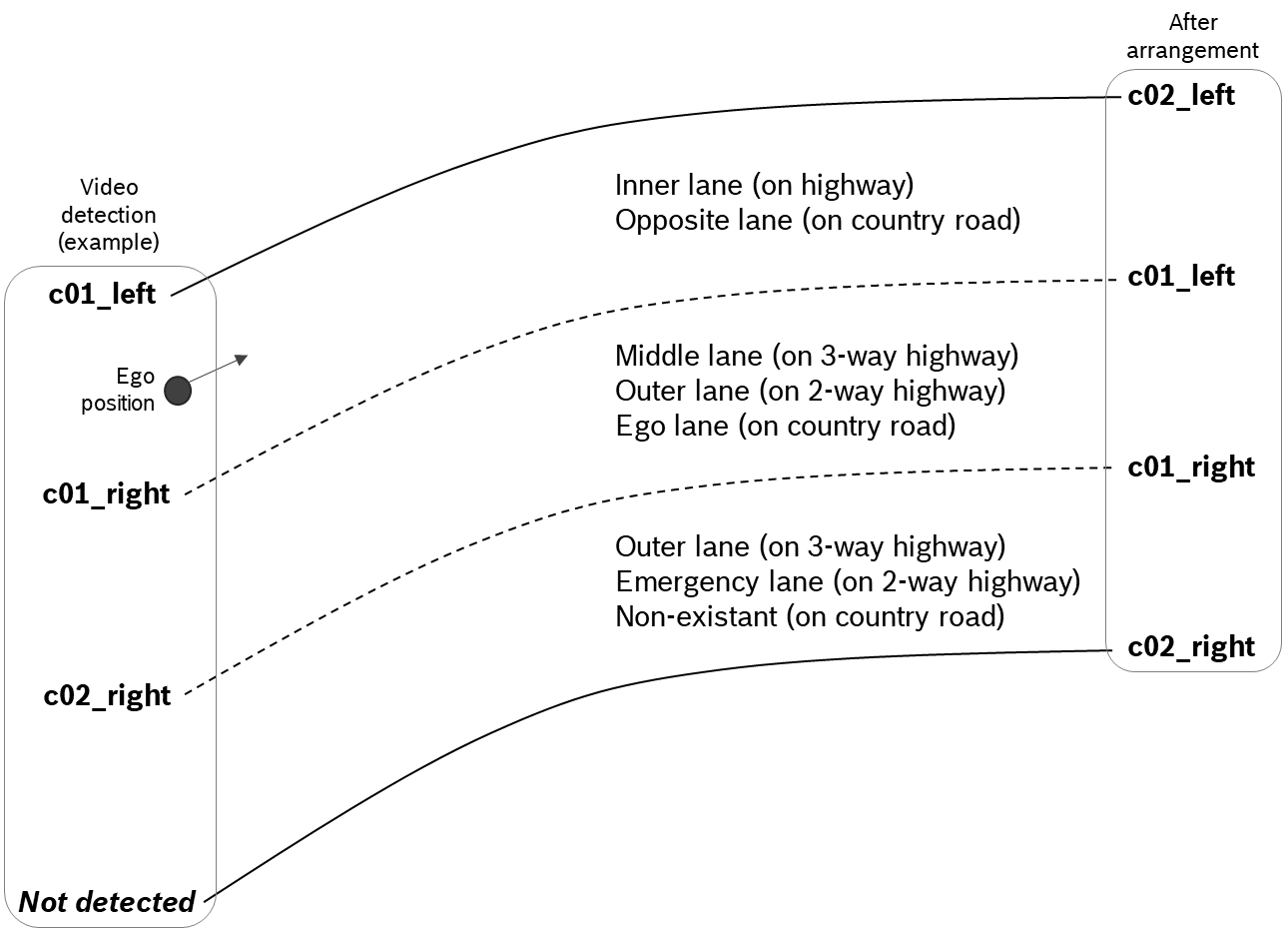
**

Figure 2: lane indexing for video (left) and after arrangement (right)

After the arrangement of the lanes, the lane change target reference line (as the middle line of the target lane) can be injected to any algorithms for trajectory planning.

### GPS status and redundant points

High precision GPS data is interferred by different objects and geographical coverity issues. In such cases, the status goes from RTK\_Fixed to course or quality even below. However, in many situations, the quality is restored after a couple of seconds. In such intervals, the Kalman filter is able to estimate the pose with good accuracy, therefore the measurement points are debounced for a calibratable time interval for GPS status lower than RTK\_Fixed.

Redundant points are filtered out, where the same measurement points in terms of position or time are detected. This can be due to:

* Too high measurement frequency
* Missing GPS detection (earlier position data is kept until new one is received).

### Lane change validity check

## Classification

After lane changes have been cut, their type / class is defined. For now, these are categorized as the following:

* Undisturbed scenarios
* Enforced scenarios

Both categories are kept in the uncut category to enable summarized evaluations. For goal 1 only the undisturbed scenarios are evaluated. ‘Undisturbed’ means the driver decided by its own intention when and how to perform the lane change. In ‘enforced’ scenarios the lane change is done differently to undisturbed ones, as the traffic situation enforces the driver to act differently. The following scenarios are distinguished as enforced classes:

* Delayed left lane change (waiting for someone to pass)
* Accelerated left lane change (due to back traffic or generic acceleration situation)
* Enforced right lane change (due to back traffic, cut in to the outer lane)

In general, longitudinal acceleration profile is chosen to classify the scenario itself. The approach phase (before indicator is set, and lane change starts) and the maneuver (until the end of lane change) is analysed. If longitudinal acceleration exceeds a precalibrated constant threshold, and the delta velocity during the maneuver exceed a constant threshold, the scenario is classified as enforced, otherwise undisturbed.

In the future, the post phase of right lane changes may be examined as well, as in enforced situations deceleration usually occurs in the later phase of the lane change maneuver.

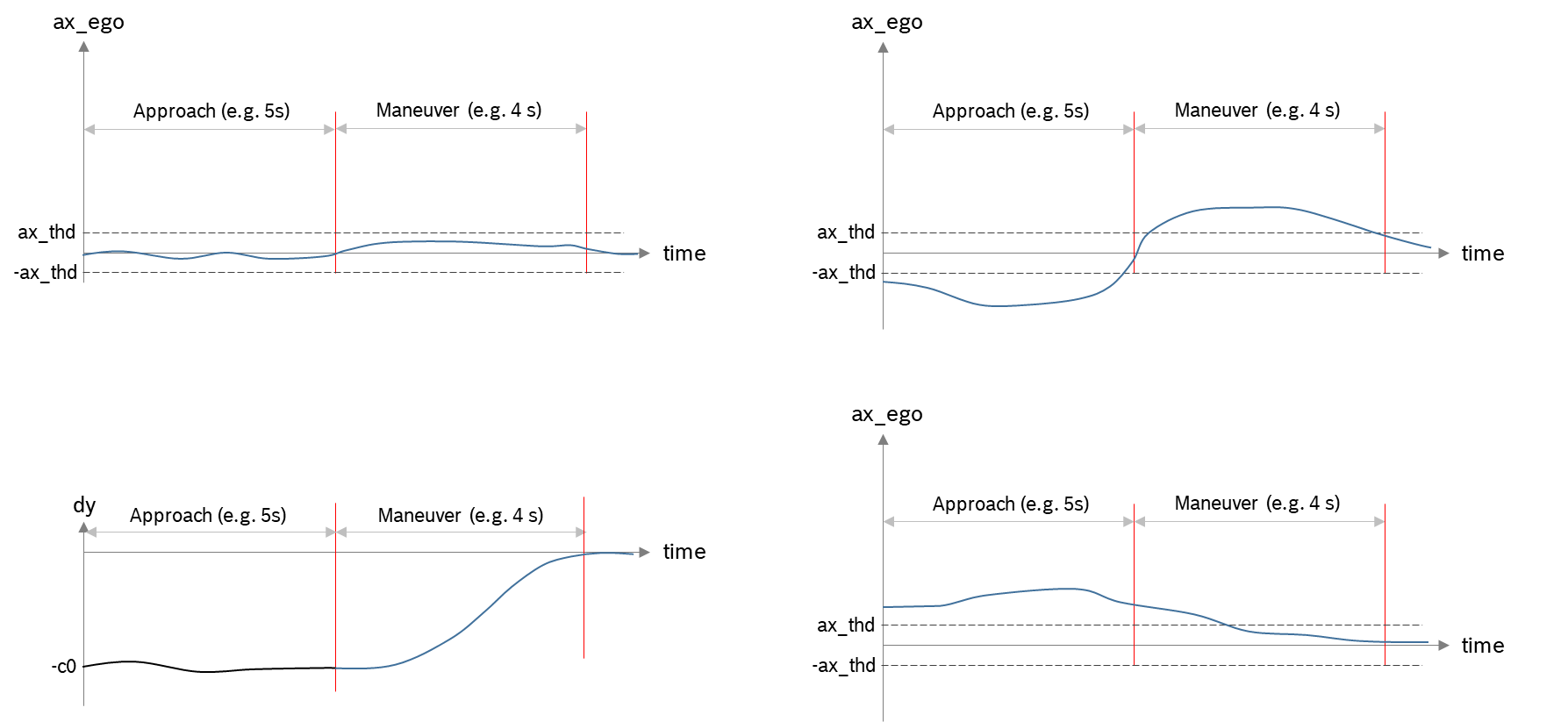


Figure 3: classification based on longitudinal acceleration; (a): undisturbed acceleration profile; (b) delayed left lane change; (c) trajectory; (d) accelerated left lane change

For goal 2 different aspects of the lane change may be examined. The initial guesses of these aspects are derived from VMC trajectory planner cost function:

* Lateral jerk squared
* Lateral acceleration squared
* Time horizon
* Overshoot over reference line.

The classification can occur offline for each scenario, then used in the learning phase accordingly. Once the activiation will occur in the future (of ALC functionality), situation dependency is examined, and the right behaviour is chosen. In the first phase of the development only undisturbed scenarios will be considered to learn.

## Resimulation

Resimulation can occur two ways:

* Open loop
* Closed loop

Open loop means that the given part of the motion software is driven by a measurement. In this case, the measurement serves as the input to the software, the ouput is directly evaluated. The corridor preparation (refLat) and trajectory planner (trpLat) are built to a simulation library (PSL) and integrated into a Simulink environment. A block diagram for this, including the optimization components are displayed in Figure 4. This loop is embedded into the kdp\_hlb\_evalframework. The model is driven by the lane change evaluation scripts, with the input set coming from the measurement.

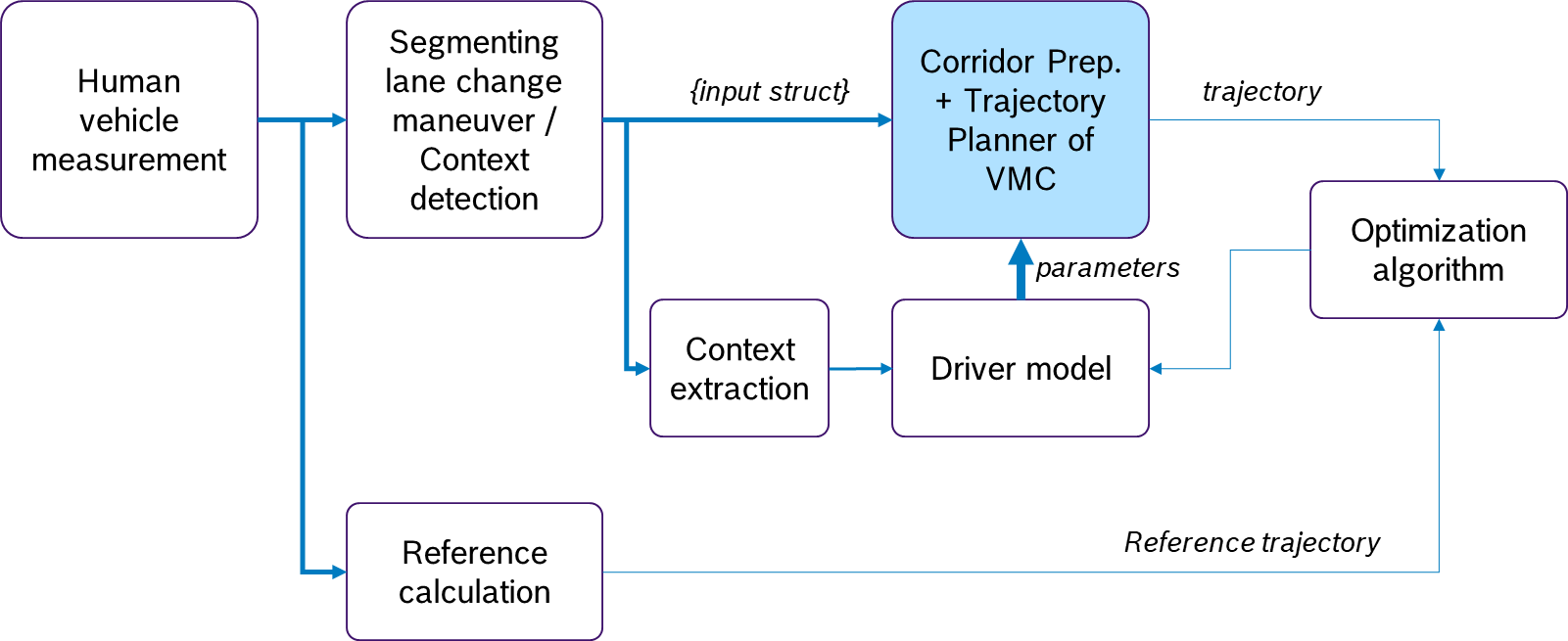


Figure 4: block diagram of open loop optimization

The output of the open loop model is the trajectory that is planned by the planner. The optimization runs after the model output is processed. The cost function is calculated within separate evaluation functions. The parameters are changed in the corresponding dcm parameter file, and the simulation is run again. Usually, the entire simulation – cost calculation function is embedded into the optimization function of MATLAB called fmincon (or fminsearch). This is used for offline learning. Online optimization functions are yet to be implemented.

In the closed loop situation, the recorded environmental conditions are automatically transformed to carmaker road and traffic definitions and the complete planner and control function can be replayed. The implementation of this is yet to be finalized.

## Lane change model

According to Figure 4 the model connects input variables (predictor variables) to selected influencing parameters (output variables). The model must work in online context, therefore signals which are available in real time must be only used for deciding the right parameter value. The predictor variables are initially chosen as the followings:

* Ego velocity
* Initial object distance (left lane change only)
* Initial object relative velocity (left lane change only).

This is valid for goal 1. Based on the data exploration phase, these may be changed.

## Evaluation plots and outcomes

## Measurements

|  |  |
| --- | --- |
| Goals | Basic description |
| **Goal 1** | Speed range: 70 – 150 kph. M0 / M4 / M6 / M7 🡪 highway and fastlanes  Undisturbed scenarios only  Various dxPP-s! 🡪 PERSONALIZATION |
| **Goal 2** | One generic driver drives only comfortable, dynamic and nominal lane changes, with various velocities and dxPP-s. 🡪 STYLE |
| **Goal 3** | For this the VMC needs to be used, until the vehicle is available, simulations can be done only 🡪 for this the self-created open-loop and the closed loop can be used. With various k\_t parameters will be checked. |

# Results

## Data exploration – personalization

Table 2: time horizon, personalization, undisturbed scenarios

|  |  |
| --- | --- |
|  |  |
|  |  |

Table 3: maximum lateral jerk in the 1/3rd of the lane change

|  |  |
| --- | --- |
|  |  |
|  |  |

## Data exploration – Style

Table 4: maximum jerk in the 1/3rd of the lane change - dynamic

|  |  |
| --- | --- |
|  |  |
|  |  |

Table 5: maximum jerk in the 1/3rd of the lane change - comfort

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Name of quantity** | **Dynamic** | **Comfort** |
| Maximum jerk in 1/3 | 4.665 / 1.187 | 2.417 / 0.96 |
| Maximum jerk | 6.384 / 2.330 | 2.976 / 1.458 |
| Time horizon | 5.447 / 0.445 | 6.116 / 1.102 |
| Maximum acceleration | 1.654 / 0.301 | 0.696 / 0.232 |
| Integrated square jerk | 20.970 / 2.879 | 4.561 / 2.864 |
| Integrated square velocity | 5.289 / 2.188 | 1.975 / 1.225 |

# Conclusions and Consequences

The following statements can be done based on the results:

* Personalization based on current measurements is not a good option, as all drivers drive the same way. Extending the type of drivers would make sense.
* Dynamic and Comfort lane changes can be identified through different indicators, most promising is time horizon and the initial jerk of the trajectory.
* Correlation is not yet seen based on the predictor variables defined (dxPP, vxPP and vxEgo). TTC as a possible good prectior variable should be added.

Next steps:

* With resimulation of the measurement data different parameter sets will be produced which will give comfort and dynamic lane changes.
* Define the key quantities which characterize the agressiveness of a lane change
* Classify lane changes automatically based on previously defined key quantities

Options:

* Predefine a few parameter sets which all correspond to a given lane change style (e.g. dynamic OR comfort)
* Have the driver option to continuously set some ‘agressiveness meter’, which – in the background – is connected to VMC parameters
* Learning: it is seen at the moment how learning is possible. Nominal driver profile may be learnt, but the assumption is this will be close to the precalibrated profile.
* Connecting predefined profile (e.g. aggressive) to current human state can be an option, but this is the scope of the knowledge model

Please enter text

|  |  |  |  |
| --- | --- | --- | --- |
| **Document approval** | | | |
|  | **Org.-Unit** | **Name** | **Signature** |
| **Author(s):** |  |  | electronic release (FEBER) |
| **Responsible person:** |  |  | electronic release (FEBER) |
| **Reviewer:** |  |  | electronic release (FEBER) |
| **Releasing person:** |  |  | electronic release (FEBER) |
|  |  |  | electronic release (FEBER) |
|  |  |  | electronic release (FEBER) |
|  | | | |
|  |  | **External release of cover sheet:** | No |
| **Enclosure(s):** | Yes | **External release of enclosure(s):** | No |

|  |  |
| --- | --- |
| **Initial mailing list** | |
| **Recipients:** |  |
| **Cc:** |  |

|  |
| --- |
| **Key Words** |
|  |

# Appendices

## Appendix 1: reference route

|  |  |
| --- | --- |
| **Enclosures** | |
| **Enclosure number** | **Title** |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Underlying documents please link documents** | | | |
| **Document number** | **Title** | **Date** | **Responsible person** |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **This report invalidates** | | | |
| **Document number** | **Title** | **Date** | **Responsible person** |
|  |  |  |  |