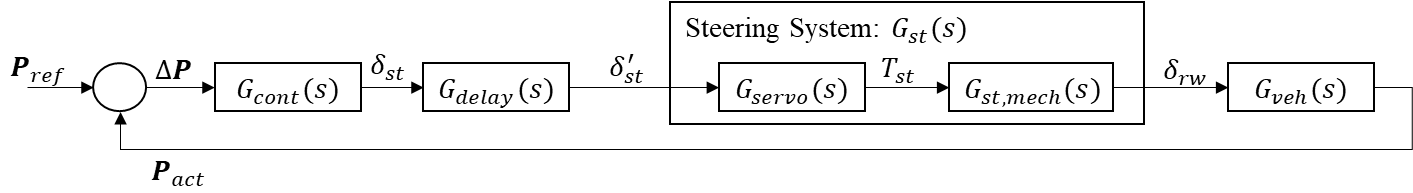
PID Lateral Control

1. Calculating steering dynamics

Steering System is modelled as a second order linear system (PT2), described by the following transfer function:

(1)



Here and are the reference and actual vehicle poses, provided by the planning and the localization. The error is controlled by the vehicle lateral controller described by the transfer function . Then, the steering angle is sent to the low level steering angle controller. Between, there is a delay block which models the transfer delay due to communication. Then, the low level control calculates the intervention torque . Please be noted, that includes the feedback of the steering angle. This is usually some linear controller, which is hidden from the vehicle controller layer. Then, steering torque is applied on the steering column (or rack, depending on the vehicle type). The steering column to road-wheel connection point system is modelled by transfer function. In this sense: . Even though, this system is usually higher ordered than 2, e.g., 4 (if both servo and mechanical transfer functions are second ordered), usually: , which means two time constant is more dominant than the rest. Thus, is simplified to a second order transfer function neglecting . is the transfer function of the vehicle reacting on the road wheel angle and outputting its new pose accordingly. is usually LPV, or NLPV system. Then, loop closes on the vehicle control level.

Our aim is, to identify the parameters of , namely gain, time constant and damping. The gain is considered to be , as the input is always normalized between 0 and 1. The gain of the real system is simply handled as a constant factor outside of the steering system. Therefore, only two parameters shall be calculated. For this, measurements with step excitation have been recorded. That has been done with multiple speeds and amplitudes.

Results are shown in the below figure. It is seen, that – even though parameters have variance – the mean values can be calculated for each speed. Damping slightly increases with speed, while the time constant is closely constant up until 13 m/s then starts a slow increase too. This is per the expectations, as at higher speeds the servo controller is usually damps the system more to prevent strong interventions.



Proposals based on results:

* calculate mean parameters, or
* fit a linear function on points and use value accordingly, or
* use look-up table with means for the various speed values, use linear interpolation between speed values and keep parameters constant outside unobserved range.

1. Inverse dynamics of the steering

(2)

Continuous – Discrete: Backward Euler

(3)

Then, the discrete transfer function is:

(4)

So, knowing what steering angle shall be on the output, the input is modified with the inverse of the transfer function:

(5)

For simplicity: , then:

(6)

Equation is non-causal, therefore it is slightly modified for practical usage:

(7)

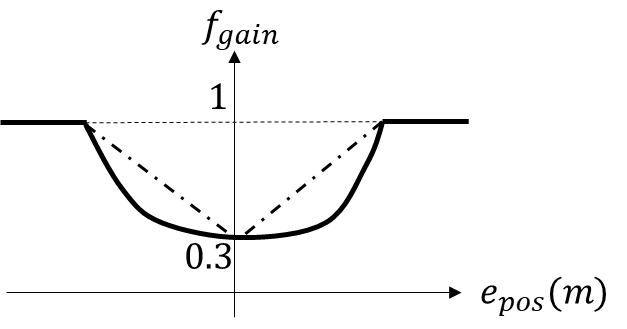
Where is the target steering angle provided by the vehicle control, are measured and stored values of the actual steering angle in the previous two cycles.

Parameters:

* time constant of , defined by identification,
* : damping of , defined by identification,
* : discrete time step of the controller.

1. Error dependent gains

The control gains can be scaled depending on the error. This way the feedback part gets more and more aggressive when error increases while have lower effects when the vehicle is close to the target path. This shape can be e.g., U-shape or V-shape, see in ***Fig. 1***. The maximum scale can be 1, meaining that the constant gain is fed through the system. This is reached at a maximum error, e.g., 1 *m,* below which the gain is decreased to a minimum value, e.g., 0.3 which is effective at zero error.



**Fig. 1:** Error dependent gains, U-shape or V-shape compensation

The controller error equations are modified accordingly:

(8)

Where , and are function of the scaling factor, shaped as given in Fig. 1.

1. Orientation error in control

Based on the tests it is seen, that position control itself is not sufficient to control the vehicle smoothly. Therefore, (8) is extended by the control of the vehicle orientation to the lane:

(9)

Where

(10)

As the target path is given in the form of a three-ordered polynomial, the orientation can be calculated simply from the coefficient:

(10)

Where is the look ahead distance.

1. Look ahead error calculation

In the discussion so far, the look ahead errors were calculated simply by evaluating the path equation in the ego coordinate frame. Instead, the vehicle path – considering the actual velocity and steering angle – can be estimated. The more accurate the estimation the better the feedback error calculation. Illustration is shown in ***Fig. 2***. The vehicle path is estimated:

(11)

(12)

(13)

Where is the yawrate of the vehicle at the control time , and can be estimated from the road-wheel angle:

(14)

(15)

Where is the actual vehicle path curvature, is the longitudinal axle distance. Then, we have a displacement vector estimated:

(16)

From this, with a simple 2D transform (global to ego), the look ahead point can be calculated, by knowing :

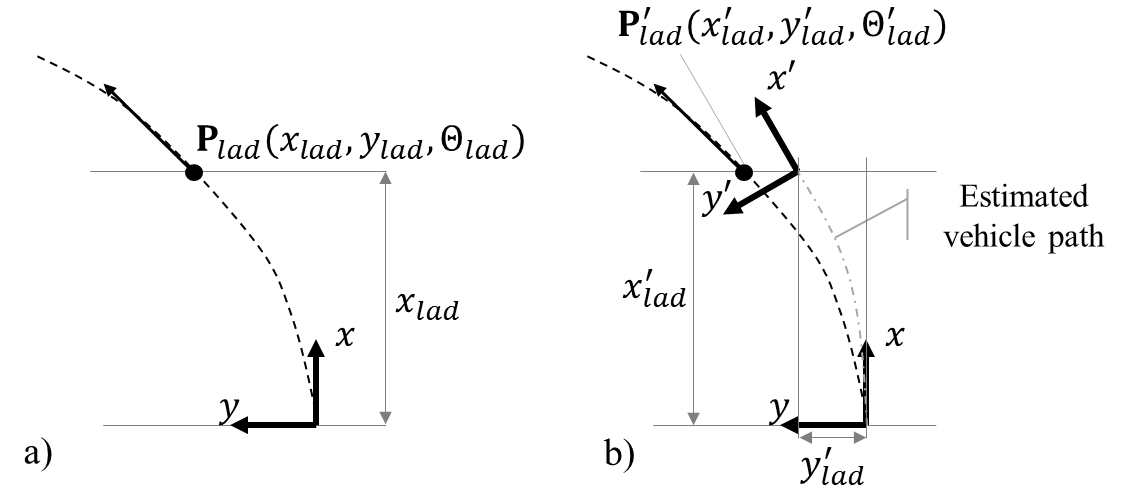
(17)

Where

(18)

And from here:

(19)



**Fig. 2**. Calculation of the look ahead error - a) original solution, b) using the estimated path of the vehicle

Then:

(20)

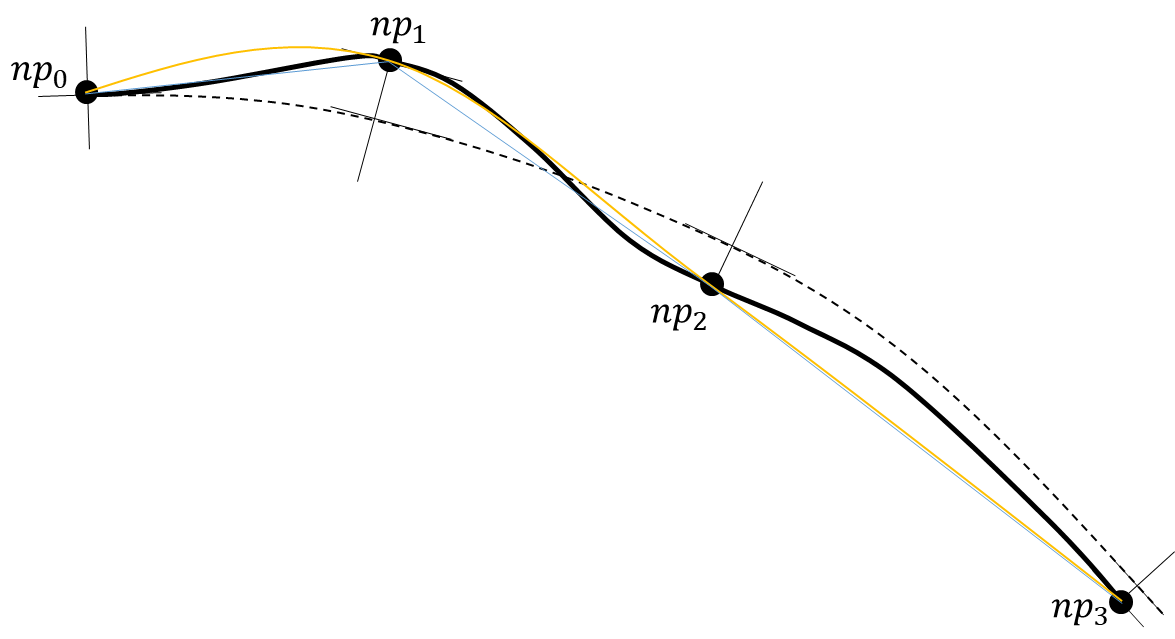
(21)

Local Path Planner

1. Orientation in the node point

In ***Fig. 3*** an example of the curve fitting experienced on the test drives is shown (bolded black). The problem is, that in the current configuration in the node points the curve segments are fitted to have the same orientation as the center line (dashed). This causes distortion in the fitted path. In order to overcome this issue, it is advised to change the curve fitting approach. Two proposals are made:

* fit third-order splines on the node points, which are continuous in terms of orientation and reflect the orange curve in the figure,
* or achieve similar effects by still fitting separate curves (polynomials) on the node points, by adding right boundary conditions for the orientation. This can be done in different ways, e.g., by calculating the orientation of the blue lines (lines connecting two neighbouring node points) and using their orientation for the fitting.



**Fig. 3**. Different curve fitting on the node points.