**Preprints of the 21st IFAC World Congress (Virtual)**
**Berlin, Germany, July 12-17, 2020**

# Experimental Verification of a Control System for Autonomous Navigation ⋆

**Benedek Szűcs** [**] **Ádám Kisari** [*,**] **Péter Kőrös** [*] **Dániel Pup** [*]
**Gábor Rödönyi** [*,**] **Alexandros Soumelidis** [*,**] **József Bokor** [**]

[*] *Széchenyi István University, Research Center of Vehicle Industry (SZE-JKK) H-9026 Egyetem tér 1. Győr, Hungary.*
[**] *Systems and Control Laboratory, Institute for Computer Science and Control (e-mail: soumelidis@sztaki.hu).*

**Abstract:** A flexible architecture is developed with the purpose of supporting education and research on the field of autonomous vehicles. A pure electric vehicle is equipped with on-board computational units, sensors and actuator interfaces. This paper presents the current status of the control system and its validation by means of navigation experiments. With the cascade control architecture, problems of actuator dead-zone, sensor offset errors, path tracking and redesign for obstacle avoidance are addressed.

*Keywords:* Autonomous Vehicles; Path Following; Obstacle Avoidance; Artificial Potential Fields

## 1. INTRODUCTION

Research on autonomous driving is motivated by the challenges of the worldwide overloaded traffic networks. Cooperating autonomous vehicles are significant components in a well organised future multi-modal traffic infrastructure. The technology promises improvement in car safety and efficiency with respect to travel time, energy consumption and reduced emission. To support research and education on this multidisciplinary field, a project is initiated by the Institute for Computer Science and Control (SZTAKI) and the Research Center of Vehicle Industry (JKK) with the purpose of developing an open platform for autonomous and cooperative driving.

In this paper the current state of control system is presented. The main objective of the specification of control architecture is flexibility that allows independent development of the different functions in a plug-and-play manner. To this end, the control problem of autonomous navigation is decomposed to the following main tasks with well defined interfaces.

(1) At the lowest level of control hierarchy, there are reference tracking controllers that manage the actuators: steering angle or yaw rate; and speed or acceleration reference tracking
(2) Path or trajectory tracking control. Given the path in the form of splines or a sequence of points, and given a corresponding speed/acceleration profile, the controller defines the reference signal based on knowledge of the closed-loop model of the low level control system.
(3) Path and trajectory planning. Furnished with the information about the surrounding environment and given the target in a relatively small time and spatial horizon, this level of control plans or modifies the reference path and trajectory to satisfy constraints and optimality criteria.
(4) Behaviour planning is responsible for analysing traffic situations and making decisions by setting short range targets.
(5) Keeping the global goals in mind, route planning selects the best routes.

Current state of the project covers some initial solutions regarding points 1)-3). The paper is organised as follows. A steering angle reference tracking controller and a path tracking controller are presented in Sections 2 and 3, respectively. Sensor offset estimation is discussed in Section 3.6. The representation of the path with quintic $G^2$-splines and the computation of the closest path point to a given point are detailed in Section 4. Path planning based on artificial potential fields are presented in Section 5. The experimental validation of the control system on the autonomous vehicle is presented in Section 6.

## 2. STEERING ANGLE REFERENCE TRACKING

The steering angle reference tracking controller is implemented on the NI cRIO9039 device. It receives the reference signal $\delta_{ref}$ through network $CAN_1$ from the path tracking controller and calculates a virtual driver's torque signal as command to the steering servo actuator. The steering servo actuator has a significant dead-zone effect

⋆ The research presented in this paper was carried out as part of the "Dynamics and Control of Autonomous Vehicles meeting the Synergy Demands of Automated Transport Systems (EFOP-3.6.2-16-2017-00016)" project in the framework of the New Széchenyi Plan. The completion of this project is funded by the European Union and co-financed by the European Social Fund.
This work was also supported by the National Research Development and Innovation Fund through the project "SEPPAC: Safety and Economic Platform for Partially Automated Commercial vehicles" (VKSZ 14-1-2015-0125).

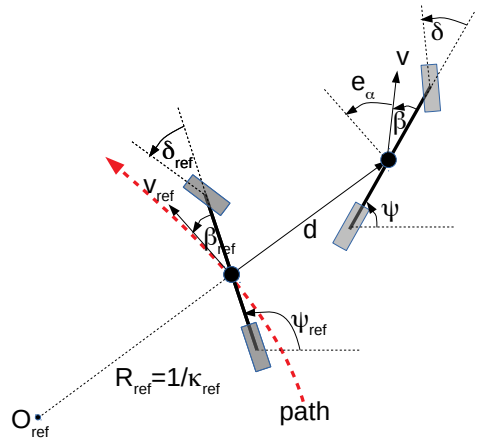


Fig. 1. Path-tracking problem formulation

on its input. The controller is a proportional-integral-derivative (PID) series compensator with integrator antiwindup, Goodwin et al. (2000), and is described by the following equations

$$u(k) = \text{sat}_{\underline{u}}^{\bar{u}}(u_P(k) + u_I(k) + u_D(k)) \quad (1)$$
$$u_P(k) = K_P e(k) \quad (2)$$
$$u_D(k) = \frac{K_D}{c_{dz}}(\delta(k-1) - \delta(k)) \quad (3)$$
$$u_I(k) = c_{dz} K_I (e(k) + e(k-1)) + x_I(k) \quad (4)$$
$$(5)$$

where $e(k) = \delta_{ref}(k) - \delta(k)$ denotes the tracking error, $\text{sat}_{\underline{u}}^{\bar{u}}(.)$ denotes the saturation function with lower and upper limits $\underline{u}$ and $\bar{u}$, respectively, $K_P, K_D, K_I$ are constant parameters, and $x_I$ is a state variable defined by

$$x_I(k+1) = \begin{cases} \bar{u} - u_P(k), & \text{if } x_I(k) + u_P(k) > \bar{u} \\ \underline{u} - u_P(k), & \text{if } x_I(k) + u_P(k) < \underline{u} \\ c_{dz} K_I (e(k) + e(k-1)) + x_I(k), & \text{otherwise} \end{cases} \quad (6)$$

$c_{dz}$ is a scaling parameter whose nominal value equals to 1, and is set to 1200 when $u(k)$ is in the assumed dead-zone. This scaling accelerates the controller in order to quickly leave the dead-zone, thus the effects of this input nonlinearity is drastically reduced.

## 3. PATH TRACKING CONTROLLER

At the next level of control hierarchy, the path tracking controller receives the reference path in the form of quintic $G^2$ splines and the GPS position and orientation of the vehicle, and calculates the steering angle (reference) required to accurately follow the path. A possible control approach can be that a virtual reference vehicle is defined that accurately tracks the path, and the deviation of the state variables of the true vehicle and this reference vehicle is minimized. The tracking problem is illustrated in Fig. 1, where v, $\beta, \delta$ and $\psi$ denote the speed vector, side-slip angle, steering angle and the orientation of the vehicle, respectively, and for the virtual reference vehicle $\text{v}_{ref}, \beta_{ref}, \delta_{ref}$ and $\psi_{ref}$ denote the target quantities calculated from the path based on the kinematic model of the vehicle.

### 3.1 Problem Formulation

Let a segment of the path curve be represented by a quintic vector valued polynomial $F(\theta) = \begin{bmatrix} x(\theta) \\ y(\theta) \end{bmatrix}$, $F : \mathbb{R} \mapsto \mathbb{R}^2$, with virtual parameter $\theta$. Let $d_F : \mathbb{R}^2 \mapsto \mathbb{R}$ denote the signed distance of a point to curve $F$, where the sign is defined to be positive on the right side of the path, and negative on the left. Formally, the goal of the path tracking problem is that the vehicle's center of gravity (CG) point $z(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$ be on the reference path at all discrete time $k$, i.e., $d_F(z(k)) = 0 \; \forall k$.

### 3.2 Controller Structure

According to the kinematic single-track model of motion, Rajamani (2011), the steering angle $\delta_{ref}(k)$ is related with the momentary curvature $\kappa_{ref}(k) = 1/R_{ref}(k)$ of the trajectory by

$$\delta_{ref}(k) = \arctan\left(\frac{\ell \kappa_{ref}(k)}{\sqrt{1 - \ell_r^2 \kappa_{ref}(k)^2}}\right) \approx \ell \kappa_{ref}(k), \quad (7)$$

where $\ell$ denotes the wheelbase. For relatively small angles and curvature, the linear approximation $\delta_{ref}(k) = \ell \kappa_{ref}(k)$ is sufficient. The side-slip angle

$$\beta_{ref}(k) = \arctan\left(\frac{\ell_r \kappa_{ref}(k)}{\sqrt{1 - \ell_r^2 \kappa_{ref}(k)^2}}\right) \approx \ell_r \kappa_{ref}(k). \quad (8)$$

together with the vehicle orientation $\psi_{ref}$ determine the direction of motion, i.e., the unit vector $\frac{\text{v}_{ref}}{\|\text{v}_{ref}\|}$. In (8) $\ell_r$ denotes the distance between CG and the rear axle.

If the vehicle's CG was on the reference path (i.e., $d(k) \triangleq d_F(z(k)) = 0$) with speed vector being tangential to the path, and the low level steering angle reference tracking was perfect, then a steering angle as defined in (7) would keep the vehicle on the $G^2$ continuous path, Piazzi et al. (2002); For this reason it is sensible to apply a feed-forward control signal term

$$\delta_{ref,ff}(k) = k_{ff} \kappa_{ref}(k), \quad k_{ff} = \ell. \quad (9)$$

Additional output-feedback terms should address the effects of uncertainties and nonzero initial conditions, and the goal is that the distance of the vehicle's CG point from the path $d$ and the angular difference between the tangent of the path and the speed vector of the vehicle are asymptotically driven to zero. The complete path tracking controller has the following form

$$\delta_{ref}(k) = k_{ff} \kappa_{ref}(k) + k_\alpha e_\alpha(k) + k_d d(k) \quad (10)$$

where $k_\alpha$ and $k_d$ are design parameters, and $e_\alpha$ denotes the heading error,

$$e_\alpha(k) \triangleq \psi_{ref}(k) + \beta_{ref}(k) - (\psi(k) + \beta(k)) \quad (11)$$
$$= e_\psi(k) - \beta(k) \quad (12)$$

where $\beta(k) = \ell_r \kappa(k)$ and $e_\psi$ defined by

$$e_\psi(k) \triangleq \psi_{ref}(k) + \beta_{ref}(k) - \psi(k) \quad (13)$$

can be computed directly from measured vehicle orientation $\psi(k)$ and the orientation of the path $\psi_{ref}(k) + \beta_{ref}(k)$.

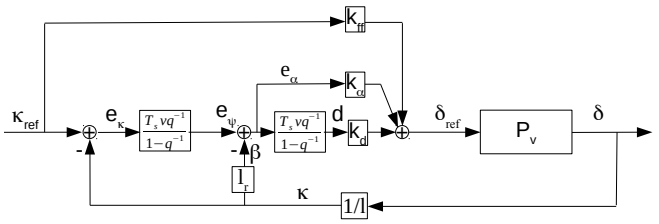


Fig. 2. Feedforward-feedback path tracking control scheme

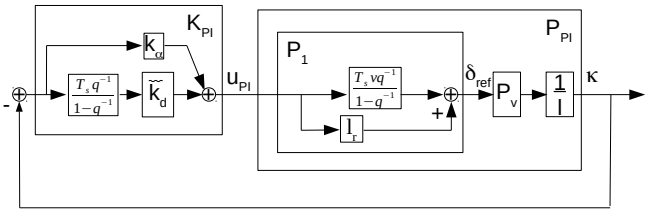


Fig. 3. Structure reformulation for a gain scheduled PI controller design

### *3.3 Modeling For Control Design*

For a model based design of parameters $k_\alpha$ and $k_d$, we have to formalise the relation between the controlled variables $d(k), e_\alpha(k)$ and the control input $\delta_{ref}(k)$. This relation can be established in two steps: 1.) by describing the vehicle dynamics that connects $\delta_{ref}(k)$ to $\delta(k) = \ell\kappa(k)$, and 2.) by relating $\kappa(k)$ with $d(k)$ and $e_\alpha(k)$. The latter characterisation is achieved in continuous time based on the the following time-derivatives (see also Fig. 1)

$$\dot{d}(t) = v(t)\sin(e_\alpha(t)) \approx v(t)e_\alpha(t), \tag{14}$$

$$\dot{e}_\psi(t) = v(t)\cos(e_\alpha(t))\kappa_{ref}(t) - v(t)\kappa(t) \tag{15}$$

$$\approx v(t)\left(\kappa_{ref}(t) - \kappa(t)\right). \tag{16}$$

where $v(t) \triangleq \|\mathrm{v}(t)\|$ is the absolute speed. In (15) we utilised the expression $\dot{\psi}(t) = v(t)\kappa(t)$ for the yaw rate.

Although the steering dynamics of the vehicle and the actuator admit a nonlinear and speed dependent behaviour, together with the steering reference tracking controller described in Section 2, the low level closed-loop system can be well approximated by a speed dependent linear system. From experimental data collected at several constant speed, a set of second order output-error models with time delay are identified. The obtained discrete time models are denoted by the rational function $P_v(q)$, where $q$ is the forward shift operator. Then the steering angle is related to the steering angle reference by

$$\delta(k) = P_v(q)\delta_{ref}(k). \tag{17}$$

The model for control design is described by the following difference equations where $T_s$ denotes the sampling time

$$\kappa(k) = \frac{1}{\ell}P_v(q)\delta_{ref}(k) \tag{18}$$

$$d(k+1) = d(k) + T_s v(k)e_\alpha(k), \tag{19}$$

$$e_\psi(k+1) = e_\psi(k) + T_s v(k)\left(\kappa_{ref}(k) - \kappa(k)\right) \tag{20}$$

$$e_\alpha(k) = e_\psi(k) - \ell_r\kappa(k). \tag{21}$$

The closed-loop model of the path tracking problem is shown in Fig. 2.

### *3.4 Output-Feedback Control Design*

The longitudinal vehicle dynamics is typically much slower than the lateral dynamics, it is, therefore, a reasonable assumption for the path tracking control design that the vehicle speed is "slowly-varying". The linear parameter-varying (speed scheduled) model of the controlled system described above allows a gain-scheduled control design approach, where the constant output-feedback controller parameters $k_\alpha$ and $k_d$ may be designed for multiple fixed (constant) speed values, and the resulted speed-parameter pairs may be interpolated during operation, or smooth regression functions may also be constructed for the speed dependent gains $k_\alpha(v)$ and $k_d(v)$.

For designing the feedback properties (stability, stability-robustness, convergence speed) of the system the feed-forward term and reference signal are temporarily omitted. Having fixed the speed, the system is linear time-invariant (LTI), thus the closed-loop scheme in Fig. 2 can be equivalently reformulated according to Figure 3 to have a standard series compensation structure

$$\kappa(k) = P_{PI}(q)u_{PI}(k), \qquad u_{PI}(k) = -K_{PI}(q)\kappa(k). \tag{22}$$

Note that the controller has a PI structure with parameters $k_\alpha(v)$ and $\tilde{k}_d(v) \triangleq vk_d(v)$. The PI controller design is standard with prescribed phase-margin and closed-loop bandwidth.

### *3.5 Analysis of Steady-State Tracking Performance*

Assuming constant speed, Z-transform can be applied to the linearized equations to obtain the transfer function from the path curvature to the tracking error $d(z) = T_{d,\kappa_{ref}}(z)\kappa_{ref}(z)$. The steady state tracking error formula $\lim_{k\to\infty} d(k) = \ell\frac{1-P_v(1)}{k_d P_v(1)}\kappa_{ref}$ for a constant path curvature $\kappa_{ref}$ follows from the finite value theorem where $P_v(1)$ denotes the value of transfer function $P_v(z)$ at $z = 1$. The control structure allows a small steady state tracking error which is proportional with the steady state steering angle tracking error of the lower level control system, and inversely proportional with control gain $k_d$.

### *3.6 Heading Offset Compensation with Kalman filtering*

The test vehicle is equipped with various positioning systems to enable simulation of different scenarios. In the current example a KVH Industries GEO-FOG 3D Dual sensor was used. This device can provide a horizontal position accuracy of 0.008 m at 10 Hz if an RTK base station is constantly available. Heading information is delivered with an accuracy of 0.01°

Depending on the availability of base station information for positioning, a slowly varying offset error may appear in the heading angle of the vehicle, which causes a distance drift in the path tracking. This offset error, denoted by $o_\psi$, affects the angle error input $e_\alpha$ of the controller, but leaves the distance error $d$ intact, since it is calculated from position coordinates. Therefore, the model for offset estimation is derived from (18)-(21) with the following modification and extension

$$d(k+1) = d(k) + T_s v(k)(e_\psi(k) - \ell_r \kappa(k)), \quad (23)$$

$$o_\psi(k+1) = o_\psi(k), \quad (24)$$

$$e_\alpha(k) = e_\psi(k) - \ell_r \kappa(k) - o_\psi(k). \quad (25)$$

Assuming Gaussian process and measurement noise, speed-scheduled steady-state Kalman filters are designed, which has the following form

$$x_K(k+1) = A_K(v(k))x_K(k) + B_K(v(k))u_x(k), \quad (26)$$

$$o_\psi(k) = C_K(v(k))x_K(k) + D_K(v(k))u_o(k), \quad (27)$$

where $A_K, B_K C_K$ and $D_K$ are state-space matrices of the current estimator (see Franklin et al. (1990)),

$$x_K = [\hat{x}_P^T, \hat{e}_\psi, \hat{d}, o_\psi]^T$$

is the estimated state vector where $\hat{x}_P$ is the estimate of the states of $P_v(q)$, $u_x = [\delta_{ref}, \kappa_{ref}, \delta, d, e_\psi - \beta]^T$ and $u_o = [0, 0, \delta, d, e_\psi - \beta]^T$ are the measurement vectors to the Kalman-filter. First, (27) is calculated with the current measurements, then instead of $e_\alpha$ by (11), the compensated angular error

$$\hat{e}_\alpha \triangleq e_\psi - \beta + o_\psi \quad (28)$$

is used for computing the control input $\delta_{ref}$. Finally the states of the filter is updated by (26).

## 4. SPLINE REPRESENTATION OF PATH

As introduced in Section 3.1, $F : \mathbb{R} \mapsto \mathbb{R}^2$ denotes the finite continuous path to be followed. It is divided into $n_S$ consecutive segments of quintic $G^2$-splines denoted by $F_i : [0,1] \mapsto \mathbb{R}^2$, $i = 1, 2, ..., n_S$, that connect smoothly so that tangent and curvature of $F(\theta)$ are continuous functions. $F$ is related to the spline segments as follows. For all $\theta \in \mathbb{R}$, $0 \le \theta \le n_S$ there exist $i = 1, 2, ..., n_S$ and $\tau \in [0,1]$ such that

$$F(\theta) = F_i(\tau) \text{ with } \theta = i - 1 + \tau. \quad (29)$$

Given the coordinates, tangent directions and curvatures at the two ends of the segments, the quintic $G^2$-splines interpolate as described in (Piazzi et al., 2002, Section III.), where four parameters $\eta_1, \eta_2, \eta_3$ and $\eta_4$ are free to shape the spline. In most cases the choice $\lambda \triangleq \eta_1 = \eta_2 = \|F_i(0) - F_i(1)\|$ with $\eta_3 = \eta_4 = 0$ is appropriate. For specific boundary conditions the minimization of the peak norm $\|\kappa_i\|_\infty = \max_{\tau \in [0,1]} |\kappa_i(\tau)|$ of the curvature vector

$$\kappa_i(\tau) = \frac{F_i'(\tau) \times F_i''(\tau)}{\|F_i'(\tau)\|^3} \quad (30)$$

might be necessary, then it is performed numerically over a grid of points $\tau \in [0,1]$ in the single parameter $\lambda$.

An important on-line task that must be performed quickly during control, is the determination of the closest point of the path to the vehicle's CG, $z(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$. This problem is solved in two steps,

(1) Check if vehicle left segment $i$

$$i(k+1) = \begin{cases} i(k) + 1 \text{ if } (z(k) - F_{i(k)}(1))^T F_{i(k)}'(1) \ge 0, \\ i(k) \text{ otherwise.} \end{cases} \quad (31)$$

(2) Find the closest point $F_{i(k)}(\tau^*(k))$ along segment $i(k)$ by finding the root $\tau^*(k)$ of the equation $E(\tau) \triangleq (z(k) - F_{i(k)}(\tau))' * \frac{F_{i(k)}'(\tau)}{\|F_{i(k)}'(\tau)\|} = 0$. An appropriate fast method is Newton's iteration described by Algorithm 1. Having find $\tau^*(k)$, the distance, reference curvature and heading angle of the path are computed respectively as follows

$$d_F(z(k)) = \frac{\det[z(k) - F^*,\ V^*]}{\|V^*\|_2}, \quad (32)$$

$$\kappa_{ref}(k) = \frac{\det[V^*,\ Q^*]}{\|V^*\|^3}, \quad (33)$$

$$\psi_{ref}(k) + \beta_{ref}(k) = \arctan_2(V_2^*, V_1^*), \quad (34)$$

where $F^* \triangleq F_{i(k)}(\tau^*(k))$, $V^* \triangleq F_{i(k)}'(\tau^*(k))$ and $Q^* \triangleq F_{i(k)}''(\tau^*(k))$.

$j = 0$, $\tau_j = \tau^*(k-1)$, Accuracy = 0.01 [m], $a = \infty$, MaxIter = 3
**while** *a>Accuracy and j<MaxIter* **do**
  $\tau_{j+1} := \tau_j - \frac{E(\tau_j)}{E'(\tau_j)}$
  $a := \|F_{i(k)}(\tau_{j+1}) - F_{i(k)}(\tau_j)\|_2$
  $j := j + 1$
**end**
$\tau^*(k) := \tau_j$

**Algorithm 1.** Newton's iteration to find the closest point of the spline to vehicle's CG point $z(k)$

## 5. PATH PLANNING BASED ON POTENTIAL FIELDS

Whenever the local goals are determined by the *behaviour planner*, the path planning layer of an autonomous vehicle is responsible for the design of the optimal path in the range of reliable environment sensing. Multiple approaches exist that have been tested experimentally. Graph search methods seek globally optimal routes over a long horizon of discretised configuration space; variational methods solve a nonlinear continuous constrained optimisation problem by nonlinear programming, for a survey see Paden et al. (2016). A special case of the latter approach when constraints on the path are built in the cost function by using penalty or barrier functions. In the method of artificial potential fields proposed by Khatib (1985), the optimum point is the target point, which is found by gradient search. An alternative interpretation is that the vehicle is placed in a force field which is generated by the potential field, and the resultant of attractive and repulsive forces pull the vehicle towards the target. An advantage of the method is low computational complexity, Hongyu et al. (2018); Bayat et al. (2018).

Assuming that the range of vision is short, a gradient search is applied on-line, which drives the vehicle ahead. A potential field $U$ is associated with the knowledge of the environment: obstacles like surrounding vehicles or pedestrians, $U_o$, lanes and road-banks, $U_l$, the local target point, $U_t$, etc, each may depend on location and time:

$$U(x,y,k) = U_t(x,y,k) + \sum_{j=1}^{n_o} U_{o,j}(x,y,k) + \sum_{j=1}^{n_l} U_{l,j}(x,y,k). \quad (35)$$

In contrast to the conventional approaches, the force field generated by the potential field acts on and modifies the reference path, in this paper, instead of affecting directly the vehicle's motion. In this way the low level path tracking control and higher level path planning tasks are kept separated. Uncertainty due to path tracking errors can be modelled by bounds which can be built into the potential functions at the path planning level. Further development on any of the control levels does not have conceptual influence on the other level.

Let $z = [x, y]^T$ denote any point in the space, $F(\theta^*(x, z))$ the closest point of the path to $z$, where $\theta^*(x, y) = \arg\min \|z - F(\theta)\|$. Let $d(x, y)$ be the signed distance between point $z$ and the nominal path. Similarly, let $z_j = [x_j, y_j]^T$, $F(\theta_j^*)$ and $d_j$ denote the location of an obstacle reference point, the closest point on, and its signed distance to the nominal path. Let $s_j(x, y)$ denote the arc length on nominal path between the two points $F(\theta^*(x, z))$ and $F(\theta_j^*)$,

$$s_j(x, y) = \int_{\theta^*(x,y)}^{\theta_j^*} \|F'(\theta)\| d\theta. \tag{36}$$

Then the potential function defined by obstacle $j$ is chosen to be

$$U_{o,j}(x, y, k) = c_{o,j} e^{-\left(\frac{s_j^2(x,y)}{\sigma_{s,j}^2} + \frac{(d_j - d(x,y))^2}{\sigma_{d,j}^2}\right)} \tag{37}$$

The potential functions for road banks are

$$U_{l,1}(x, y, k) = c_{l,1} e^{(d(x,y) - d_{right})} \tag{38}$$

$$U_{l,2}(x, y, k) = c_{l,2} e^{(d_{left} - d(x,y))}, \tag{39}$$

where $y_{right}, y_{left}$ are the distances of the road-bank from the nominal path. Let $s_t(x, y)$ denote the arc length between $F(\theta^*(x, z))$ and the closest path point to the target point, then the potential function associated with the target is defined by

$$U_t(x, y, k) = c_t s_t(x, y), \quad c_t < 0. \tag{40}$$

Parameters $c_o, c_l, c_t, \sigma_{s,j}, \sigma_{d,j}$ of the potential functions may depend on time according to the behaviour of the obstacle and their relative position to the vehicle (asymmetry).

The gradient field (force field) at the current path $F(\theta)$ defines the new target path as follows. At each discrete time $k$ let $\theta(k) = i(k) - 1 + \tau^*(k)$, i.e., $F(\theta(k))$ is the target point on the nominal path at the current location of the vehicle. A new target point $F^{(0)}$ is defined in the direction of the negative gradient of the potential at the current target point

$$F^{(0)} := F(\theta(k)) - \text{grad}_{x,y} U(F(\theta(k)), k). \tag{41}$$

Then the subsequent points are generated by iteration

$$F^{(j+1)} := F^{(j)} - \text{grad}_{x,y} U(F^{(j)}, k), \quad j = 0, 1, 2, .... \tag{42}$$

The obtained sequence of points may be used to construct a new quintic $G^2$ spline sequence. We used instead linear interpolation and modified only the distance error $d(k)$ and angular error $e_\alpha(k)$ inputs of the feedback controller in (10), while we left the curvature feed-forward term intact.

## 6. EXPERIMENTAL VALIDATION

In this section the operation of the control system is demonstrated through an experiment. The scene can be

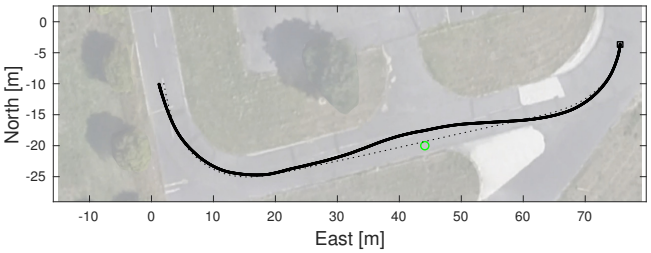


Fig. 4. Vehicle position while avoiding an obstacle.

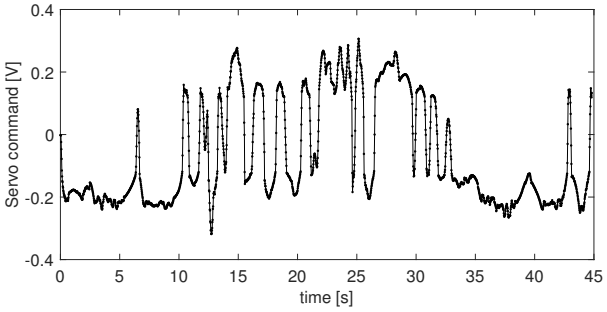


Fig. 5. Control signal to the servo unit [V]. The dead-zone is quickly leaved by controller (1)

observed in Fig. 4 showing the path of the vehicle during a manoeuvre of avoiding a static obstacle marked by a green circle. The nominal path is plotted by a thin dotted line.

At this phase all tests were conducted on dry, flat asphalt roads in generally good conditions. The performance implications of varying environmental factors such as rain, dirt or potholes will be tested later.

However the trajectory planning layer of the control system is capable of calculating a safe cornering speed according to Mihaly et al. (2019), the current tests were carried out at a fixed speed for easy repeatability. The speed in the example case was set to 9 km/h.

The next two figures show the performance of the low level steering angle reference tracking controller. The control signal to the steering servo actuator is plotted in Fig. 5, where the effect of the dead-zone about $[-0.15V, 0.1V]$ can bee seen. The amplification of the control gains in the interval make the controller quickly leave the dead-zone. The approach is effective on the tracking performance, however, still not perfect, as can be seen in Fig. 6. The fluctuation around the reference signal cannot be felt by the passengers, yet modelling of the speed-dependent dead-zone and a more accurate tracking control is necessary to decrease the stress of the actuator.

The yaw angle of the vehicle is provided by the GPS unit. It is plotted by black line in Fig. 7 and can be compared to the calculation $\arctan_2(z(k) - z(k-1))$ plotted by red line. The difference (green dash-dot line) is composed by the offset error of the sensor and the noise in the position data $z(k)$. A self-correction by the sensor can be seen at 12s. The Kalman-filter in Section 3.6 is designed to track this offset error, the result is plotted by the blue dotted line. It significantly improves the path tracking performance of the system.

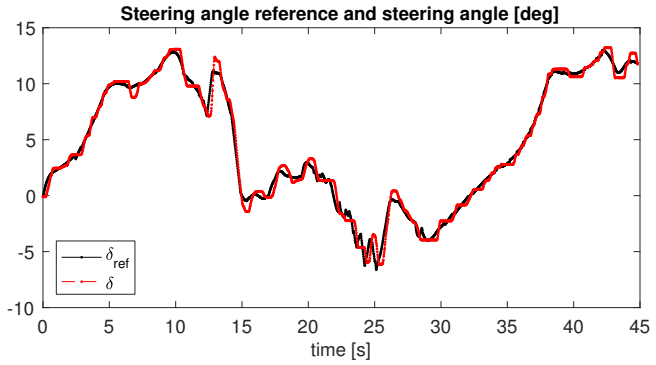


Fig. 6. Steering angle reference and measured steering angle [deg]. The effect of dead-zone can be observed.

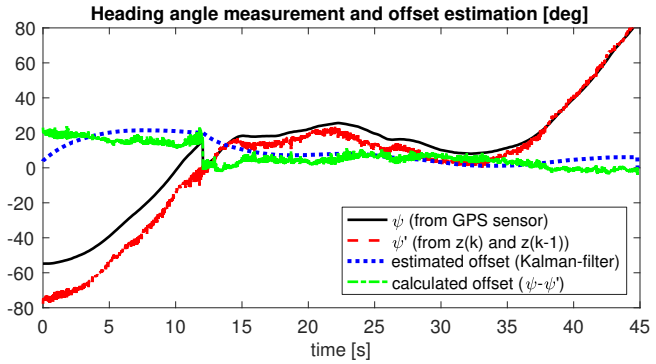


Fig. 7. Heading offset estimation by Kalman-filtering.

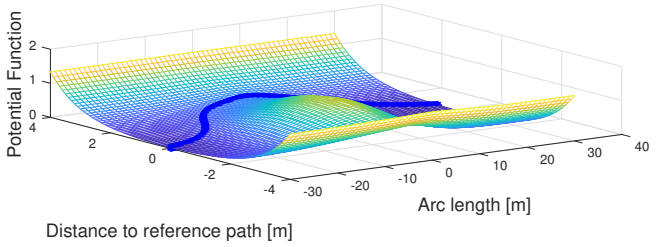


Fig. 8. Two components of the potential function associated with the obstacle and the road banks ($U_t(x, y, k) = 0$). The coordinates are defined with respect to the nominal path. The modified path (thick blue line) is obtained by the iteration (42), where the force vector is the negative gradient of the full potential function including the term of the target direction $U_t(x, y, k)$.

The functioning of the path planning algorithm can be tracked in the next two figures. Fig. 8 show the potential field associated with the obstacle and the road banks. (The total potential field is obtained by adding a slope in the increasing arc length direction.) The results of the path planning iteration (42) is represented by the thick blue line which follows the valley-bottom of the potential function as the effect of the force (gradient) field shown in Fig. 9.

## 7. CONCLUSION

The components of the base-line control system of an autonomous vehicle has been detailed in the paper. Steering reference tracking, path following and planning, sensor

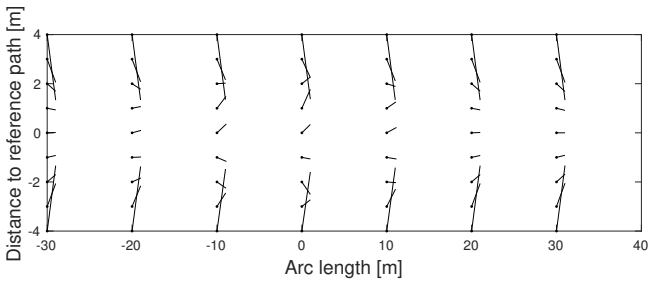


Fig. 9. Force field (negative gradient) of the potential function

offset estimation and compensation, and obstacle avoidance problems are presented and tested. Experimental results in a navigation problem validated the functioning of the system. It is concluded that the autonomous vehicle platform initiated for supporting education and research is capable to test and analyse methods for autonomous vehicle control problems.

In the future new, or other well known algorithms can be tested, compared and demonstrated on this platform.

## REFERENCES

Bayat, F., Najafinia, S., and Aliyari, M. (2018). Mobile robots path planning: Electrostatic potential field approach. *Expert Systems with Applications*, 100, 68 – 78. doi:https://doi.org/10.1016/j.eswa.2018.01.050.

Franklin, G., Powell, J., and Workman, M. (1990). *Digital Control of Dynamic Systems, Second Edition*. Addison-Wesley.

Goodwin, G.C., Graebe, S.F., and Salgado, M.E. (2000). *Control System Design*. Prentice Hall PTR.

Hongyu, H., Chi, Z., Yuhuan, S., Bin, Z., and Fei, G. (2018). An improved artificial potential field model considering vehicle velocity for autonomous driving. *IFAC-PapersOnLine*, 51(31), 863 – 867. doi: https://doi.org/10.1016/j.ifacol.2018.10.095. 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, 500–505. doi: 10.1109/ROBOT.1985.1087247.

Mihaly, A., Kisari, A., Gaspar, P., and Nemeth, B. (2019). Adaptive semi-active suspension design considering cloud-based road information. *IFAC AAC 2019*, 249–254.

Paden, B., Čáp, M., Yong, S.Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1), 33–55. doi: 10.1109/TIV.2016.2578706.

Piazzi, A., Bianco, C.G.L., Bertozzi, M., Fascioli, A., and Broggi, A. (2002). Quintic g2-splines for the iterative steering of vision-based autonomous vehicles. *IEEE Transaction on Intelligent Transportation Systems*, 3(1), 27–36.

Rajamani, R. (2011). *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer.