

Vehicle Localization and Control on Roads with Prior Grade Map

Roya Firoozi, Jacopo Guanetti, Roberto Horowitz, Francesco Borrelli

Abstract—We propose a map-aided vehicle localization method for GPS-denied environments. This approach exploits prior knowledge of the road grade map and vehicle on-board sensor measurements to accurately estimate the longitudinal position of the vehicle. Real-time localization is crucial to systems that utilize position-dependent information for planning and control. We validate the effectiveness of the localization method on a hierarchical control system. The higher level planner optimizes the vehicle velocity to minimize the energy consumption for a given route by employing traffic condition and road grade data. The lower level is a cruise control system that tracks the position-dependent optimal reference velocity. Performance of the proposed localization algorithm is evaluated using both simulations and experiments.

I. INTRODUCTION

Localization in autonomous driving applications refers to determining the vehicle's position and attitude. Navigation, motion planning and real-time control rely on accurate localization. Inaccurate localization can be detrimental to performance and may lead to accidents. The Global Positioning System (GPS) is able to provide positioning with meter-level accuracy in clear open sky, but since its accuracy suffers from degraded satellite availability or multi-path error in city areas, a stand-alone GPS is not reliable. A common method to overcome these limitations is using dead-reckoning techniques. GPS data and dead-reckoning information extracted from vehicle's on-board sensors can be fused to improve the position estimate of the vehicle [1], [2]. Since dead-reckoning estimates the current position based on a previously determined position, it is subject to significant cumulative error. In sparse GPS environments like tunnels, underground passages or forested paths, using only dead-reckoning may lead to large estimation errors. A solution to this problem is incorporating sensor data with a prior road map to correct the position estimates. Map-aided localization approaches exploit road features including lane markings, traffic signs, curbs and buildings to correlate the vehicle states to the prior road map.

Today's self driving cars equipped with three-dimensional (3D) Light Detection and Ranging (LIDAR) scanners are capable of localizing themselves with centimeter-level accuracy. High accuracy localization requires a high definition map of the environment. LIDAR generates point clouds by illuminating laser pulses to the surroundings and measuring the reflected results. In order to build a high-resolution map of the environment, a survey vehicle is driven in the road and

measurements of GPS, Inertial Measurement Unit (IMU), wheel odometry and LIDAR data are integrated together. Once an a-priori map of the environment is generated, the vehicle can match the features extracted from LIDAR point clouds with the features stored in the map to correct its position estimate in real-time [3], [4]. Since high-accuracy LIDAR is expensive, a cheaper way to do localization is to use LIDAR for mapping and camera for localization. A survey vehicle equipped with LIDAR generates a Point Cloud Map (PCL) of the road and autonomous cars can make use of just camera to localize itself within the (3D) pre-built PCL [5].

Vision-based matching of road features to a pre-built map of the environment can augment localization performance[6]. However, these feature-based localization methods yield accurate localization in feature-rich environments like urban areas. In rural and suburbs areas, where density of features decreases dramatically, these approaches might not be reliable. A feature that is present in both urban and country roads is road grade. We can make use of a road grade map as the a-priori map for localization. The authors of [7] and [8] utilize a road terrain map and in-vehicle pitch sensor measurements to localize the vehicle using particle filter. This paper proposes a simpler localization method using an Extended Kalman Filter (EKF). We use accelerometer and wheel speed sensors together with the road grade map to accurately estimate the vehicle's longitudinal position. Although the application of our proposed approach is restricted to hilly roads, it can correct dead-reckoning in extended GPS blackout or augment IMU/GPS system where GPS signal is available. Also this method can be combined with other techniques discussed above to improve the localization performance. Applications that benefit from localization include planning and control systems that rely on the route spatial data like road grade and curvature, speed limit, road optimal reference velocity trajectory and distance to the next stop sign or traffic signal. We validate our localization approach on a control architecture comprised of two levels. At the high-level, we optimize the vehicle velocity trajectory over a given route, by incorporating traffic flow and road grade data into the problem. At the low-level, we run a Model Predictive controller (MPC) that follows the optimal reference velocity trajectory calculated by a high-level planner. Our simulations show how errors in position estimation affect energy consumption and confirm the effectiveness of the proposed approach.

The rest of the paper is structured as follows. Section II describes road grade map generation and presents the localization approach using EKF formulation. Section III shows

The authors are with the Department of Mechanical Engineering, at the University of California, Berkeley.
{royafiroozi, jacopoguanetti, horowitz, fborrelli}@berkeley.edu

simulation and experimental results for localization. Section IV explains the hierarchical control system. Section V illustrates the simulation results for control system and Section VI makes concluding remarks.

II. LOCALIZATION

In case of temporary GPS loss, odometry sensors such as wheel speed encoder can be used to estimate the change of position. However, this method is sensitive to errors due to the integration of velocity measurements over time. To mitigate this cumulative error, we propose to fuse the sensor measurements from wheel speed encoders and accelerometer with grade data extracted from a global road map using the EKF. This section describes road grade map generation and process and measurement models that are included in the EKF estimator.

A. Road Grade Map Generation

Roadway elevation data is crucial for many transportation applications. To generate a grade map for the entire road, a high-precision Real Time Kinematic (RTK) GPS can be used to measure latitude, longitude and altitude along the route. Also several APIs such as Google Elevation API provide elevation data for all locations on earth surface. We can query elevation data for specified coordinates and obtain the road elevation profile. This API also provides the resolution of each elevation sample, defined as the maximum horizontal distance between data points from which the elevation was interpolated. Digital Elevation Models (DEMs) data throughout the earth's surface has been provided by the NASA Shuttle Radar Topographic Mission (SRTM). This elevation data in the United States is available in National Elevation Dataset (NED), provided by the Geological Survey (USGS), at resolutions between 1 arc-second (about 30 meters) and 1/9 arc-second (about 3 meters) depending on the location. The authors of [9] assessed the accuracy of road elevation data extracted from Google Earth by reporting root mean squared error (RMSE) and standard deviation of roadway elevation error are 2.27 meters and 2.27 meters, respectively, even in the areas where USGS NED provides 1/3 arc-second (about 10 meters) resolution.

B. Extended Kalman Filter (EKF)

We model the vehicle as a point mass moving along a path with velocity v . The system state at time t is

$$x(t) = [s(t) \ v(t)]^T,$$

where $s(t)$ and $v(t)$ are the vehicle position and velocity respectively. The process model is a nonlinear function f of the state x and control input u . Process noise w is assumed to be additive and normally distributed with zero mean and covariance of Q , $w \sim \mathcal{N}(0, Q)$. The measurement model is also a nonlinear function h that maps the state x to output y . The measurement noise is assumed to be additive Gaussian noise with zero mean and covariance of

R , $\eta \sim \mathcal{N}(0, R)$. EKF linearizes the nonlinear system about the current estimates.

$$\begin{aligned} \dot{x} &= f(x, u) + Gw \\ y &= h(x) + \eta. \end{aligned} \quad (1)$$

C. Process model

The longitudinal kinematic model of the vehicle is defined as

$$\begin{aligned} \dot{s} &= v \\ \dot{v} &= a + w, \end{aligned}$$

where a is the longitudinal acceleration treated as a known input. The system is discretized using Euler method with constant sampling time interval Δt .

$$\begin{aligned} s(t+1) &= s(t) + v(t) \Delta t \\ v(t+1) &= v(t) + (a(t) + w(t)) \Delta t. \end{aligned}$$

Since Δt is small, the second-order terms are neglected. The longitudinal acceleration $a(t)$ is derived from the acceleration measured by the vehicle accelerometer. When a vehicle is moving on an inclined road, the longitudinal acceleration of the sensor is affected by gravity,

$$a_{sensor}(t) = a(t) + g \sin(\theta(t)), \quad (2)$$

where a_{sensor} denotes acceleration measured by the accelerometer, g is gravitational acceleration and θ is slope of the road. By substituting $\sin(\theta)$ with a prior known grade map $p(s)$, longitudinal acceleration is calculated as

$$a(t) = a_{sensor}(t) - g p(s(t)).$$

Thus, the prediction model is

$$\begin{aligned} s(t+1) &= s(t) + v(t) \Delta t \\ v(t+1) &= v(t) + (a_{sensor}(t) - g p(s(t))) \Delta t + w(t) \Delta t. \end{aligned} \quad (3)$$

D. Measurement Model

The vehicle speed v_m is measured by the wheel speed encoders and the inclination $\theta(t)$ can be constructed by measurements of IMU accelerometer and wheel speed encoder. By reformulating (2) the inclination $\theta(t)$ can be indirectly measured as

$$\theta_m = \sin^{-1} \left(\frac{a_{sensor} - \dot{v}}{g} \right), \quad (4)$$

where \dot{v} is the vehicle acceleration obtained by differentiating the longitudinal velocity measured from the wheel speed sensor. Measurement noises for both IMU and wheel encoder sensors are assumed to be additive with Gaussian distributions $\eta^v \sim \mathcal{N}(0, \sigma_v^2)$ and $\eta^\theta \sim \mathcal{N}(0, \sigma_\theta^2)$. The measurement covariance matrix is defined as $R = \text{diag}(\sigma_v^2, \sigma_\theta^2)$. The measurement model is

$$\begin{aligned} v_m(t) &= v(t) + \eta^v \\ \theta_m(t) &= \sin^{-1}(p(s(t))) + \eta^\theta. \end{aligned} \quad (5)$$

(3) and (5) define the f and h functions in the general formulation (1), respectively.

E. Inertial Sensor Noise Filtering and Bias Removal

Since the acceleration signal measured by IMU is noisy, a low pass filter is used for noise reduction. Also, inertial sensors are subject to bias drift. In the absence of input, the IMU accelerometer reads non-zero output. This bias should be compensated, but the challenge is that bias drifts over time due to temperature changes and other factors. Hence, bias drift should be modeled mathematically and removed from the measurements. In this study we assume the bias drifts linearly as a function of time. Therefore (4) is reformulated as

$$\theta_m = \sin^{-1}\left(\frac{a_{sensor} - \dot{v}}{g}\right) - \underbrace{bt}_{\text{bias}}, \quad (6)$$

where t is the time and b is the bias parameter. We will discuss bias parameter estimation in more detail in the next section.

III. LOCALIZATION RESULTS

The performance of the algorithm has been examined through an experiment. Assuming that the GPS signal is lost, we compare the localization performance of our method to velocity integration. The absolute position error e is defined as

$$e = \hat{s} - s,$$

where \hat{s} is the estimated position and s is the true position. The experiment was carried out in a hilly road located near University of California Berkeley. The test vehicle is a Hyundai Genesis equipped with OTS (Oxford Technical Solutions) RT2002 sensing system which is comprised of a high precision GPS and an IMU. RT2002 GPS without its base-station GPS receiver provides positioning with 0.6m Circular Error Probable (CEP) accuracy. We drove the car on the road and collected the position and altitude data measured by the RT2002 GPS system to build a high-resolution road grade map. The obtained map is compared with the map generated by Google elevation API in Fig. 1. To measure the inclination, bias drift is modeled as a linear function of time and removed from the measurement. To determine the mathematical model of the acceleration bias drift, first the road slope profile for a segment of the road is computed using (4) with collected acceleration and velocity data. Then the computed slope profile θ_n is compared against the slope profile obtained by Google Elevation API θ_{Google} . The difference between the two profiles is equivalent to the bias term in (6),

$$e_{slope}(t) = \theta_n(t) - \theta_{Google}(t) \simeq bt.$$

The bias parameter b is estimated by fitting a linear function to the above slope error profile $e_{slope}(t)$ using least squares. In this study, the high-precision RT2002 IMU is used to obtain the acceleration measurements. However, by accurate modeling of the acceleration bias drift, the vehicle's on-board accelerometer can be used for acceleration measurement. After generating the high-resolution grade map of the road, the position is estimated using the proposed localization approach. Although we don't use GPS data to estimate the

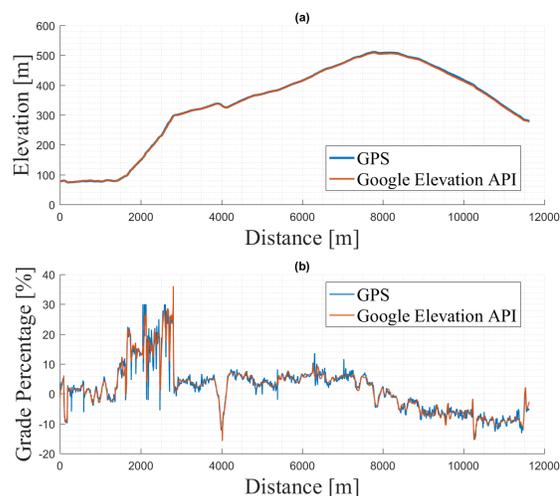


Fig. 1. Elevation and grade maps are generated using both GPS and Google Elevation API for a hilly route located near UC Berkeley.

TABLE I

EXPERIMENT RESULTS: COMPARISON OF ROOT-MEAN-SQUARE ERROR

Experiment	Position RMSE [m]	
	Velocity Integration	EKF Estimate
1	25.0	3.7
2	17.9	9.4
3	21.3	4.3
Average	21.4	5.8

position, we still collect them and use them as the ground truth. We ran 3 experiments and calculated RMSE for both methods in Table I. As shown, the EKF algorithm achieves significantly smaller RMSE than velocity integration.

Table II compares the drift error of the two methods for just the first experiment. As seen, the estimated position calculated by velocity integration drifts over time. The absolute error between the velocity integration estimated position and the ground truth at final point is 30 times larger than EKF estimated position. According to the results, EKF estimates have small errors, whereas position estimates calculated by integration of wheel speed diverge as time goes on.

IV. CONTROL SYSTEM ARCHITECTURE

In this section we describe an example application of the proposed localization algorithm, a hierarchical control system that exploits position-dependent information for planning and control. The system architecture is shown in Fig. IV. The goal is to plan an energy-optimal velocity trajectory for a given route offline and to follow the ob-

TABLE II

EXPERIMENT RESULTS: COMPARISON OF DRIFT ERROR OF FINAL POSITION ESTIMATE

Method	Absolute Error [m]	Error Percentage [%]
Velocity Integration	60.3	0.52
EKF estimate	2.4	0.02

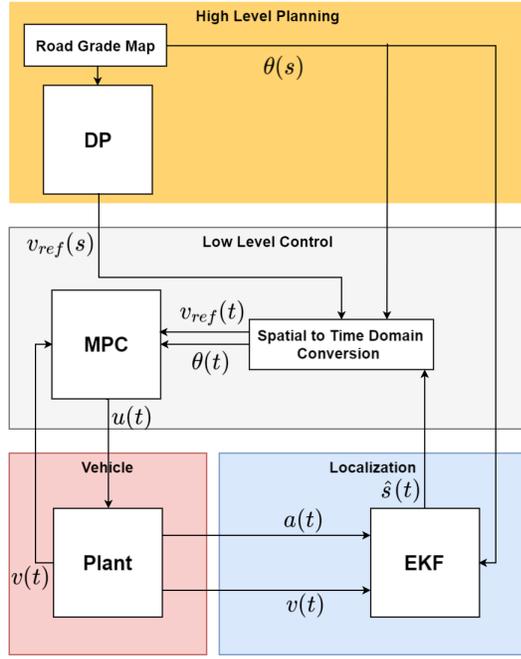


Fig. 2. System Architecture: position-dependent data including reference velocity $v_{ref}(s)$ and road grade $\theta(s)$ are converted from spatial domain to time domain

tained reference trajectory in real-time. Both the offline and online formulations with the vehicle longitudinal model are explained in the following sections.

A. Vehicle Model

The longitudinal motion of a vehicle moving on an inclined road can be modeled by the following force balance in the longitudinal direction

$$m\dot{v} = \underbrace{F_{traction} - F_{brake}}_{u(t)} - F_{airdrag} - F_{rolling} - F_{gravity}, \quad (7)$$

where m is mass of the vehicle, \dot{v} is the longitudinal acceleration and $F_{traction}$ and F_{brake} are throttle and brake forces, respectively and $F_{traction} - F_{brake}$ is the total force taken as control input $u(t)$. The aerodynamic drag is determined by vehicle speed v , air density ρ , air drag coefficient C_d and frontal area A_f .

$$F_{airdrag} = \frac{1}{2}\rho C_d A_f v^2.$$

The rolling resistance is defined as

$$F_{rolling} = mgC_r \cos(\theta),$$

where g is gravitational force, C_r is rolling friction coefficient and θ is road slope. The gravity force due to road grade can be expressed as

$$F_{gravity} = mg \sin(\theta).$$

B. High-Level Planner: Dynamic Programming Formulation

Traffic condition and roadway grade have a significant impact on the vehicle's energy consumption and emission. For a given route with specified origin and destination, the energy-optimal velocity trajectory along the route can be determined by employing information about traffic condition and road grade. To find the optimal position-dependent velocity profile, it is convenient to convert the optimization problem formulation from time domain to spatial domain,

$$\dot{v} = \frac{dv}{dt} = \frac{dv}{ds} \frac{ds}{dt} = \frac{dv}{ds} v.$$

Velocity (v) is a state variable and travelled distance (s) is the independent variable. The vehicle longitudinal dynamics (7) in the spatial domain is

$$\frac{dv}{ds} = \frac{1}{mv} (u(s) - \underbrace{F_{airdrag} + F_{rolling} + F_{gravity}}_{F_{total}}), \quad (8)$$

which is reformulated as

$$v(s+1) = (v(s)^2 + \frac{2\Delta s}{m} F_{total})^{\frac{1}{2}}, \quad (9)$$

after trapezoid discretization.

The goal is to minimize the total power consumption over the entire route within a reasonable trip time interval. Therefore, the cost function is a trade-off between power consumption and trip time. The first term of the cost function (10) penalizes the wheel power consumption and the second term compensates deviation from maximum speed profile to guarantee a short trip time. The objective

$$J(s_{cur}) = \sum_{s=s_{cur}}^{s_{final}} (v(s)u_p(s) + \gamma(v(s) - v_{max}(s))^2)\Delta s \quad (10)$$

is minimized, where s_{cur} and s_{final} represent the current and final positions respectively. γ is a weight factor, $u_p(s)$ denotes acceleration force in longitudinal direction and $v_{max}(s)$ is velocity upper bounds at position s . Although $u(s)$ can hold positive or negative values, $u_p(s)$ in the cost function (10) refers to only positive inputs. The problem is formulated as

$$\begin{aligned} \min_{u(s)} J &= \sum_{s=s_{cur}}^{s_{final}} (v(s)u_p(s) \\ &\quad + \gamma(v(s) - v_{max}(s))^2)\Delta s \\ \text{s.t.} \quad v(s+1) &= f(v(s), u(s)), \\ v_{min}(s) &\leq v(s) \leq v_{max}(s), \\ u_{min} &\leq u(s) \leq u_{max}, \\ u_p(s) &= u(s) \quad \text{if } u(s) > 0 \\ u_p(s) &= 0 \quad \text{if } u(s) \leq 0 \\ v(0) &= 0, \\ v(s_{final}) &= 0, \end{aligned} \quad (11)$$

where the objective (10) is minimized while satisfying the longitudinal dynamics (8), denoted as f , as well as state and input constraints and boundary conditions. $v_{min}(s)$ and

$v_{max}(s)$ are velocity lower and upper bounds at position s and u_{min}, u_{max} are control inputs lower and upper bounds, respectively. The velocities at the origin and destination are assumed to be zero.

The nonlinear optimization problem (11) is solved by dynamic programming (DP) with the solver from [10]. The state and input spaces are discretized according to their corresponding upper and lower bounds. To incorporate the effect of traffic condition, the velocity at each step is upper bounded according to the traffic flow data acquired for a specific route. If data is not available, the road speed limit $v_{max}(s)$ is used as upper bound. Road slope information of the route is also included in the vehicle dynamic model (9). Thus, the DP algorithm makes use of the road geometry data to find the optimal reference velocity trajectory. Starting from destination, the DP proceeds backward and evaluates the optimal cost-to-go function (10) at each node based on Bellman's principle of optimality. The backward sweep outputs a map of optimal cost and optimal policy over the state grid. In the forward sweep, DP starts from the route origin and applies the obtained optimal control policy to calculate the optimal velocity trajectory. The optimized velocity $v(s)$ will be set as the reference velocity $v_{ref}(s)$ for the low-level controller.

C. Low-Level Controller: MPC Formulation

The low-level controller is a cruise control system that controls the longitudinal dynamics of the vehicle to track the reference velocity generated by the high-level planner and minimize the control effort in real time. To design the controller, we implemented a Model Predictive Control (MPC) algorithm. The proposed controller solves the constrained finite horizon optimization problem

$$\begin{aligned} \min_{u(\cdot|t)} \quad & J = \sum_{k=t}^{t+N} \|v(k|t) - v_{ref}\| \\ & + \gamma \sum_{k=t}^{t+N-1} \|u(k|t)\| + p(v(t+N|t)) \quad (12) \\ \text{s.t.} \quad & v(k+1|t) = f(v(k|t), u(k|t)), \\ & v_{min} \leq v(k|t) \leq v_{max}, \\ & u_{min} \leq u(k|t) \leq u_{max}, \end{aligned}$$

where N is the MPC horizon, $v(k|t)$ and $u(k|t)$ are the state variable and control input at step k predicted at time t , respectively, γ is a weight factor, f represents the vehicle longitudinal dynamics (7) and v_{ref} is the reference velocity trajectory obtained from the high level planner. A terminal cost p is introduced as a constant large value.

The proposed MPC controller extracts grade data from the provided road grade map. It incorporates this prior knowledge to accurately predict the future longitudinal velocity of the vehicle over the horizon and plan accordingly. Both grade and optimal reference velocity are defined as position-dependent profiles for a specific route. In order to employ these data, vehicle has to localize itself and estimate its position with respect to the map. Since our MPC operates

TABLE III
MODEL PARAMETERS

m	vehicle mass	kg	1360
A_f	vehicle frontal surface	m^2	2.30
ρ	air density	kg/m^3	1.225
C_d	vehicle drag coefficient	-	0.24
C_r	vehicle roll coefficient	-	0.01
T_s	sampling time	s	0.2

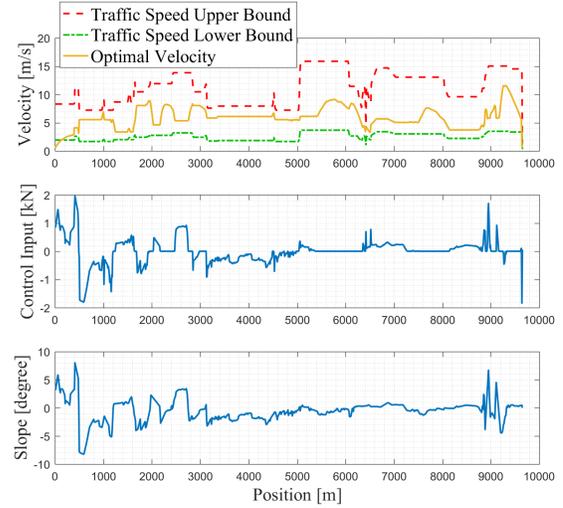


Fig. 3. Optimal velocity is bounded by traffic speed profile upper and lower bounds and control policy is optimized based on road slope map.

in the time domain and route information is available in the spatial domain, these forecasts are projected in the time domain assuming constant velocity over the horizon of MPC.

V. CONTROL RESULTS

A. High-level planner

An urban route near UC Berkeley (from UC Berkeley Etcheverry Hall to the Richmond Field Station) is selected and traffic flow velocity as well as grade map are obtained through the HERE API and Google Elevation API, respectively. The traffic flow velocity profile is taken as the upper-bound for velocity. The lower-bound is taken as half of the upper bound. The parameters of the longitudinal dynamic model are shown in Table III. The controller parameters are presented in Table IV. Fig. 3 illustrates the traffic flow

TABLE IV
CONTROLLER PARAMETERS

v_{min}	minimum velocity	m/s	traffic speed lower bound
v_{max}	maximum velocity	m/s	traffic speed upper bound
u_{min}	minimum control input	kN	-3
u_{max}	maximum control input	kN	3

velocity profile as well as the velocity profile obtained by trajectory planning optimization carried out via DP. The grade map of the road is also illustrated. Table V presents energy consumption as well as total trip time for traffic flow velocity profile and optimal velocity profile. As seen, the

TABLE V

DP RESULTS: COMPARISON OF THE TWO VELOCITY PROFILES

Velocity Profile	Energy Consumption [kWh]	Trip Time [min]
Traffic Flow	0.9	20.4
Optimal ($\gamma = 10$)	0.47	25.7
Optimal ($\gamma = 0.1$)	0.4	31.4
Improvement [%] ($\gamma = 10$)	47.8 %	-26.0 %
Improvement [%] ($\gamma = 0.1$)	55.5 %	-53.9 %

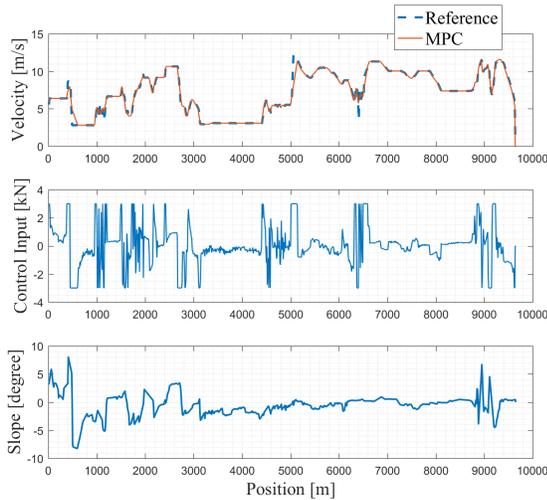


Fig. 4. MPC controller tracks the optimal reference velocity trajectory obtained from high-level planner and simultaneously minimizes the control input by incorporating road grade data in its short-term planning.

energy consumption with the optimal velocity profile (for $\gamma = 10$) is 47.8% less than with the traffic flow profile, while the trip time is 26.0% longer.

B. Low-Level Controller

We formulated the optimization problem (12) in YALMIP [11]. MPC parameters are the same as DP parameters presented in Table IV and the MPC horizon is selected as 5. Fig. 4 shows the simulation results of the MPC controller in closed-loop with the vehicle longitudinal dynamics model. As illustrated, the controller tracks the optimal reference velocity calculated in previous section. At the same time it minimizes the control effort by employing the road prior grade knowledge.

Now by assuming 0.5% of localization error based on the results in the table II, after about less than 12 km of traveling, the position estimate can be off by 60 meters. If the controller extracts position-dependent data including optimal reference velocity and road grade associated with incorrect position, the energy efficiency can significantly be affected by this error. As an example, we selected a portion of the road shown in Fig. 4 between position 8500 m to 9500 m and assumed that the controller employs knowledge of 60 meters ahead instead of its true position. Table VI compares the energy consumption for both cases. The results show that energy efficiency can be affected by about 15% in case of a 0.5% localization error.

TABLE VI

ENERGY CONSUMPTION COMPARISON

Position	Energy Consumption [kWh]
True Position	0.20
60 meters ahead	0.23

VI. CONCLUSION

We presented a localization method for autonomous driving in GPS-denied areas using a prior grade map. According to the results when there is no GPS signal, vehicle can use its own on-board sensors and a prior road grade map to localize itself relative to the map and correct its position estimate. Furthermore, we developed an energy-efficient control system by exploiting traffic information as well as road slope and speed limit data. We verified that errors in localization can significantly impact energy efficiency.

REFERENCES

- [1] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "Gps/imu data fusion using multisensor kalman filtering: Introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006.
- [2] L. R. Sahawneh, M. A. Al-Jarrah, K. Assaleh, and M. F. Abdel-Hafez, "Real-time implementation of gps aided low-cost strapdown inertial navigation system," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 527–544, Jan. 2011.
- [3] W. Burgard, O. Brock, and C. Stachniss, "Map-based precision vehicle localization in urban environments," in *Robotics: Science and Systems III*. MIT Press, 2008, pp. 352–.
- [4] L. Wang, Y. Zhang, and J. Wang, "Map-based localization method for autonomous vehicles using 3d-lidar.," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 276–281, 2017, 20th IFAC World Congress.
- [5] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular camera localization in 3d lidar maps," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1926–1931, 2016.
- [6] I. Miller, M. Campbell, and D. Huttenlocher, "Map-aided localization in sparse global positioning system environments using vision and particle filtering," *Journal of Field Robotics*, vol. 28, no. 5, pp. 619–643, 2011.
- [7] A. J. Dean, P. K. Vemulapalli, and S. Brennan, "Highway evaluation of terrain-aided localization using particle filters," Jan. 2008.
- [8] A. J. Dean, R. D. Martini, and S. Brennan, *Terrain-based road vehicle localization using particle filters*, Jul. 2008.
- [9] Y. Wang, Y. Zou, K. Henrickson, Y. Wang, J. Tang, and B.-J. Park, "Google earth elevation data extraction and accuracy assessment for transportation applications," *PLOS ONE*, vol. 12, no. 4, pp. 1–17, Apr. 2017.
- [10] O. Sundstr and L. Guzzella, "A generic dynamic programming matlab function," *2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC)*, pp. 1625–1630, 2009.
- [11] J. Lofberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.