

Real-life implementation of a GPS-based path-following system for an autonomous vehicle

de Winter, Alexander; Baldi, Simone

DOI

[10.3390/s18113940](https://doi.org/10.3390/s18113940)

Publication date

2018

Document Version

Final published version

Published in

Sensors (Basel, Switzerland)

Citation (APA)

de Winter, A., & Baldi, S. (2018). Real-life implementation of a GPS-based path-following system for an autonomous vehicle. *Sensors (Basel, Switzerland)*, 18(11), Article 3940. <https://doi.org/10.3390/s18113940>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright


Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Article

Real-Life Implementation of a GPS-Based Path-Following System for an Autonomous Vehicle

Alexander de Winter and Simone Baldi * 

Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2624 CD Delft, The Netherlands; ajdewinter@live.nl

* Correspondence: s.baldi@tudelft.nl; Tel.: +31-15-2781823

Received: 23 October 2018; Accepted: 13 November 2018; Published: 14 November 2018



Abstract: This work is meant to report on activities at TU Delft on the design and implementation of a path-following system for an autonomous Toyota Prius. The design encompasses: finding the vehicle parameters for the actual vehicle to be used for control design; lateral and longitudinal controllers for steering and acceleration, respectively. The implementation covers the real-time aspects via LabVIEW from National Instruments and the real-life tests. The deployment of the system was enabled by a Spatial Dual Global Positioning System (GPS) system providing more accuracy than the regular GPS. The results discussed in this work represent the first autonomous tests on the Toyota Prius at TU Delft, and we expect the proposed system to be a benchmark against which to test more advanced solutions. The tests show that the system is able to perform in real-time while satisfying comfort and trajectory tracking requirements: in particular, the tracking error was within 16 cm, which is compatible with the 13 cm precision of the Spatial Dual GPS, whereas the longitudinal and lateral acceleration are within comfort levels as defined by available experimental studies.

Keywords: autonomous driving; path-follower; Spatial Dual GPS; real-life tests

1. Introduction

The last decades have seen an increasing amount of research on automated driving. What started as driving assist systems, such as cruise control or a lane departure warning system, expanded rapidly into advanced driving assist systems (ADAS) that are capable of stopping the vehicle in critical situations, keeping the vehicle in its own lane and regulating the vehicle velocity from 0 to 130 km/h with adaptive cruise control (cf. [1–3] among other references).

This work is meant to report on some activities at TU Delft on the design and implementation of a path-following system for an autonomous Toyota Prius: the deployment of the system was enabled by a Spatial Dual GPS system providing more accuracy than the regular GPS. The focus of this work is on the path-follower system: such system receives a trajectory (in terms of desired position and velocity) from a path planner. While extremely relevant, higher-level tasks such as changing the trajectory when the surroundings change (e.g., other vehicles merging in front of the autonomous vehicle, pedestrians crossing the road) are not addressed here. The design encompasses: finding the vehicle parameters for the actual vehicle to be used for control design; lateral controller for steering and longitudinal controller for acceleration. The implementation covers the real-time aspects via LabVIEW from National Instruments and the real-life tests performed at the Valkenburg Naval Air Base. While the goal of the control would be to make the steering and acceleration as human-like as possible, achieving such task would require very advanced strategies such as non-linear model predictive controllers (cf. [4–7] and references therein). Such strategies are well known for being computationally intensive and not straightforward to deploy (due to many design parameters such as control horizon, prediction horizon, terminal constraints, multi-objective weights, etc.). Therefore,

it is worth remarking that the controllers presented here have been designed with the aim of real implementation, with few parameters to be tuned and low requirements from the computational point of view. The focus was on relatively simple solutions rather than on optimality.

The well-known book “Vehicle Dynamics and Control” [8] refers to several control algorithms for path following, all of them using to some extent the information on the curvature of the desired path: the most basic algorithms use the information of the path at the current position [9–11], while more advanced algorithms use a look-ahead distance and calculate the desired steering angle with use of some future location of the vehicle [12,13]. The controller selected for implementation in this work is the Future Predictive Controller of [14]. Even though more advanced strategies exist, e.g., based on optimality arguments [15,16], we found that the strategy in [14] has few design parameters and allows for fast implementation. For the longitudinal acceleration, we used a proportional-derivative strategy, which is standard even in the more advanced strategies for longitudinal control [1–3].

The system is developed in LabVIEW from National Instruments [17]. LabVIEW gives the opportunity to have real-time simulations, and most importantly, it is compatible with the National Instruments hardware that the Toyota Prius is equipped with. A code programmed in LabVIEW can be compiled to a National Instruments device from which it is possible to run the code in real-life systems [18]. The real-life tests show that the system is able to perform in real-time while satisfying comfort and trajectory tracking requirements: in particular, the tracking error was within 15.8 cm (compatible with the Spatial Dual GPS precision, which is of around 13 cm according to the manufacturer) and the lateral/longitudinal acceleration were within comfort levels as defined in well-known empirical tables [19,20]. As the results discussed in this work represent the first autonomous tests on the Toyota Prius at TU Delft, we expect the proposed system to be a benchmark against which to test more advanced solutions in the future.

This paper is organized as follows. Section 2 introduces the vehicle we consider. Section 3 outlines the control design for both lateral and longitudinal dynamics. Section 4 discusses the LabVIEW implementation and validates the model with real-time simulations. Section 5 deals with the real-life experiments and conclusions are drawn in Section 6. Comfort levels of lateral/longitudinal acceleration are briefly recalled in the Appendix.

2. The Vehicle

Within the Dutch Automated Vehicle Initiative (DAVI), TU Delft and other partners are working on the development of high automated vehicles for research and demonstrations on public roads. Among other vehicles, a Toyota Prius is being equipped to such purposes (cf. Figure 1), e.g., with sensors and radar to achieve 360 degrees of sensing. The autonomous driving system in the Prius encompasses a set of modules, as shown in Figure 2: some of the systems are ready to use, whereas others are still under development. Most important to the scope of this work is that the vehicle has been equipped with a system that is capable of controlling the steering, acceleration and brakes of the vehicle. This system is called Move-Box, developed by TNO (Dutch for: Netherlands Organisation for Applied Scientific Research), and serves also as an interface between the PCI eXtensions for Instrumentation (PXI) mounted on the vehicle. An eHorizon system by Continental [21] is capable of placing the vehicle inside a high-definition map with use of GPS. For higher level tasks (not considered in this work), such as path planning tasks, the vehicle is able to verify the GPS location with use of camera images: furthermore, the tracker can recognize objects and track them via the radars and cameras that are mounted on the vehicle.



Figure 1. The Toyota Prius used for this work (the two devices on the roof of the vehicle are part of the Spatial Dual GPS).

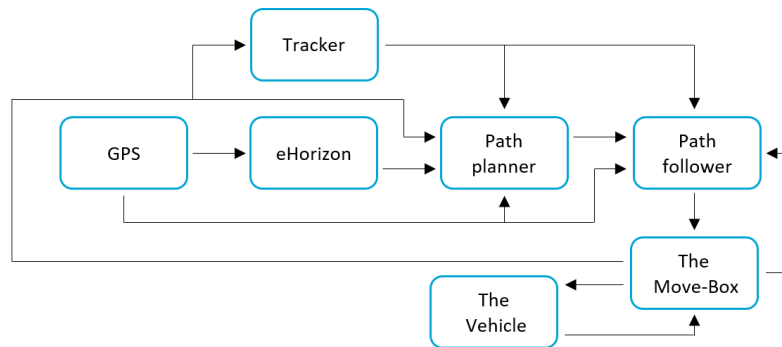


Figure 2. Schematic view of the autonomous vehicle system.

The focus of this work is on the path-follower system: such systems receives a trajectory (in terms of desired position and velocity) from the path planner that the autonomous vehicle must follow. Tasks such as changing the trajectory when the surroundings change (e.g., other vehicles merging in front of the autonomous vehicle, pedestrians crossing the road) are decided at the tracker/path planner level, and will not be addressed here. From the sensing point of view, it is worth mentioning that the GPS installed on the Toyota Prius is an advanced system from a company named Advanced Navigation. The GPS system is called Spatial Dual and uses two antenna mounted on the roof of the vehicle as shown in Figure 1. Basically, Spatial Dual is a ruggedized miniature GPS which is augmented with sensor fusion from an inertial navigation system (INS) and attitude and heading reference system (AHRS). This makes it more precise than regular GPS systems with respect to position, velocity, acceleration and orientation.

The model of the vehicle used for control design is the standard bicycle model [22–24], which can be summarized as follows

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_{\alpha_f} + C_{\alpha_r}}{mv_x} & -v_x + \frac{l_r C_{\alpha_r} - l_f C_{\alpha_f}}{mv_x} \\ \frac{l_r C_{\alpha_r} - l_f C_{\alpha_f}}{I_z v_x} & \frac{-l_r^2 C_{\alpha_r} - l_f^2 C_{\alpha_f}}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha_f}}{m} \\ \frac{l_f C_{\alpha_f}}{I_z} \end{bmatrix} \delta \quad (1)$$

where v_y is the longitudinal velocity, r is the yaw rate, δ is the wheel steering angle, and the other parameters are defined in Table 1. Note that the longitudinal velocity v_x is a time-varying parameter of the bicycle model. The steering ratio k_δ is needed to convert the wheel steering angle into the actual

angle of the steering wheel given by the driver and constrained in $[-\delta_{max}, \delta_{max}]$ (please note that we use the term “wheel steering angle” for the angle of the front wheels, and the term “angle of the steering wheel” for the steering angle of steering wheel, i.e., the rotation of the steering wheel). τ_δ is the time constant of a first-order filter used to model the delay of the steering actuator. In Equation (1), the tire model is taken as a linear model, as typically assumed in the bicycle model: this implies that the model is valid as soon as the vehicle operates at a linear regime (low acceleration/deceleration and smooth cornering) [22,25]. The vehicle parameters in Table 1 were initially unknown and have been found as explained in the following subsection.

Table 1. Acquired vehicle parameters.

Parameter	Value	Symbol	Unit
Length from front axle to CoG	1.0868	l_f	m
Length from rear axle to CoG	1.6132	l_r	m
Vehicle mass	1590	m	kg
Cornering stiffness of the front wheels	22,200	C_{α_f}	N/rad
Cornering stiffness of the rear wheels	22,200	C_{α_r}	N/rad
Moment of inertia around Z axis	800	I_z	kg/m ²
Steering ratio	14.6	k_δ	-
Maximum angle of the steering wheel	7.592	δ_{max}	rad
Dynamic steering time constant	0.2	τ_δ	s

2.1. Vehicle Parameters

To find the distance of the front and rear axles from the Centre of Gravity (CoG), it was first necessary to determine the location of the CoG. To locate its position, it was necessary to measure the weight that rests on the front axle and on the rear axle. Unfortunately, we could not rely on the vehicle manufacturer information, because all the equipment added to the vehicle had likely changed the position of the CoG. Therefore, weighting has been performed on a weightbridge with two persons in the front chairs.

The steering parameters k_δ , δ_{max} and τ_δ have been retrieved from the data-sheets of Toyota. Finally, the values of the cornering stiffness and the moment of inertia are found with use of data fitting. In particular, while driving the vehicle, the angle of the steering wheel, longitudinal velocity, lateral acceleration and yaw rate have been recorded. The last two quantities have been measured via the INS mounted of the vehicle. Then, the missing parameters have been fit on the recorded data with a non-linear least squares method by using the algorithm `lsqnonlin` in Matlab. The vehicle has been driven in such away to have the tires working in a linear regime [26,27], in line with the linear bicycle model in Equation (1). The experiments have been performed at the Valkenburg Naval Air Base, a location that provides the appropriate space for this type of tests.

Several datasets have been used for identification and validation: one such set is shown in Figure 3. The result of the fitting for this dataset can be seen in Figure 4. As the Variance Accounted For (VAF)

$$VAF = \left(1 - \frac{\text{variance}(y - y_{est})}{\text{variance}(y)}\right) \times 100 \quad (2)$$

is 96.36% for the lateral acceleration and 98.41% for the yaw rate, we conclude that the resulting bicycle model can be used for control design purposes, as explained in the next section.

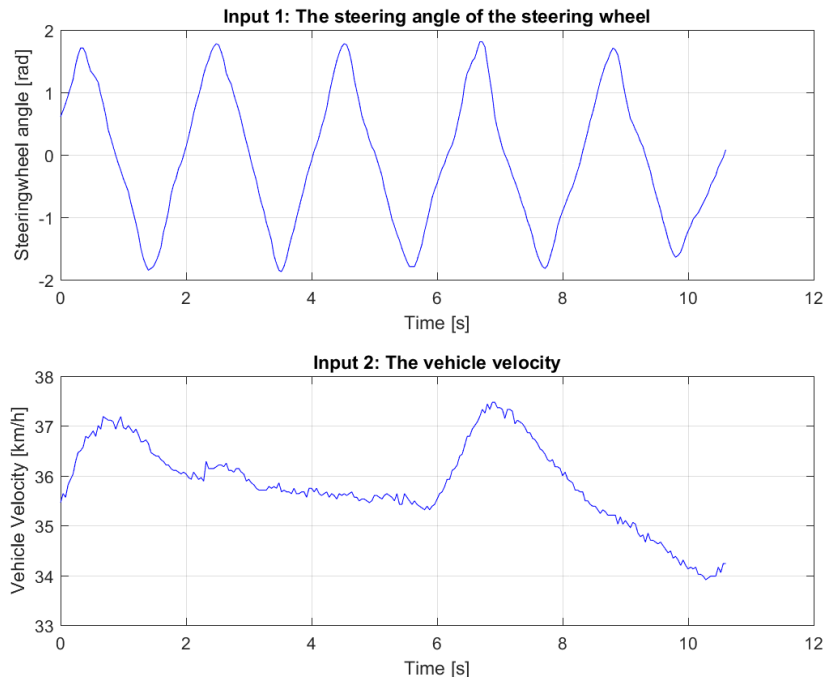


Figure 3. The inputs of one dataset.

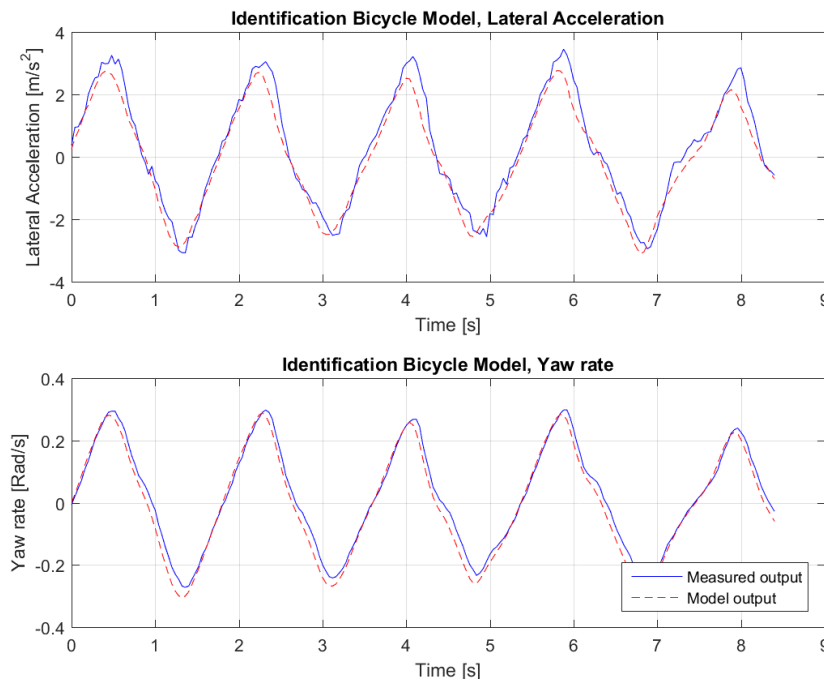


Figure 4. Data fit.

3. Controls

The goal of the controllers is to make the steering and acceleration as human-like as possible. While it is acknowledged that advanced non-linear model predictive control techniques would make this possible, a primary goal was to deliver a simpler solution which could be more easily designed, validated and tested. Most importantly, the controller should be able to run in real-time with low computational requirements.

3.1. Lateral Control

The controller selected among many possible options was the Future Predictive Controller [14]. This control algorithm makes use of a future location to generate the angle of the steering wheel. The controller can be written as follows

$$\begin{aligned}
 \theta_e &= \theta - \theta_p(t) \\
 f_x &= L_f \cos(\theta) + c_x \\
 f_y &= L_f \sin(\theta) + c_y \\
 y_{ef} &= -(f_x - p_x) \sin(\theta) + (f_y - p_y) \cos(\theta) \\
 L_f &= k_f v_x \\
 \delta &= k_h \sin(\theta_e) + \frac{k_s y_{ef}}{v_x}
 \end{aligned} \tag{3}$$

where the first equation represents the heading error, (c_x, c_y) is the current location of the vehicle, (f_x, f_y) is the future location of the vehicle, y_{ef} is the future lateral error, and L_f is the look-ahead distance dependent on v_x [12].

A graphical explanation of the control strategy is illustrated in Figure 5. The tuning of this control algorithm is done with three different parameters, k_f , k_s and k_h . The parameters have been tuned using the bicycle model, in such a way that the tracking error is small while the lateral acceleration is within human comfort limits. The optimization has been performed using patternsearch in Matlab. Clearly, the fact that only three parameters need to be tuned make the design and tuning quite straightforward, whereas the same might not be true for other control strategies like linear quadratic or model predictive control. The following rules of thumb can be derived: by increasing k_s and k_h the steering is more aggressive: the error will in general be smaller, but at the price of high lateral acceleration and sometimes instability. Increasing k_f will result in a larger look-ahead distance: if the distance is too large the vehicle will cut corners, and if this distance is too small the vehicle will oscillate.

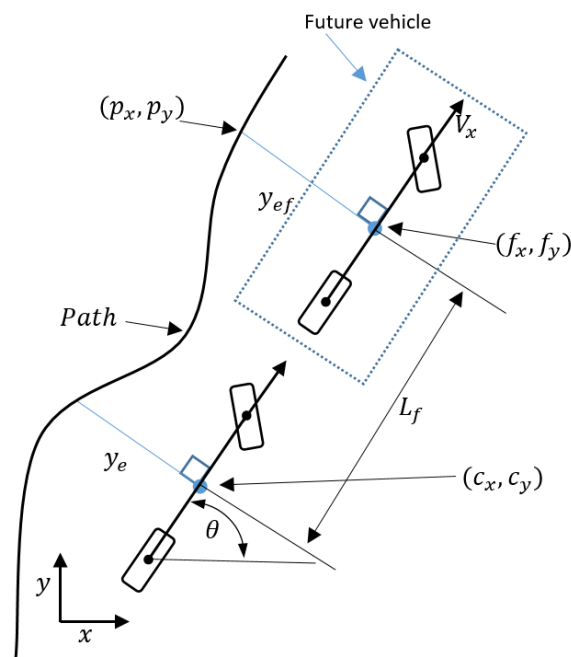


Figure 5. The future vehicle of the future prediction control algorithm.

3.2. Longitudinal Control

The longitudinal control also satisfies the criteria of simplicity of design and tuning. It is a Proportional-Derivative (PD) action that regulates the acceleration of the vehicle [28]

$$v_e(t) = v_c(t) - v_p(t) \quad (4)$$

$$a_c(t) = K_p v_e(t) + K_d \frac{dv_e(t)}{dt} \quad (5)$$

where v_e is the error in velocity, v_c is the vehicles velocity, v_p is the velocity that is requested by the path, and a_c is controlled acceleration. The parameters of this control law are the proportional control gain K_p and the derivative control gain K_d . These parameters have been tuned using the bicycle model, in such a way that the tracking error is small while the longitudinal acceleration is within human comfort limits. The comfort level for humans (both for longitudinal and lateral comfort) has been evaluated from experimental studies [19,20] that define threshold values of comfort: comfortable level, medium comfort level and discomfort level.

The tuning of K_p and K_d has been performed by trial and error: because in total we have only five parameters to tune, it is also possible to perform the tuning of the lateral and longitudinal controllers jointly via patternsearch, even though we have noticed minor differences as compared to a separated tuning. The resulting parameters are shown in Table 2, and their tuning has been performed by interfacing Matlab with LabVIEW as explained in the next section.

Table 2. Control parameters.

	Value
k_f	1.1
k_s	0.7
k_h	1.0
K_p	0.3
K_d	1.18

4. Real-Time Implementation

The program used to construct the system is LabVIEW from National Instruments. LabVIEW gives the opportunity to construct a Virtual Instrument (VI) that can be simulated when the time is synchronized to a timing source. This makes it possible to have real-time simulations. While such capabilities are also possible with other programs, e.g., Matlab, the main advantage of LabVIEW is its compatibility with the National Instruments hardware that the Toyota Prius is equipped with. This means that a code programmed in LabVIEW can easily be compiled to a National Instruments device from which it is possible to run the code in real-life systems.

For example, the Move-Box in the Toyota Prius (cf. Figure 2) is actuated by a PCI eXtension for Instrumentation (PXI) from National Instruments. The implementation of the system to the PXI can then be easily done, and then it is possible to test if the full system is able to run real-time. To generate a path the following conventions are adopted: the positions x and y are in a global coordinate system. The Universal Transverse Mercator coordinate system (UTM) is chosen. The heading angle is in radians, increasing counter clockwise from a polar axis that is drawn horizontal and pointed to the right. The velocity is in meters per second, even though in the plots it will be reported in km/h for better understanding. For the sake of readability, let us skip all the practical issues that must be taken into account when implementing a path follower, namely: the LLtoUTM submodule, which takes care of the conversion from latitude and longitude to UTM coordinates [29]; and the curve equation submodule, which finds the nearest path point and makes a curve fit to a part of the trajectory.

4.1. Simulations

This section presents the simulation tests before going to the real-life tests. Some standard paths have been used for simulating and tuning the controllers. When looking for the best performance, the following cost functions have been considered:

- The Root Mean Square (RMS) of the error distance y_e
- The maximum value of y_e
- The minimum value of y_e
- The maximum lateral acceleration a_y

Tables 3–6 show how the performance changes after changing the different parameters for some representative values of k_f , k_s and k_h (the different simulations are numbered 1–8). The final settings used in the controller are the ones in Simulation 6 (and reported in Table 2). It has to be noted that, due to the noise in the GPS heading error, we found it beneficial to filter the heading error. The benefits of this action can be better seen from the smoother response in Figure 6, where the “delay” refers to a first-order lag element used to model the steering actuator. Note that, in Figure 6, the filter removes the fast oscillations present in the original control action.

Table 3. Evaluation parameters for variation k_s .

		Simulation 1	Simulation 2	Simulation 3
Steering Parameter	k_s	0.7	0.1	1.3
RMS	(m)	0.052	2.044	0.331
Max y_e	(m)	0.097	0.010	0.601
Min y_e	(m)	−0.099	−3.220	−0.001
Max a_y	(m/s ²)	2.282	2.165	2.188

Table 4. Evaluation parameters for variation k_f .

		Simulation 1	Simulation 4	Simulation 5
Look-Ahead Parameter	k_f	1.1	0.1	2.1
RMS	(m)	0.052	0.769	1.633
Max y_e	(m)	0.097	0.002	2.708
Min y_e	(m)	−0.099	−1.184	−0.0001
Max a_y	(m/s ²)	2.282	2.623	3.624

Table 5. Evaluation parameters for evaluation of steering output.

		Simulation 1	Simulation 6
RMS	(m)	0.052	0.052
Max y_e	(m)	0.097	0.104
Min y_e	(m)	−0.099	−0.100
Max a_y	(m/s ²)	2.282	2.357

Table 6. Evaluation parameters for variation L_f .

		Simulation 7	Simulation 8
Feed-forward length L_f	(m)	9.2	12.2
RMS	(m)	0.393	0.285
Max y_e	(m)	0.209	0.559
Min y_e	(m)	−0.797	−0.366
Max a_y	(m/s ²)	2.6566	2.5432

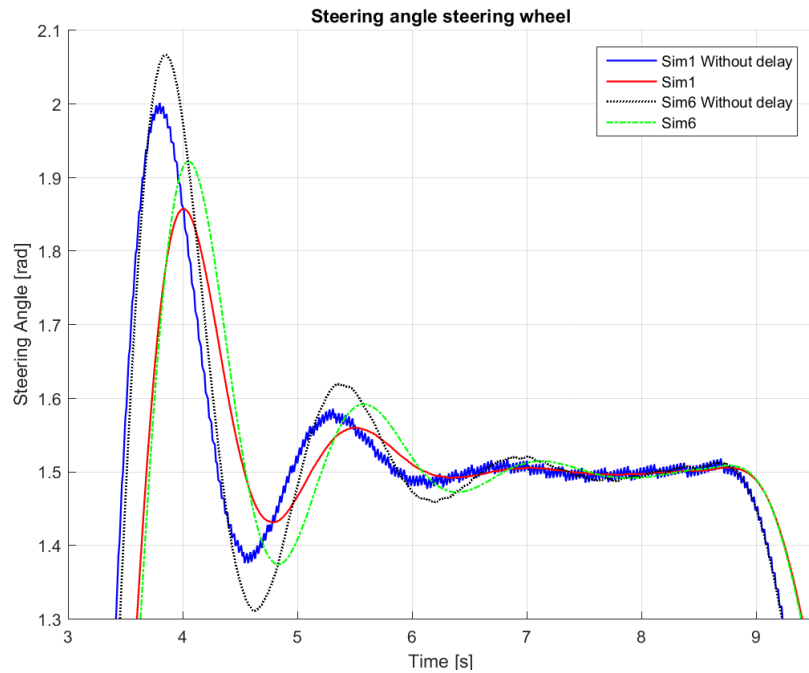


Figure 6. Angle of the steering wheel responses for evaluation of steering output.

While these simulations have been performed at 30 km/h, we also tried simulations at higher velocity (50 km/h): our findings were that the same control parameters did not work well at such velocity, and the hypothesis is that the control gains should be scheduled according to the velocity. In particular, we found that the steering response is highly dependent on the velocity of the vehicle.

5. Real-Life Experiments

The path in Figure 7 was used for real testing. The path was driven at Valkenburg Naval Air Base. Note that, because this path is driven with the actual vehicle, the end and beginning of the path do not coincide; in addition, the corners of the path are not identical and the lines between the corners are not completely straight. The velocity of the path is variable: this is achieved by imposing to the Move-Box appropriate acceleration and deceleration. The resulting velocities are approximately of 15 km/h at straight lines and 10 km/h at cornering. The velocity profile can be seen in Figure 8.

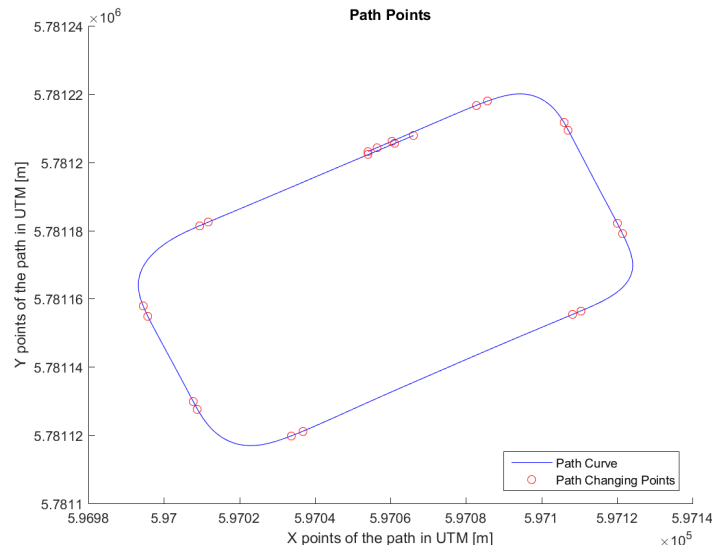


Figure 7. Trajectory for the test drive.

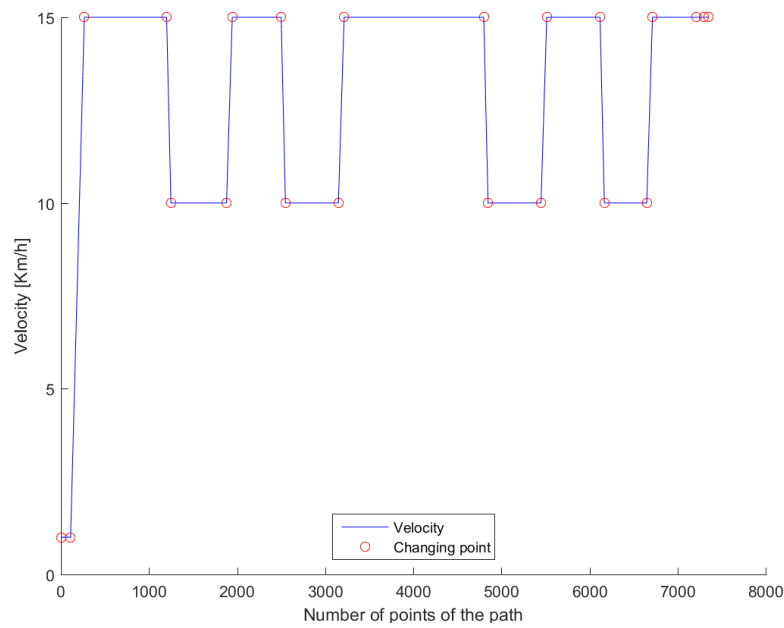


Figure 8. Velocity profile for the test drive.

For comparison purposes, we simulated the same path on the LabVIEW system, so that we are able to compare the real-life behavior with the simulated ones. Figures 9–13 show that such behaviors are quite close. The actual distance error turns out to be smaller than in simulations, whereas the actual velocity tracking error turns out to be bigger. This is mainly due to actuator dynamics (power train dynamics) that cannot be modeled in the bicycle model.

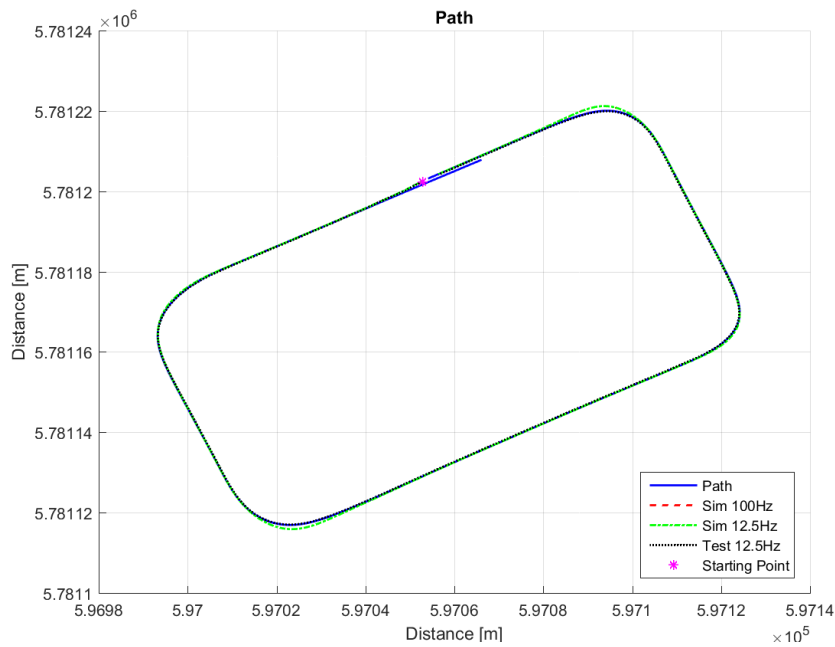


Figure 9. The path, simulation responses and the response from the test drive.

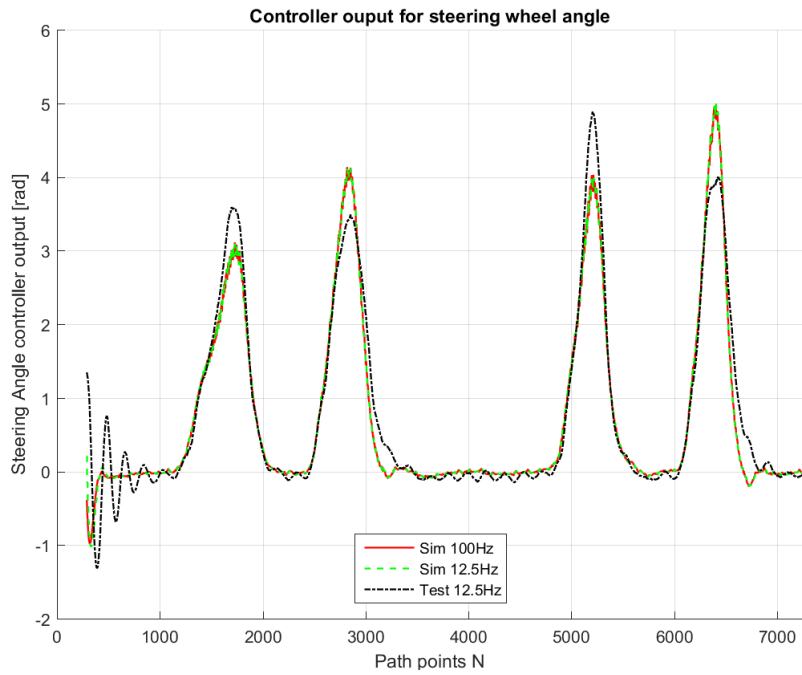


Figure 10. Controller output for simulations and test drive.

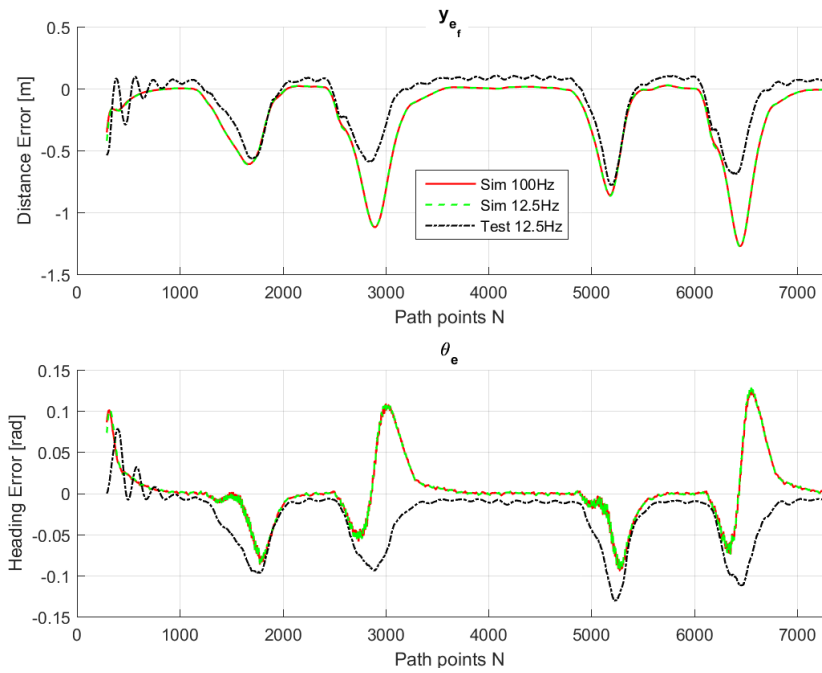


Figure 11. Lateral control inputs for simulations and test drive.

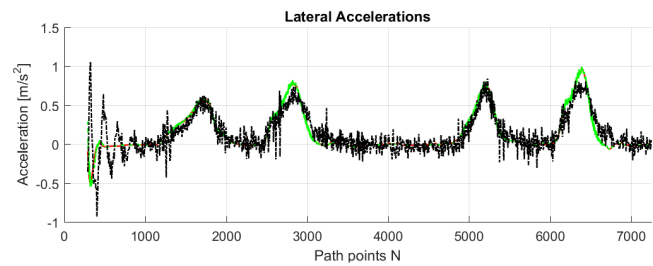


Figure 12. Lateral acceleration for simulations and test drive.

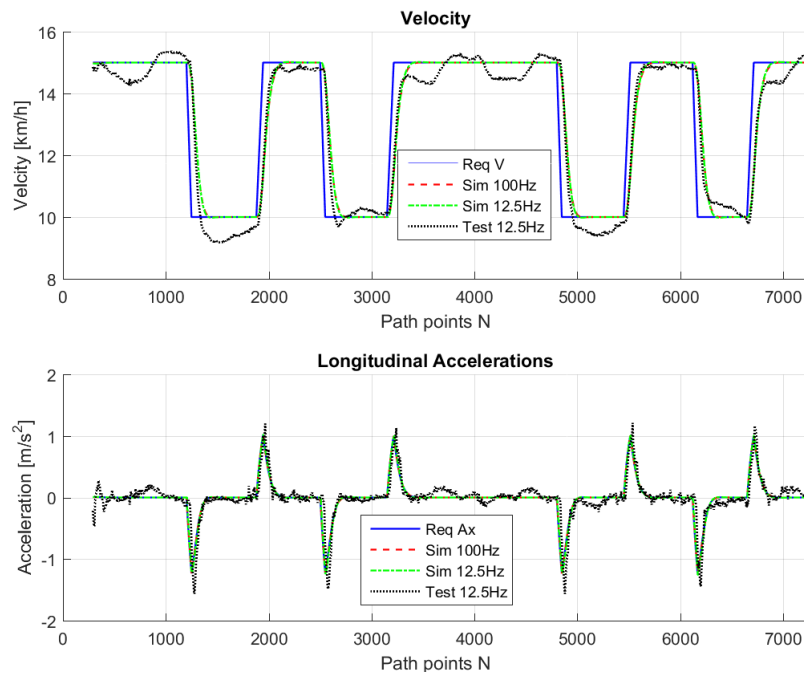


Figure 13. Longitudinal control output and resulting velocity for simulations and test drive.

The comparison between simulations and real-test shown in Table 7 reveal that the various errors are of a similar order of magnitude. It is worth mentioning that tests at different frequencies had to be performed, since the system on the actual vehicle had to work at 12.5 Hz. It can be noted that the maximum error is of around 16 cm, which is consistent with the accuracy of the Spatial Dual GPS on the Toyota Prius which is 13 cm. The minimum error is of the order of 1 m, which is because the vehicle smooths the corners. In addition, the accelerations are always within the expected comfort limits. For completeness, the comfort limits are reported in the Appendix.

Table 7. Evaluation parameters for difference between simulation and real test.

Test		Simulation 100 Hz	Simulation 12.5 Hz	Real Test 12.5 Hz
RMS	(m)	0.332	0.333	0.122
Max y_e	(m)	0.113	0.114	0.158
Min y_e	(m)	-1.193	-1.195	-0.654
Max a_y	(m/s^2)	0.981	1.014	1.063

5.1. Supplementary Materials

Videos of other actual tests can be found at [30,31]. The two tests are performed at a maximum velocity of 20 km/h and 30 km/h, respectively. In these tests the velocities are higher than the 15 km/h reported in the previous section. These tests are meant to show the effectiveness of the path-following even at higher speeds. From the videos, it is possible to see the prediction used to generate the angle of the steering wheel. The steering wheel moves quite smoothly, in a human-like manner. In addition, it can be seen that the current velocity smoothly follows the desired velocity (which is pre-selected by the path planner), and, furthermore, no harsh acceleration is registered. As these results represent the first autonomous tests on the Toyota Prius at TU Delft, we expect the proposed system to be a benchmark against which to test more advanced solutions in the future.

6. Conclusions and Future Work

This paper reports some recent activities at TU Delft on the design and implementation of a path-following system for an autonomous Toyota Prius. The design encompassed: finding the vehicle

parameters for the actual vehicle to be used for control design, and lateral controller for steering and longitudinal controller for acceleration. The implementation covered the real-time aspects via LabVIEW from National Instruments and the real-life tests. It is worth remarking that the controllers have been designed with the aim of straightforward implementation, with low requirements from the computational point of view. Therefore, the focus was on relatively simple solutions rather than on optimality. As the results discussed in this work represent the first autonomous tests on the Toyota Prius at TU Delft, we expect the proposed system to be a benchmark against which to test more advanced solutions. Future work will cover: a more advanced vehicle model (which requires measuring more parameters of the vehicle, especially during nonlinear regime); more advanced controllers (compatibly with the computational requirements of the equipment); and further development of all other autonomous driving modules. Finally, before it will be possible to do more research on vehicle control, especially during extreme manoeuvres, it will be required to have a better understanding of the Move-Box system.

Author Contributions: Individual contributions: conceptualization, A.d.W.; methodology, A.d.W.; software, A.d.W.; validation, A.d.W.; formal analysis, A.d.W.; investigation, A.d.W. and S.B.; resources, A.d.W. and S.B.; data curation, A.d.W.; writing—original draft preparation, S.B.; writing—review and editing, S.B.; visualization, A.d.W.; supervision, S.B.; project administration, S.B.; and funding acquisition, S.B.

Acknowledgments: The authors gratefully acknowledge the support received by some researchers involved in the Dutch Automated Vehicle Initiative (DAVI): in particular, Tom Dalhuisen for the help during the measurements and the test drive, Daniel den Hartog for the tests to acquire the vehicle parameters, and Riender Happee for allowing to perform the tests on the real vehicle.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Acceleration for Human Comfort

The comfort level for humans in terms of lateral acceleration have been quantified in an experimental study [20]. Different levels have been found, as shown in Table A1.

Table A1. Lateral acceleration thresholds of comfort.

Lateral Acceleration (m/s^2)	Consequence
$0 < a_y \leq 1.8$	Comfort level
$1.8 < a_y \leq 3.6$	Medium comfort level
$3.6 < a_y \leq 5$	Discomfort level
$5 < a_y$	Uncomfortable

The maximum and minimum longitudinal accelerations depend on the vehicle's velocity and have been determined with use of human driving data [19]. These acceleration values are shown in Figure A1.

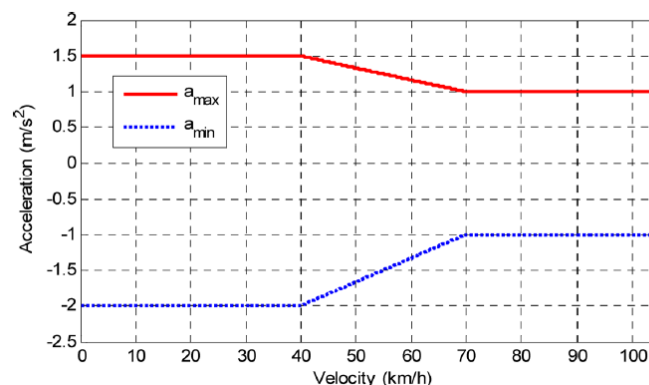


Figure A1. Velocity-dependent range of longitudinal accelerations.

References

1. Reschka, A.; Böhmer, J.R.; Saust, F.; Lichte, B.; Maurer, M. Safe, dynamic and comfortable longitudinal control for an autonomous vehicle. *IEEE Intell. Veh. Symp. Proc.* **2012**, 346–351. [[CrossRef](#)]
2. Attia, R.; Orjuela, R.; Basset, M. Combined longitudinal and lateral control for automated vehicle guidance. *Veh. Syst. Dyn.* **2014**, *52*, 261–279. [[CrossRef](#)]
3. Nilsson, J.; Brannstrom, M.; Fredriksson, J.; Coelingh, E. Longitudinal and Lateral Control for Automated Yielding Maneuvers. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1404–1414. [[CrossRef](#)]
4. Raffo, G.V.; Gomes, G.K.; Normey-Rico, J.E.; Kelber, C.R.; Becker, L.B. A predictive controller for autonomous vehicle path tracking. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 92–102. [[CrossRef](#)]
5. Solea, R.; Nunes, U. Trajectory planning with velocity planner for fully-automated passenger vehicles. In Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference, Toronto, ON, Canada, 17–20 September 2006; pp. 474–480.
6. Shakouri, P.; Ordys, A. Nonlinear Model Predictive Control approach in design of Adaptive Cruise Control with automated switching to cruise control. *Control Eng. Pract.* **2014**, *26*, 160–177. [[CrossRef](#)]
7. Arrigoni, S.; Cheli, F.; Manazza, S.; Gottardis, P.; Happee, R.; Arat, M.; Kotiadis, D. Autonomous vehicle controlled by safety path planner with collision risk estimation coupled with a non-linear MPC. In Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD 2015), Graz, Austria, 17–21 August 2015; CRC Press: Boca Raton, FL, USA, 2016; p. 199.
8. Rajamani, R. *Vehicle Dynamics and Control*; Springer Science & Business Media: New York, NY, USA, 2011.
9. Coulter, R.C. *Implementation of the Pure Pursuit Path Tracking Algorithm*; Technical Report; DTIC Document: Los Angeles, CA, USA, 1992.
10. De Luca, A.; Oriolo, G.; Samson, C. Feedback control of a nonholonomic car-like robot. In *Robot Motion Planning and Control*; Springer: London, UK, 1998; pp. 171–253.
11. Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; et al. Stanley: The robot that won the DARPA Grand Challenge. *J. Field Robot.* **2006**, *23*, 661–692. [[CrossRef](#)]
12. Snider, J.M. *Automatic Steering Methods for Autonomous Automobile Path Tracking*; Technical Report CMU-RITR-09-08; Robotics Institute: Pittsburgh, PA, USA, 2009.
13. Idriz, A.F.; Rachman, A.S.A.; Baldi, S. Integration of auto-steering with adaptive cruise control for improved cornering behaviour. *IET Intell. Transp. Syst.* **2017**, *11*, 667–675. [[CrossRef](#)]
14. Zakaria, M.A.; Zamzuri, H.; Mamat, R.; Mazlan, S.A. A path tracking algorithm using future prediction control with spike detection for an autonomous vehicle robot. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 309. [[CrossRef](#)]
15. Sharp, R.; Valtetsiotis, V. Optimal preview car steering control. *Veh. Syst. Dyn.* **2001**, *35*, 101–117.
16. Sharp, R.S. Driver steering control and a new perspective on car handling qualities. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2005**, *219*, 1041–1051. [[CrossRef](#)]
17. National Instruments Corporations. National Instruments. Available online: <http://www.ni.com/en-us/shop/labview.html> (accessed on 15 August 2018).
18. Rachman, A.S.A.; Idriz, A.F.; Li, S.; Baldi, S. Real-time Performance and Safety Validation of an Integrated Vehicle Dynamic Control Strategy. *IFAC-PapersOnLine* **2017**, *50*, 13854–13859. [[CrossRef](#)]
19. Moon, S.; Yi, K. Human driving data-based design of a vehicle adaptive cruise control algorithm. *Veh. Syst. Dyn.* **2008**, *46*, 661–690. [[CrossRef](#)]
20. Xu, J.; Yang, K.; Shao, Y.; Lu, G. An experimental study on lateral acceleration of cars in different environments in Sichuan, Southwest China. *Discret. Dyn. Nat. Soc.* **2015**, *2015*, 494130. [[CrossRef](#)]
21. Continental. Dynamic eHorizon. Available online: http://www.continental-automotive.com/www/automotive_de_en/themes/commercial_vehicles/ch_interior_en/ehorizon_en/03_dynamic_ehorizon_en.html (accessed on 15 August 2018).
22. Awan, M. Compensation of Low Performance Steering System Using Torque Vectoring. Ph.D. Thesis, Cranfield University, Bedfordshire, UK, 2012.
23. Isermann, R. Diagnosis methods for electronic controlled vehicles. *Veh. Syst. Dyn.* **2001**, *36*, 77–117. [[CrossRef](#)]

24. Gillespie, T.D. *Fundamentals of Vehicle Dynamics*; Society of Automotive Engineers (SAE): Warrendale, PA, USA, 1992.
25. Jagga, D.; Lv, M.; Baldi, S. Hybrid Adaptive Chassis Control for Vehicle Lateral Stability in the Presence of Uncertainty. In Proceedings of the 2018 26th Mediterranean Conference on Control and Automation (MED), Zadar, Croatia, 19–22 June 2018; pp. 1–6.
26. Pacejka, H.B.; Bakker, E.; Nyborg, L. Tyre modelling for use in vehicle dynamics studies. *SAE Trans.* **1987**, *870421*, 190–204.
27. Dugoff, H.; Segel, L.; Fancher, P. An Analysis of Tire Traction Properties and Their Influence on Vehicle Dynamic Performance. In Proceedings of the 1970 International Automobile Safety Conference Compendium, Detroit, MI, USA, 13–15 May 1970; pp. 341–366. [[CrossRef](#)]
28. Moon, S.; Moon, I.; Yi, K. Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance. *Control Eng. Pract.* **2009**, *17*, 442–455. [[CrossRef](#)]
29. Robert. GPS Position to UTM Coordinates VI. Available online: <https://forums.ni.com/t5/Community-Documents/GPS-position-to-UTM-coordinates/ta-p/3536992> (accessed on 15 August 2018).
30. De Winter, A.; Baldi, S. Path Following System Test on Real-Vehicle (Max Velocity 20 km/h). Available online: <https://www.youtube.com/watch?v=V0prb1wem14&feature=youtu.be> (accessed on 15 August 2018).
31. De Winter, A.; Baldi, S. Path Following System Test on Real-Vehicle (Max Velocity 30 km/h). Available online: <https://www.youtube.com/watch?v=XIRnLPQo6nw&feature=youtu.be> (accessed on 15 August 2018).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).