Proc. of World Cong. on Multimedia and Computer Science

Design and Implementation of a Path Tracking Steering Controller for EO Smart Connecting Car

Mohammed Ahmed[1]and Mehmed Yu¨ksel[2]
[1] German Research Center for Artificial Intelligence (DFKI) - Robotics Innovation Center, Bremen, Germany
[1] mohammed.ahmed@dfki.de, [2]mehmed.yueksel@dfki.de

Abstract: **Within this paper, we reconsider the path-following problem with the objective to synthesize a control law which allows the prototype electric vehicle EO smart connecting car 2 to autonomously follow a desired path in a stable manner. Furthermore, the software architecture controlling the car and all its subsystems is depicted. A hardware-in-the-loop (HIL) frame work that can provide an effective platform for developing and testing different subsystems of the vehicle in real-time is described and the general workflow is outlined. The performance of the designed controller is discussed and demonstrated with realistic simulations. An experiment with the HIL framework interfaced to the car is presented, conclusions and an outlook of future work are also given.**

Keywords: **Electric Vehicle, Path Following, Automatic Steering, Hardware-in-the-loop, Software-in-the-loop, nonlinear systems, real-time control**

I. INTRODUCTION

EO smart connecting car 2 (EO2) is a two seated intelligent electric vehicle which is the second generation of EO car [1][2]. EO2 is a four wheel-driving electric vehiclewith extended manoeuvrability through its innovative suspension/axle design, optimized wheel hub motors, and high urbancapability thanks to its foldable body and high adaptivity with a coupling mechanism for Car2Car or Car2Extender modules.EO2 is developed to use drive assistance and autonomous driving systems.

Fundamental to the design of an Ackerman steered autonomous driving system is the development of a low-level controller that effectively performs trajectory or path tracking. In the field of autonomous vehicle guidance, navigation and control, path-following problem of car-type vehicles is of particular interest. Though ample literature is available on various methods for controlling ground vehicles with various techniques and strategies for this problem [3][4][5][6][7][8], little information is presented on the implementation and tuning of such controllers.

Recent trends in automotive industry point in the directionof increased content of electronics, computers, and controls with emphasis on the improved functionality and overall system robustness. For this reason and to enhance the design, implementation, and test of such components and subsystems, rapid control prototyping (RCP) tools are used for car development and optimization [9][10]. The RCP real-time units are used as interface for electro-mechanical subsystems of these vehicles for different application fields, such as engine

[1]EO means ”I go” in Latin, however, the true meaning is more ”to there”, in the sense of being on the way to a goal

DOI: 02.WCMCS.2013.1.4

control units (ECU) and vehicle control units (VCU)[11].They perform different functions like x-by wire (steer-brake throttle)as well as data logging, and interfacing for different applications (sensor integrations, gateway or bypassing for data buses etc.) [12]. Because of these benefits, RCP units are integrated into autonomous vehicles as low-level vehicle controllers and as subsidiary for autonomy (high-level) controllers in test development platforms [13][7][14].
This work presents an approach to design and implement a proportional input-Scaling feedback controller for a vehicle to follow a desired path, using its forward velocity and angular acceleration as control inputs. For this controller, a kinematic vehicle model is used to map from the path curvature specified by the given path to the vehicle’s actual steering angle. This controller is then tested in a co simulation framework that utilizes a dynamic model to predict the vehicle’s response to a series of different steering inputs. Successful trajectory control results for the ISO3888-2 double lane change test track and the implemented RCP environment and work flow for development, test and verification of the controller and various other subsystems are presented.

II. PATH FOLLOWING MODULE

This module implements how the vehicle tracks a reference path via the control of its actuators. The module is being implemented based on simple and effective control strategies. Our approach consists of applying a control signal based on state feedback control. The implementation supposes that one is able to measure the variables involved in the control loop (typically the position and orientation of the vehicle with respect to either a fixed frame or a path that the vehicle should follow).

A. Vehicle Model

The vehicle kinematic model approximates the mobility of a car [15]. The configuration of non holonomic vehicle [6] is represented by the position and orientation of its main in the plane, and by the angle of the steering wheels.

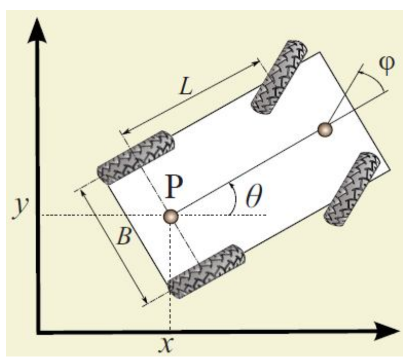

Figure 1. Configuration variables of the vehicle kinematic model

Two velocity inputs are available for motion control. This situation covers in a realistic way our car. The non holonomic nature of the car is related to the assumption that the car wheels roll without slipping. The kinematic model used (shown in Fig.1)is given as:

$$\begin{aligned} \dot{x} &= u_1 \cos\theta \\ \dot{y} &= u_1 \sin\theta \\ \dot{\theta} &= \frac{u_1}{L}\tan\varphi \\ \dot{\varphi} &= u_2 \end{aligned} \tag{1}$$

where *(x,y)* represents the coordinates of the point *P* locatedat mid-distance of the actuated wheels, the angleθ characterizes the vehicle chassis orientation,φrepresents the vehicle steering wheel angle, and *L* is the distance between the rear and front wheels axles. In this equationu_1, $u_2$and represents the driving and the steering velocity input, respectively.

B. Path Reference Frames

The analysis is limited to the case of a vehicle workspace free of obstacles. In fact, it is implicitly considered that the controller is to be embedded in a hierarchical architecture in which a higher–level planner solves the obstacle avoidance problem and provides a series of motion goals to the lower control layer. In this perspective, the controller deals with the basic issue of converting ideal plans into actual motion execution. Wherever appropriate, the interactions between feedback control and motion planning primitives shall be highlighted, such as the generation of open-loop commands and the availability of a feasible smooth path joining the current vehicle position to the destination.

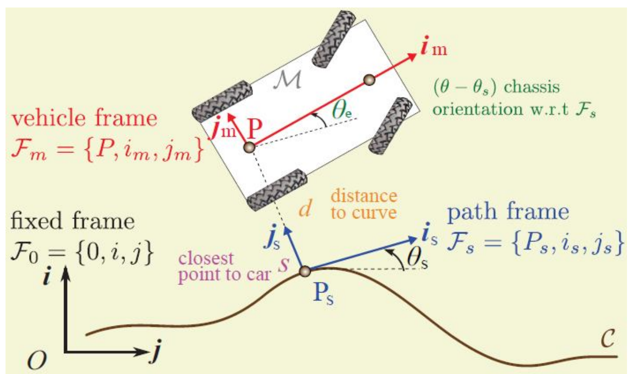

Figure 2. Path-following geometric reference frames

Let us consider a curve Cin the plane of motion (Fig.2).

- s is the curvilinear abscissa at P_s by projecting $\boldsymbol{P}$ orthogonally on C. This point exists and isunique if the point $P$is close enough to $\boldsymbol{C}$.
- $d$is the ordinate of Pin $\boldsymbol{Fs}$; its absolute value is also the distance between $\boldsymbol{P}$ and $\boldsymbol{C}$.
- $\theta_e = \theta - \theta_s$is the angle characterizing the orientation of the car chassis w.r.t $\boldsymbol{F_s}$.

The equations of motion of $\boldsymbol{P}$ w.r.t $\boldsymbol{C}$ can be expressed based on the three variables s, d, and θ_e as follows:

$$\begin{aligned} \dot{s} &= \frac{u_1}{1 - dc(s)} \cos\theta_e \\ \dot{d} &= u_1 \sin\theta_e \\ \dot{\theta}_e &= \frac{u_1}{L} \tan\varphi - \dot{s}c(s) \\ \dot{\varphi} &= u_2 \end{aligned} \tag{2}$$

The model (2) is transformed to the chained form [16] through the change of coordinates and control variables $(s, d, \theta_e, u1, u2) \mapsto (z_1, z_2, z_3, v_1, v_2)$ defined by:

$$
\begin{aligned}
& z_1 = s, \quad z_2 = d, \\
& z_3 = [1 - dc(\mathrm{s})] \tan\theta_e, \\
& z_4 = -c(\mathrm{s})[1 - dc(\mathrm{s})]\left(1 + 2\tan^2\theta_e\right) \\
& \quad - d\frac{\partial c(\mathrm{s})}{\partial s}\tan\theta_e + [1 - dc(\mathrm{s})]^2 \frac{\tan\varphi}{L\cos^3\theta_e}; \\
& v_1 = \dot{z}_1, \quad v_2 = \dot{z}_4
\end{aligned}
\tag{3}
$$

The (2, 4) single-chain form (3), although nonlinear, has a strong underlying linear structure. This clearly appears when $u_1$is assigned as a function of time, and is no longer regarded as a control variable. In this case, (3) becomes a single-input, time-varying linear system.

C. Path Following with Orientation Control

The control law should asymptotically stabilize $(d = 0,\ \theta_e = 0)$ and also ensures that the constraint on the distanced to the path (i.e.,$|dc(s)|<1$) is satisfied along the trajectories of the controlled system. A proportional input-Scaling controller is designed to provide convergence when the car starts sufficiently close to the desired trajectory. Consider the control law:

$$
\begin{aligned}
v_2 &= -v_1 \sum_{i=2}^{4} \mathrm{sgn}(v_1) k_i z_i \\
&= -v_1\left[\mathrm{sgn}^3(v_1)k_2 z_2 + \mathrm{sgn}^2(v_1)k_3 z_3 + \mathrm{sgn}(v_1)k_4 z_4\right] \\
&= |v_1| k_2 z_2 - v_1 k_3 z_3 - |v_1| k_4 z_4
\end{aligned}
\tag{4}
$$

Where k_2 , k_3 , $k_4$denote the controller parameters.

The constraint$|dc(s)| <1$ is satisfied with the initial stability criteria:

$$
z_3^2(0) + \frac{1}{k_3 - k_2 / k_4} z_4^2(0) < \frac{1}{c_{\max}^2}
\tag{5}
$$

where $c_{\max} = \max_s |c(\mathrm{s})|$.

III. TEST PLATFORM AND RCP WORKFLOWDESCRIPTION

The test platform SujeeCar2 (shown in Fig. 4) is built in actual size to enable the implementation and integration of the vehicle subsystems (mechanical, electrical, software design, sensor and autonomy) and the test of their integration and planned features as early as possible. SujeeCar is constructed with T-slot aluminum profile, which houses the electronic and mechanical components. It consists mainly of two identically constructed axles, which will be used on the target vehicle, four actuators (48 V–485 W) for the steer and lift function of two suspensions, four modified wheel hub motors (52V–2.5 kW, peak 4 kW). For each motor, a BLDC motor controller (60 V–125 A, peak 250 A). The platform is powered with a quick replaceable main battery (LiFePO4 52 V– 100Ah), other electric equipment/parts, main wiring harness and a hydraulic brake system. The platform is controlled with MABX-II 1401/1511/1512 RCP unit.

A. RCP Workflow Description

For system design and easy iteration process in development phases of the control system, RCP methodology is used. Our suggested workflow (shown in Fig. 3) is as follows: The low-level control algorithms for major subsystems (steering and lifting actuator control,steer/drive/brake-by-wire, communication, and user interfaces) and the kinematic models (to calculate steering actuator positions and lifting and wheel speeds based on drive mode and car length) are developed in MATLAB/Simulink environment. The developed subsystem is then compiled and transferred to the RCP unit (blue Simulink model in Fig. 4) and tested in a

---

[2] The test platform Sujee Car is named after Sujeef Shanmugalingam, who drafted its initial chassis design.

hardware in the loop (HIL) platform with the SujeeCar. The communication components of the system are constructed according to the requirements and specifications of the car components as self-contained subsystem (orange blocks in Fig. 4). This has the advantage of flexibility to adapt these components to the different temporary test or permanently used devices and their communication protocols (e.g. CAN or CAN open).

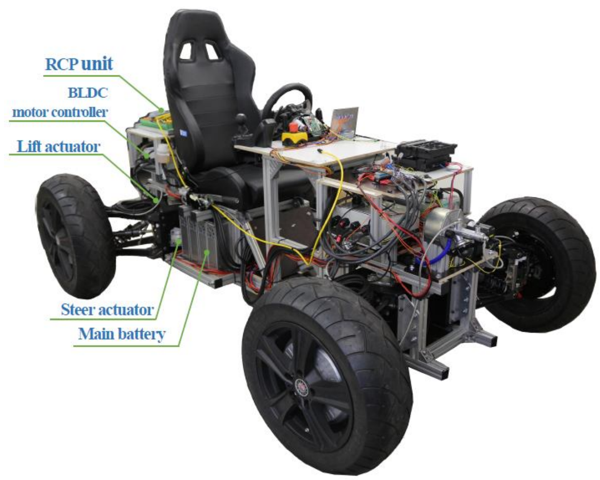

Figure 3. The SujeeCar test platform of EO2 with its suspension and control elements

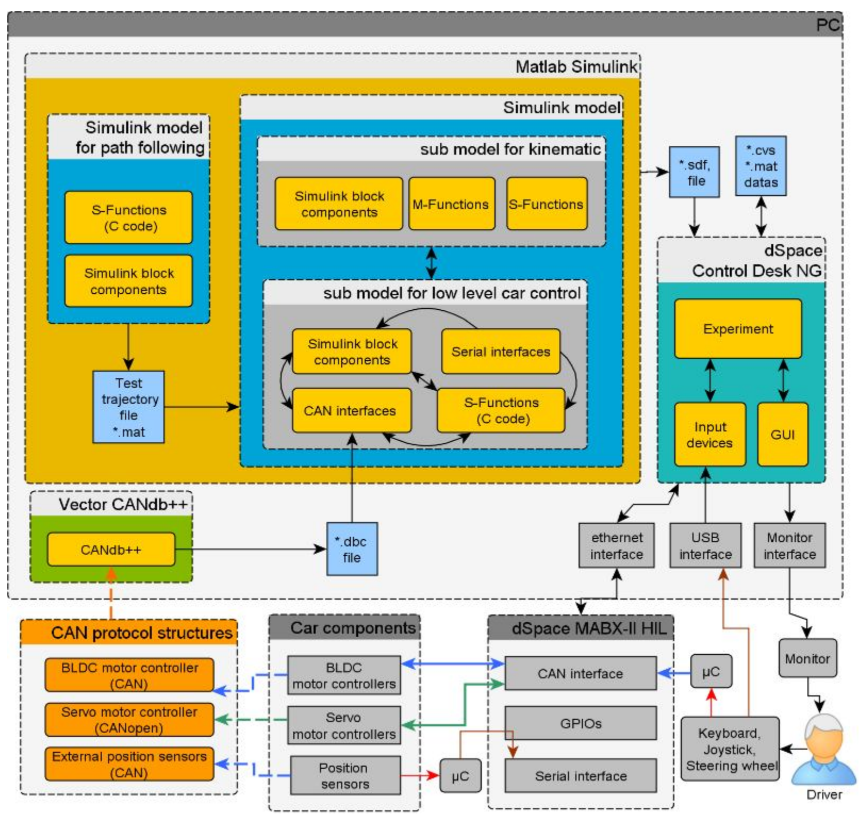

Figure 4. Suggested workflow of the RCP application, gray blocks: hardware components and physical interfaces (car components, RCP unit, Driver Laptop, input and output devices, blue arrows: CAN communications, green arrows: CANopen, red arrows:raw analogue/digital signals, brown arrows: serial/USB communications, dashed arrows: user inputs.

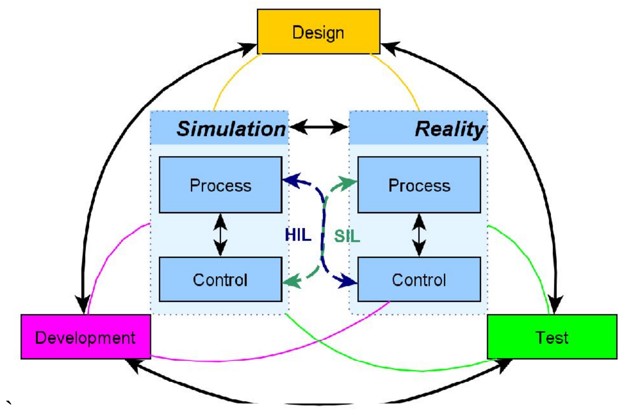

Figure 5. Designed model used in components and subsystems development

B. Designed Model Description

During the development process, it is very beneficial to test the designed components directly and rapidly on the target hardware. Therefore the developing cycle would get shorter and the iteration loops would be shorter leading toless complexity and hence less costs. For this, we adopted the design model given in Fig. 5 in which a planer structure that uses HIL development environment (MATLAB/Simulink–Control Desk with RCP unit as a hardware interface) [17].Consequently, the different phases of the development (design-development-test) would be in crossover or parallel accessible.

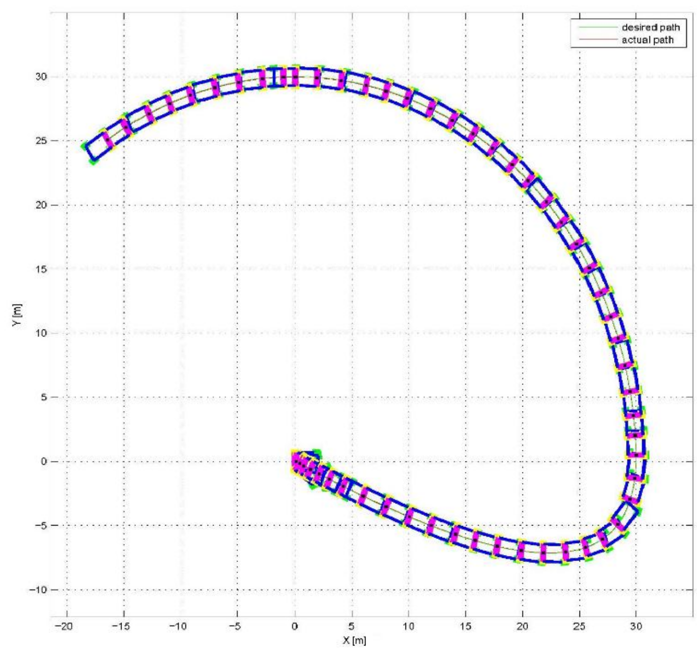

Figure 6. Path following controller and car kinematic model simulation results for an arbitrary desired path

IV. EXPERIMENTS AND RESULTS

The designed controller is implemented as a MATLAB-Simulink C++ S-Function. To test the controller, the implemented MATLAB/Simulink model is connected to the car kinematic model (implemented also in MATLAB-Simulink).In this test, EO2 kinematic parameters are: car length L = 1.9 m, distance between wheels $B = 1.35$ m, and wheel radius r = 0.325 m. The controller parameters were set to $K_2 = 1.0$, $K_3 = 3.0$, $K_4 = 30.0$ (tuned with simple try and error), sampling time $T_s = 0.001$s, and the car is commanded to move with constant forward speed of $u_1 = 0.5$ m/s. An arbitrary desired path is used. The results of this simulation are given in Fig. 6. For a more precise simulation and to be as close as possible to reality, the implemented MATLAB/Simulink controller model is interfaced to a more detailed and dynamic car model in Adams/View. The real car Ackermann steering and inverse kinematics modules are also used in a software-in-the-loop fashion (shown in Fig. 7) within the cosimulation framework [18]. The car model in Adams/ View takes mechanical constraints and dynamic effects into consideration.

The controller is tested with the same car and controller parameters but the car is commanded to move with constant forward speed of $u_1 = 2.8$ m/s. A desired path for the standard test track described in ISO3888-2 [19] is used (depicted in Fig. 9). The results of the simulation with the desired testpath are shown in Fig. 9. Fig.8). The results of the simulation with the closely. The convergence can also be seen in the error graph shown inFig.10, where the initial conditions are also visible. It is clear from these figures that the controller does not attain extremely large values, and is bounded. This is an essential property for real systems.

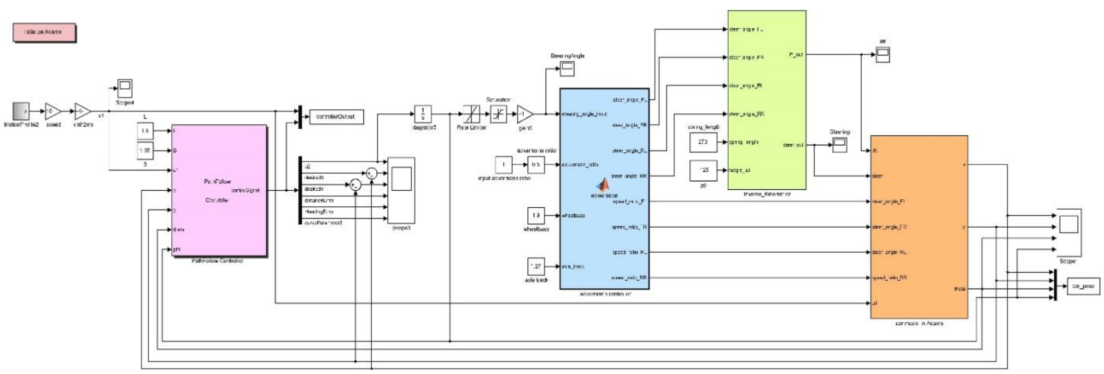

Figure 7. Block diagrams of the path following controller in MATLAB/Simulink with interface to car model in Adams/View

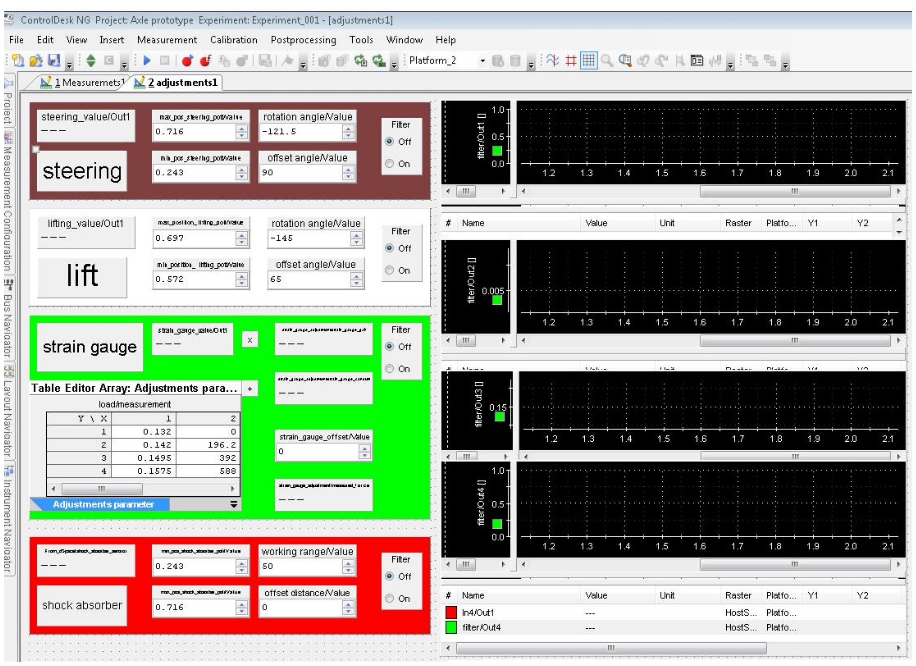

Figure 8. RCP experiment GUI for actuator control and logging

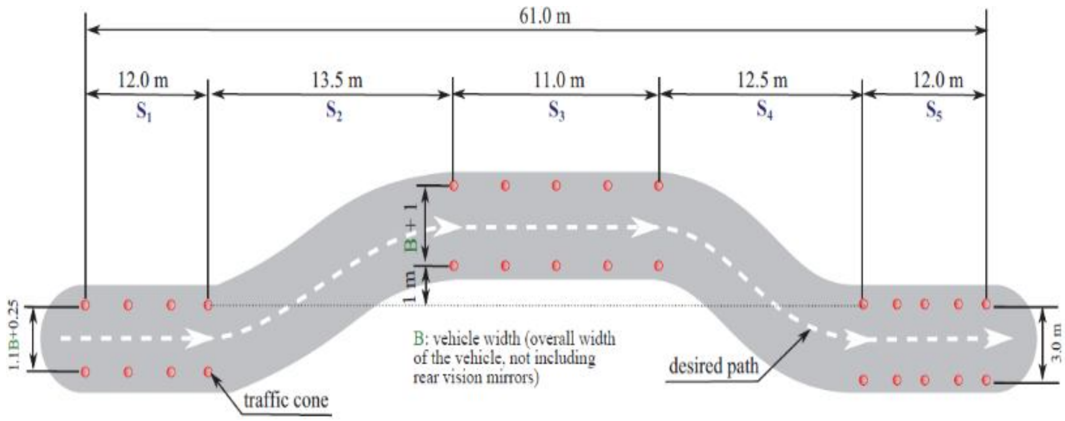

Figure 9. ISO3888-2 double lane change test track

A. Design of HIL Experiments

Because of the complexity of EO2, it is very important toanalyze the dynamics and characteristics of the controlledsystems before actual testing of the designed controller. Theimplemented HIL experiments that can provide an effective platform for developing and testing real-time control systems comprises of different subsystem models. These subsystems are:a dSPACE control system with its ControlDesk environment and MABX-II RCP unit to interface with the test platform, and a human-machine interface for user inputs, parameter setting/inspection and data logging (one of the implemented GUIs is shown in Fig. 8).

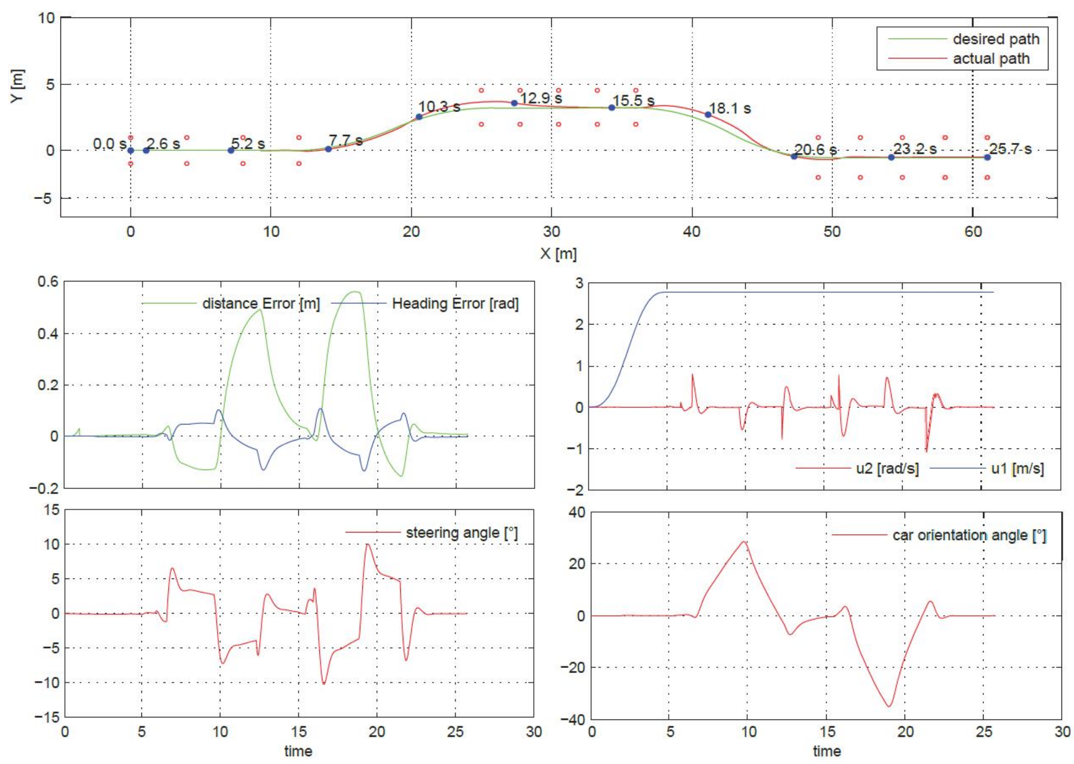

Figure 10. Path following controller in Adams/View – MATLAB/Simulink cosimulation results for ISO3888-2 double lane change test track

As a first test, the developed system is used to analyze the behavior of the steering actuators commanded with the desired path generated from the previously described cosimulation for ISO3888-2 double lane change test. In this experiment, Hall sensor encoders are used on each wheel to measure its steering angle (see in Fig. 3). The desired path commands are recorded in alookup table and directly applied in real time through the experiment model GUI. In addition, the actual steer wheel joint actual positions for the four wheels are depicted and recorded on GUI during the experiment. The results of this experiment are shown inFig.11.From these results, it can be seen that the desired steering commands are applicable to the mechanical system and the actual behavior is identical to that of the simulation. These results will be verified in the next experiments.

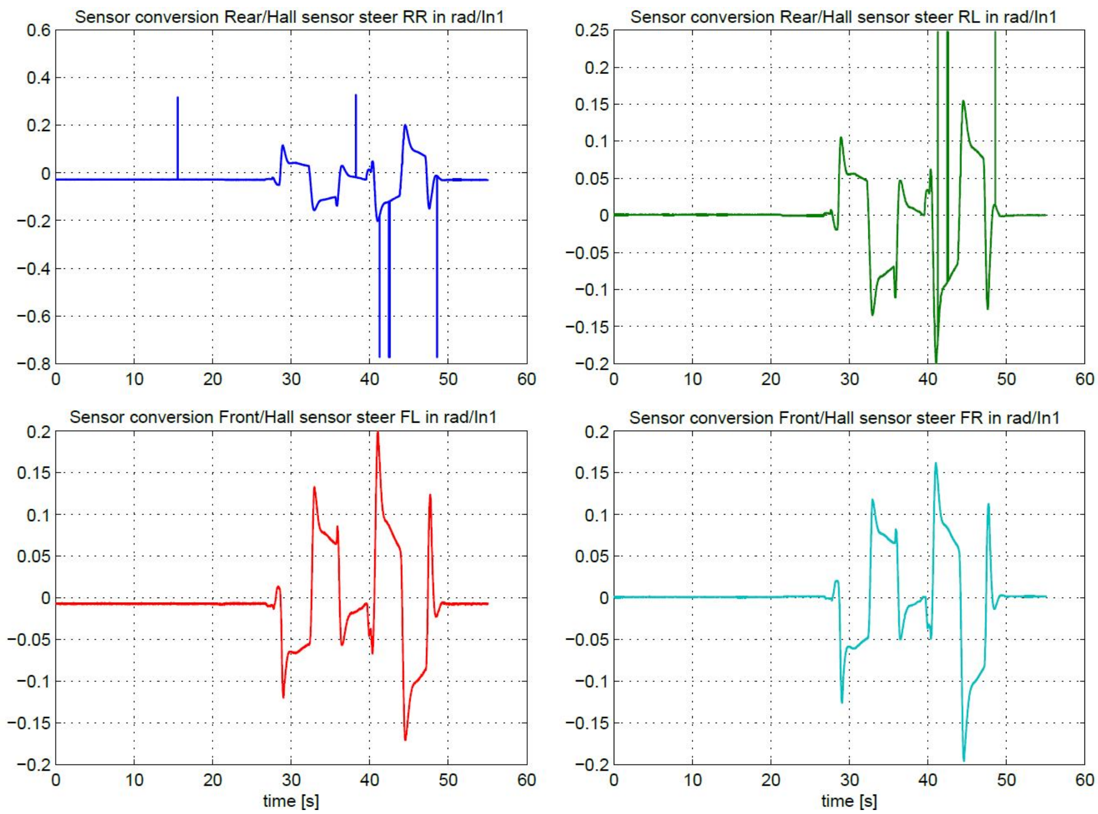

Figure 11. Results of the HIL experiment to test the desired path results of cosimulation for ISO3888-2 double lane change.

V. CONCLUSION

A proportional input-Scaling controller is designed and implemented to autonomously steer the vehicle to follow the desired path in a stable manner. Through realistic simulations results, it is shown that the controller is robust and converges giving good performance provided that the car starts sufficiently close to the desired path. The controller model has the advantages that it can be directly used for differential robots with a trailer and it can be generalized to robot with N trailers in a straightforward manner.

An RCP use-case/application for real time HIL as a test platform to validate hardware and software components and systems of the car is also presented. The system structure, design and development steps (workflow) are described. The first test results and experience is given. Experimental results have verified that this platform is indispensable, effective and adaptive solution as a development and test environment withthe hardware platform. It reduces time and costs significantlyfor implementing component prototypes of the target system.Because of the benefits of the RCP development environment and it capabilities as a realtime hardware interface,this development environment is selected for the integration phase of car as well as the development and optimization of mostsoftware components for the car control. The development status will be tested on the SujeeCar test platform, which is a technology demonstrator for EO2. The control integration of wheel hub motors, steer and lift actuators will be tested for different drive modes and autonomous driving and steering. In addition, the technical characteristics of the car will be measured within this environment.

ACKNOWLEDGMENT

We thank all team members of the EO smart connecting car project [20] (in alphabetical order: Benjamin Girault, Christian Öckermann, David Grünwald, Janosch Machowinski, Roman Szczuka, SujeefShanmugalingam, Sven Kroffke, TimoBirnschein, Yong-Ho Yoo) who provided valuable comments, ideas, and assistance, which were essential to this study.

This work is developed for the EO car which is evolved in the subproject "Innovative Technologies Electromobility (ITEM)" of main project "Model Region Electric Mobility (PMC)"–Module2: "Intelligent IntegrationofElectric Mobility"and funded by the German Federal Ministry of Transport, Buildingand Urban Development(Grant Nr. 03ME0400G).Program coordination is carried out by the National Organization Hydrogen and Fuel Cell Technology (NOW GmbH).

REFERENCES

[1] M. Jahn, M. Schröer, Y.-H. Yoo, M. Yüksel, and F. Kirchner, "Concept of actuation and control for the EO smart connecting car," in Intelligent Robotics and Applications, 2012, pp. 1028–1033.

[2] T. Birnschein, F. Kirchner, B. Girault, M. Yüksel, and J. Machowinski, "An innovative, comprehensive concept for energy efficient electric mobility –EO smart connecting car," in Energy Conference and Exhibition (ENERGYCON), 2012 IEEE International, 2012, pp. 1028–1033.

[3] M. Egerstedt, X. Hu, H. Rehbinder, and A. Stotsky, "Path planning and robust tracking for a car-like robot," in Proceedings of the 5th SympoREFERENCES 29 sium on Intelligent Robotic Systems, 1997, pp.237–243.

[4] P. Morin and C. Samson, "Control of nonlinear chained systems: from the routh-hurwitz stability criterion to time-varying exponential stabilizers," Automatic Control, IEEE Transactions on, vol. 45, no. 1, pp. 141–146, 2000.

[5] J. Baltes and Y. Lin, "Path-tracking control of non-holonomic car-like robot with reinforcement learning," in Working notes of the IJCAI'99Third International Workshop on Robocup.IJCAI Press, 2000, pp.162–173.

[6] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: design, implementation, and experimental validation," Control Systems Technology, IEEE Transactions on, vol. 10, no. 6, pp. 835–852, 2002.

[7] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli, "Autonomous vehicles control in the vislab intercontinental autonomous challenge," in Annual Reviews in Control, vol. 36.Elsevier, September 2011, pp.161–171.

[8] L. Güvenc¸, I. Uygan, K. Kahraman, R. Karaahmetoglu, I. Altay, M. Sent¨urk, M. Emirler, A. Karci, B. Güvenc¸, E. Altug, M. Turan,O. Tas, E. Bozkurt, U. Özgüner, K. Redmill, A. Kurt, and B. Efendioglu,"Cooperative adaptive cruise control implementation of team mekarat the grand cooperative driving challenge," Intelligent TransportationSystems, IEEE Transactions on, vol. 13, no. 3, pp. 1062–1074, 2012.

[9] H. Hanselmann, "Automotive control: From concept to experiment toproduct," IEEE International Symposium on Computer Aided ControlSystem Design (CACSD), pp. 129–134, September 1996.

[10] T. Maxwell, S. Patil, K.and Bayne, and R. Gale, "Hardware-in-the-looptesting of gm two-mode hybrid electric vehicle two-mode hybrid electricvehicle," in 2010 IEEE 12th Workshop on Control and Modeling forPower Electronics (COMPEL). IEEE, 2010, pp. 1–5.

[11] dSPACE, "Wellcome to future! microautobox ii: Flexible, powerful, andready for future applications," dSPACE Magazine, vol. 3, pp. 36–39,2010.

[12] S. Mathewy, "Rapid control prototyping of radar-based active safetyapplications using dspacemicroautobox," in dSPACE User Conference2012. Delphi Technical Center, India, September 2012.

[13] V. Sezer, C. Dikilita, Z. Ercan, H. Heceoglu, A. Oner, A. Apak,M.Gokasan, and A. Mugan, "Conversion of a conventional electricautomobile into an unmanned ground vehicle (ugv)," 2011 IEEEInternationalConference on Mechatronics (ICM), pp. 564–569, April2011.

[14] R. Werner, "Robotik im rosenbeet,"dSPACE Magazin, vol. 1, pp. 33–39, 2013. [Online].Available: http://www.dspace.com/shared/data/pdf/2013/33 39 robotik.pdf

[15] B. Siciliano and O. Khatib, Eds., Handbook of Robotics. Springer,2008.

[16] A. Pesterev and L. Rapoport, "Canonical representation of the pathfollowing problem for wheeled robots," Automation and RemoteControl, vol. 74, no. 5, pp. 785–801, 2013. [Online]. Available:http://dx.doi.org/10.1134/S0005117913050044

[17] D. Abel and A. Bollig, Eds., Rapid control prototyping. Springer, 2006.

[18] M. Ahmed, Y.-H. Yoo, and F. Kirchner, "A co-simulation frameworkfor design, test and parameter optimization of robotic systems,"in ISR / ROBOTIK 2010. The joint conference of the 41stInternational Symposium on Robotics and the 6th German Conferenceon Robotics (ISR/ROBOTIK-2010), June 7-9, Munich, Germany, 2010.
[19] K. Lundahl, J. Aslund, and L. Nielsen, "Vehicle dynamics platform,experiments, and modeling aiming at critical maneuver handling,"Linköping University, Vehicular Systems, Tech. Rep. 3064, 2013.
[20] ITEM–Project Web Page," http://robotik.dfki-bremen.de/en/research/projects/item.html"