UDC 004.031.2

B. I. Tymchenko, V. V. Samodelok, D.V. Putilina,
O. M. Galchonkov, PhD.

THE ROBUST CONTROL SYSTEM FOR SKID ELIMINATION IN DYNAMIC ROAD ENVIRONMENTS

Abstract. *This article presents the theoretical development and practical implementation of the algorithm to ensure stability in the case of yaw speed maneuvers. Yaw stability control systems provide a control action, which prevents the vehicle from under- or oversteering in a handling maneuver (e.g. lane change, slalom, etc.).*

Many new vehicle features (like Electronic Stability Programs (ESP), indirect Tire Pressure Monitoring Systems (TPMS), road-tire friction monitoring systems, and so forth) rely on models of the underlying vehicle dynamics. The so-called bicycle vehicle model is a rather simple model structure that used frequently in the vehicle dynamics literature. The algorithm was developed based on this dynamic model.

To prove the algorithm stability, was built the simulator using Unreal Engine and the hardware prototype was developed. Results of simulation and real-world performance were measured and compared, so they validate the approach with low error.

Keywords: *robotic vehicles, high-speed maneuvers, safety, intelligent car, car-like robot, emulator, Unreal Engine, hardware solution, embedded system, dynamic model, stability, traffic safety*

Б. И. Тимченко, В. В. Самоделок, Д. В. Путилина,
О. Н. Галченков, канд. техн. наук

РОБАСТНАЯ СИСТЕМА УПРАВЛЕНИЯ ДЛЯ ПРЕДОТВРАЩЕНИЯ ЗАНОСА В ДИНАМИЧЕСКИХ ДОРОЖНЫХ УСЛОВИЯХ

Аннотация. *В представленной статье, основанной на исследованиях о повышении стабильности при манёврах на скорости, детально рассмотрена процедура проверки и отбора конкретного программно-аппаратного решения при помощи построения эмулятора динамики авто. Наше решение применимо к концептам автоматических машин так как стабильность в высокоскоростных маневрах является критичной в дорожной обстановке и в безопасности движения в целом.*

Ключевые слова: *робокар, высокоскоростные маневры, безопасность, интеллектуальные машины, эмулятор, Unreal Engine, аппаратное решение, встроенная система, модель динамики, стабильность*

Б. І. Тимченко, В. В. Самоделок, Д. В. Путіліна,
О. М. Галчонков, канд. техн. наук

РОБАСТНА СИСТЕМА КЕРУВАННЯ ДЛЯ ЗАПОБІГАННЯ ЗАНОСУ В ДИНАМІЧНИХ ДОРОЖНІХ УМОВАХ

Анотація. *У представленій науковій статті, заснованій на дослідженнях про підвищення стабільності при маневрах на швидкості, була детально розглянута процедура перевірки і відбору конкретного програмно - апаратного рішення за допомогою побудови емулятора динаміки авто. Наше рішення може бути застосовано до концептів автоматичних машин бо стабільність при високошвидкісних маневрах є критичним фактором, який впливає на безпеку дорожнього руху в цілому.*

Ключові слова: *робокар високошвидкісні маневри, безпека, інтелектуальні машини, емулятор, Unreal Engine, апаратне рішення, вбудована система, модель динаміки, стабільність*

Introduction

Modern computer systems make it possible to solve numerous problems. Especially, among all areas, stands interest of modern science to the control algorithms and robotics.

At present moment, begins the era of robotic vehicles, which would be able to follow the designated route autonomously. Vehicle stability at high-speed maneuvers is a critical factor that affects traffic safety in general. This article presents the theoretical development and practical implementation of the algorithm to ensure stability in the case of yaw speed maneuvers. Yaw stability control systems have been established in the automotive industry as a safety/performance/comfort feature. They generally provide a control action which prevents the vehicle from under- or oversteering in a handling maneuver (e.g. lane change, slalom, etc.), particularly on a low-friction-coefficient surface.

The Aim

Nowadays, there are a many concepts of automatically driven vehicles, such as Google cars, Mercedes approaches, Tesla Autopilot, etc. The first step to design such complex systems is the prototype. In order to do this kind of work, we choose the Roborace Intelligent Car racing. This championship involves building of car-like robot models (agents) that are running on the track made of side borders. At the same time there are from four to seven cars on the track, so the environment can be treated as dynamic with high level of confidence. Other cars on the track add situations, where the agent needs to perform high-speed cornering in order to overtake the concurrent agent. Therefore, for this set of conditions we need to develop the robust control system. Informally, a controller designed for a particular set of parameters is said to be robust if it also works well under a different set of assumptions.

Compared to real road conditions, we consider less slippery surface due to nearly perfect race conditions. In addition, model vehicles have smaller mass to tire surface ratio and relatively low speed compared to the full-size automobiles. These factors allow us to exploit simpler model and reduce computation and execution time, which is important while the model is executed at microprocessor with limited computational resources.

Related works

The problem of stabilization of an automated car-like vehicle has been treated in many ways in the literature. Chee and Tomizuka have studied the computation of comfortable maneuvers, based on acceleration and jerk constraints [1]. The emergency maneuver issue has been addressed in the literature, too. In 1994, Smith and Starkey determined emergency maneuvers by optimizing the gains of a linear controller using the step response of a nonlinear vehicle model [2]. After that, Shiller and Sundar found a solution for emergency lane-change maneuvers by the use of a clearance curve [3].

Many new vehicle features (like Electronic Stability Programs (ESP), indirect Tire Pressure Monitoring Systems (TPMS), road-tire friction monitoring systems, and so forth) rely on models of the underlying vehicle dynamics. The so-called bicycle vehicle model is a rather simple model structure that used frequently in the vehicle dynamics literature. With this model structure estimate the longitudinal and the lateral stiffness of a tire is possible. Erik Narby originally carried out the actual modeling work in his MSc work at NIRA Dynamics AB, Sweden [4].

Model

The following model (Fig. 1) can describe using the Newton's law of motion and geometric constraints for vehicle. We use the rigid body assumption in order to simplify the model with relatively small error.

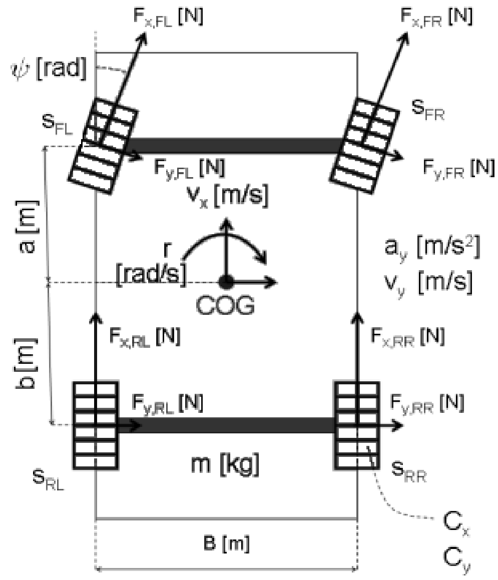


Fig. 1. Top view of vehicle

We can describe the model using the following differential equations for parameters $v_x(t), v_y(t), r(t)$ where:

$v_{x(t)}$ is a lateral velocity;

$v_{y(t)}$ is a longitudinal velocity;

$r(t)$ is a yaw rate of the vehicle, measured around the center of gravity:

$$\begin{aligned}\frac{dv_x(t)}{dt} &= v_y(t)\cdot r(t) \\ &+\frac{1}{m}\cdot\left(F_x^{FL}(t)+F_x^{FR}(t)\right)\cdot\cos(\psi(t)) \\ &-F_y^{FL(t)}+F_y^{FR}(t))\cdot\sin(\psi)) \\ &+F_x^{RL}(t)+\ F_x^{RR}(t).\end{aligned} \quad (1)$$

$$\begin{aligned}\frac{dv_y(t)}{dt} &= -v_x(t)\cdot r(t) \\ &+\frac{1}{m}\cdot(F_x^{FL}(t)+F_x^{FR}(t)\cdot\sin(\psi(t)) \\ &+F_y^{FL}(t)+F_y^{FR}(t)\cdot\cos(\psi(t))\ ) \\ &+F_y^{RL}(t)+F_y^{RR}(t).\end{aligned} \quad (2)$$

$$\frac{dr(t)}{dt}=\frac{1}{J}\cdot(a\cdot((F_x^{FL}(t)+F_x^{FR}(t))\cdot\sin(\psi(t)) \\ +(F_y^{FL}(t)+F_y^{FR}(t))\cdot\cos(\psi(t))) \\ -b\cdot(F_y^{RL}(t)+F_y^{RR}(t))). \quad (3)$$

Where the subscript x is for forces, that act in longitudinal direction and the subscript y is for lateral forces. The superscripts *FR, FL, RR, RL* are indexes for tires: front right, front left, rear right, etc. $\psi(t)$ is the steering angle, J is a moment of inertia, and a and b the distances from the center of gravity (COG) to the front and rear axles.

From Newtonian dynamics, we can derive forces F_{ij} , where *i, j* is tire indexes [5]:

$$F_{xi}(t)=C_x\cdot s_i(t);F_{yi}(t)=C_y\cdot\alpha_i(t). \quad (4)$$

Here *Cx* and *Cy* are the friction coefficients in longitudinal and lateral directions respectively. They are chosen different in order to deal with different form of tire in these directions and to include the rolling friction of wheel.

We assume the tire parameters are the same for all tires, so we can derive the equations for slip angles:

$$\alpha_{Fj}(t)=\psi(t)-atan2((v_y(t)+a\cdot r(t)),v_x(t)), \\ \alpha_{Rj}(t)=-atan2((v_y(t)-b\cdot r(t)),v_x(t)). \quad (5)$$

Here, the index j is to denote the left and right tires.

We consider the full differential model with limited slip on rear wheels in order to simplify the model. This assumption allows us to eliminate rear wheels longitudinal slip $\alpha_r=0$

Implementation

To successfully participate in the competition, we built the four-wheel-drive model of the vehicle. It is equipped with standard brushed direct current (DC) motor with mechanical planetary differential to transfer torque to wheels.

Direct current motor is equipped with electromagnetic incremental encoder for speed sensing. It is needed to implement the closed-loop speed control system, which emulates the natural process of accelerating/braking in real conditions.

To know angular speed of vehicle and its linear acceleration, we added the 6-DOF integrated measurement unit with accelerometer and gyroscope on a single chip [6].

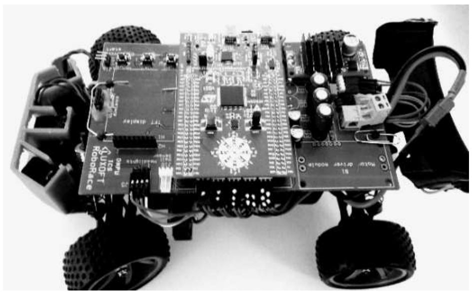


Fig. 2. Hardware vehicle

Path sensing unit consists of five low-range infrared proximity sensors. They are mounted in the front part of vehicle and directed to see the obstacles at the front and at sides of the car.

We choose the ARM Cortex-M4 microprocessor because there are many floating-point operations. Integrated floating-point unit (FPU) takes care of that for us.

Model parameter estimation

Based on the available hardware sensing units, we estimated the basic parameters of the mechanical vehicle.

The main critical coefficient we needed to estimate in runtime is the characteristic of tire-ground soil – the lateral stiffness coefficient.

From the mechanical vehicle we can obtain the following parameters:

$a_y(t)$ – lateral acceleration of the vehicle;

$\omega_z(t)$ – yaw rate measured from gyroscope;

$v(t)$ – vehicle linear speed;

$\psi(t)$ – angle of steering wheels;

From the model, we can derive the coefficients *Cx* and *Cy.* To simplify the model, let us make assumption, that the vehicle tires do not show any slip so *Cx* can be nullified. This assumption is possible because of good track coverage and robust speed controller in the vehicle [7].

Solving the equation (2) with appropriate substitutions for *Cy*, we can derive following [8]:

$$C_y=\frac{(a_y(t)+v_x(t)\cdot\omega_z(t))\cdot m}{\dfrac{2\cdot(v_y(t)-b\cdot\omega_z(t))}{v_x(t)} \\ -2\cdot\cos(\psi(t))\cdot\dfrac{\psi(t)-v_y(t)+a\cdot\omega_z(t)}{v_x(t)}}. \quad (6)$$

We can approximately calculate $v_x(t)$ and $v_y(t)$ using the following equations [9]:

$$v_x(t) = v(t)\cdot\sin(\omega(t)),$$
$$v_y(t) = v(t)\cdot\cos(\omega(t)). \quad (7)$$

From these equations and data samples from sensors, we found the coefficient *Cy* in runtime for different road conditions. These calculations based on real time data are used in the model and in automatic driver algorithm.

We tested the hardware vehicle on the surface with bumps to estimate *Cy* coefficient. Results of testing are presented on the Fig. 3. Outliers on the figure correspond to bumps on the surface.

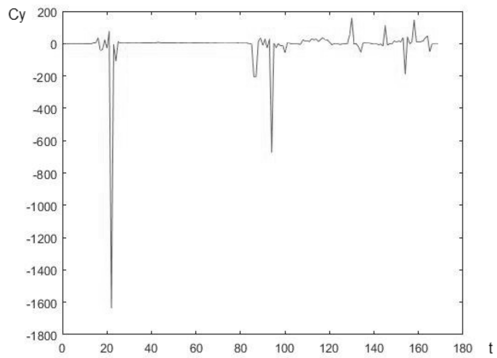


Fig. 3. Estimation of *Cy* coefficient

In order to deal with uncertainties of the terrain, we decided to use Hampel filtering [10] for the coefficient. It is also suitable because we need as low lag as possible between actual and filtered measurements, so we cannot use filters such as sliding mean, MAD, etc. Results for filter with window size 5 is shown on the Fig. 4. Enlarging window size leads to slower response and false detection of actual terrain friction glitches as bumps.

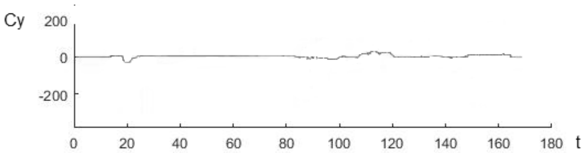


Fig. 4. Filtered *Cy*

Control system

Path sensing unit gives information about obstacles in following directions (Fig. 5)

Maximal ray length for sensors is 150 cm.

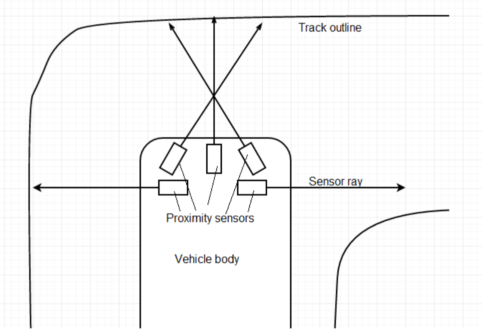


Fig. 5. Path sensing unit information

According to the value of the distance sensor, we built in coordinate plane polygon with zero in the COG of the car. Then we find the most distant vertex and select two neighboring vertexes relative to the most distant. The midpoint between them is the needed direction. The angle between the line to this point and the vertical axis is the angle of wheel turn. Obviously, it will be the chord of circle with no obstacles on the path.

To deal with slippery surface we use the approximating predicting controller.

Deriving from (2) with several approximations, we obtained the formula of lateral acceleration. This formula predicts the lateral acceleration in the next frame based on measured acceleration in the current frame:

$$a_y(t+1) = \frac{1}{m}\cdot\Big(\big(F_{FR}(t)+F_{FL}(t)\big)\cdot\sin(\psi(t))\Big)$$
$$+2\cdot C_y(t)\cdot\big(\psi(t)-v_y(t)+a\cdot\omega_z(t)\big)/v_x(t))\cdot\cos(\psi(t)))$$
$$+\big(2\cdot C_y(t)\cdot\big(b\cdot\omega_z(t)-v_y(t)/v_x(t)\big)\big). \quad (6)$$

Where F_{FR} and F_{FL} are denoted as following:

$$F_{Fi} = \frac{m\cdot a_{yi}(t)}{\cos(\psi(t))} \quad (8).$$

If the acceleration is sufficient to overcome the force of friction of the road surface [11], acting on a tire, we need to use the following sequence:

1. Obtain angle from path sensing unit.
2. Calculate the minimal viable acceleration $a_y(t+1)$, in order not to skid on the surface.
3. Based on it, calculate the minimal angle that will not cause skidding.
4. If the acceleration is still high, we solve (2) on speed to slow down without skidding.

5. If the speed cannot be lowered, we apply digressive braking to stop anyways.

This algorithm showed excellent results in simulation and good result in real conditions.

Simulation

To prove the reliability of the model, we built the emulator of dynamic vehicle using one of the popular game engines. We have chosen the Unreal Engine game framework with respect to its perfect physics simulation. In addition, there is a model of vehicle with the same type of suspension as in the real vehicle available for free download.

Using the estimated parameters from the hardware, all algorithms and measurements were proved in simulation before being executed on the physical mechanism. That saved a plenty of time and allowed us not to waste time to solve the problems associated with hardware and mechanic parts of the vehicle. Usage of the simulator, in addition, allowed us to test multiple algorithm branches to select the best-fit solution for real conditions. Main view of the simulator is shown on the Fig. 6. Small window on the left is the sight of track by robot.

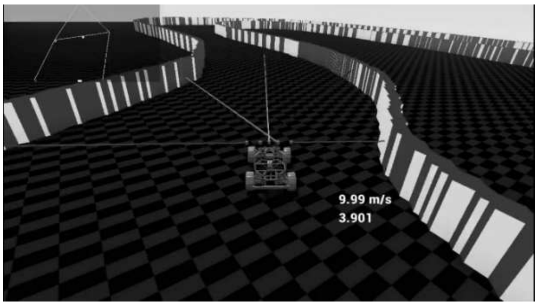


Fig. 5. View of the simulator workbench

We can adjust all the parameters of the simulated vehicle, in order to make modelling as close to reality as possible. In addition, racetracks in simulation are built from pixelated border map, so we can model any track from reality and, even the fully automatic generated tracks.

To compare performance in simulation with reality, we measured the steering angle $\psi(t)$ and lateral acceleration $a_y(t)$. Simulated track was built according to parameters of the real track. Differences between simulation and the real data are shown on Fig. 6 and Fig. 7.

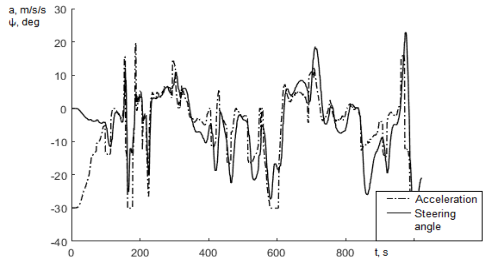


Fig. 6. Time series of $\psi(t)$ and $a_y(t)$ in simulation

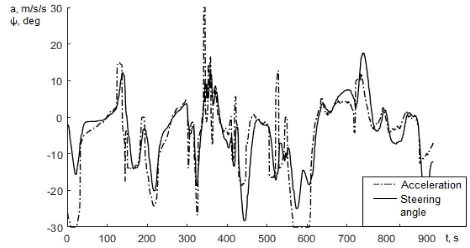


Fig. 7. Time series of $\psi(t)$ and $a_y(t)$ in reality

From plots it is clearly seen, that hardware tends to saturate [12] in moments, when simulated model stands stiff.

Conclusion

An algorithm of speed and steering angle control for cornering safety was presented. A simple vehicle model for control design and a nonlinear vehicle model for simulation has been used. An algorithm for tire friction coefficient in runtime was developed.

In order to test models the simulator with advanced physics and modelling capabilities was built. Differences between simulation and hardware vehicle were found. Proved tire lateral acceleration saturation in case of skidding.

References

1. Tomizuka M., and Chee W., (1994), Lane Change Maneuver of Automobiles for the Intelligent Vehicle and Highway System, *American Control Conference*, pp. 3586 – 3587 (In English).

2. Starkey J.M., and Smith D.E., (1994), Effects of Model Complexity on the Performance of Automated Vehicle Steering Controller: Controller Development and Evaluation, *Journal of Vehicle System Dynamics*, pp. 627 – 645 (In English)

3. Sundar S., and Shiller Z., (1998), Emergency Lane-Change Maneuvers of Autonomous Vehicles, *Journal of Vehicle Dynamic Systems, Measurement and Control*, pp. 37–44 (In English).

4. Narby E, (2006), Modeling and Estimation of Dynamic Tire Properties, *Linkoepings University,* pp. 19 – 31 (In English).

5. Pacejka H.B., (2002), Tyre and Vehicle Dynamics, *Butterworth Heinemann*, Oxford, UK (In English).

6. Kim J., Park C., Hwang S., Hori Y., and Kim H., (2010), Control Algorithm for an Independent Motor-Drive Vehicle, *IEEE Transactions on Vehicular Technology*, Vol. 59, pp. 3213 – 3222 (In English).

7. He Z., and Ji X., (2012), Nonlinear Robust Control of Integrated Vehicle Dynamics, *Vehicle System Dynamics*, Vol. 50, pp. 247 – 280 (In English).

8. Limroth J., (2009), Real-Time Vehicle Parameter Estimation, *Clemson University*, Clemson, SC, USA (In English).

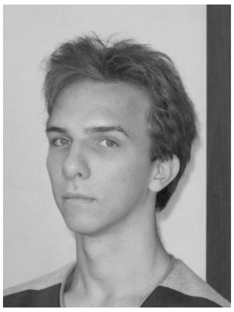
9. Ahmadi J., Sedigh A. K., and Kabganian M., (2009), Adaptive Vehicle Lateral-Plane Motion Control Using Optimal Tire Friction Forces with Saturation Limits Consideration, *IEEE Transactions on Vehicular Technology*, Vol. 58, pp. 4098 – 4107 (In English).

10. He P., and Hori Y., (2006), Optimum Traction Force Distribution for Stability Improvement of 4WDEV in Critical Driving Condition, *Proceedings of the 9th IEEE International Workshop on Advanced Motion Control*, Istanbul, Turkey, pp. 596 – 601 (In English).

11. Khalil H., (2002), Nonlinear Systems, *Prentice-Hall*, Englewood Cliffs, NJ, USA, 3rd edition (In English).

12. Lhomme-Desages D., Grand Ch., and Guinot J.C., (2007), Trajectory Control of a Four-Wheel Skid-Steering Vehicle over Soft Terrain Using a Physical Interaction Model, *In Proceedings of ICRA'07: IEEE/Int. Conf. on Robotics and Automation*, pp. 1164 – 1169, Roma, Italy (In English).

Received 07.06.2016



Tymchenko
Borys Igorovych, bachelor
System Software department
Institute of Computer Systems
Odessa National Polytechnic University.
Tel.: 0664471907.
E-mail:tim4bor@gmail.com

Samodelok
Vladyslav Viktorovych,
3rd year student
Computerized control systems department, Institute of Computer Systems, Odessa National Polytechnic University.
Tel.: 0674669535.
E-mail: Vladfux4@gmail.com

Putilina
Daria Viktorovna,
1st year student
System Software department, Institute of Computer Systems, Odessa National Polytechnic University.
Tel.: 0671029831.
E-mail:
emillyguner@gmail.com

Galchonkov
Oleg Mykolayovych,
PhD., docent,
Computer Science department, Institute of Computer Systems Odessa National Polytechnic University.
Tel.: 0674880459.
E-mail:
o.n.galchonkov@gmail.com