Lateral Control Calibration and Testing in a Co-simulation Framework for Automated Vehicles

Duc-Tien Bui *, Hexuan Li, Francesco De Cristofaro, Arno Eichberger

Posted Date: 8 November 2023

doi: 10.20944/preprints202311.0567.v1

Keywords: Linear Quadratic Regulator; Calibration Optimization; Virtual Simulation; Automated Driving

Article

Lateral Control Calibration and Testing in a Co-Simulation Framework for Automated Vehicles

Duc-Tien Bui *, Hexuan Li , Francesco De Cristofaro and Arno Eichberger

Institute of Automotive Engineering, Graz University of Technology, 8010 Graz, Austria; d.t.bui@tugraz.at (D.B.); hexuan.li@tugraz.at (H.L.); francesco.decristofaro@tugraz.at (F.D); arno.eichberger@tugraz.at (A.E.)

* Correspondence: d.t.bui@tugraz.at (D.B.)

Abstract: Lateral vehicle control is a high importance in automated vehicles as it directly influences the vehicle's performance and safety during operation. The Linear Quadratic Regulator (LQR) controller stands out due to its high-performance characteristics and is used in the open source for self driving functions. However, a notable limitation of the current approach is the manual calibration of LQR controllers based on the experience and intuition of the designers, leading to empirical uncertainties. To address this issue and enhance the lateral control performance, this paper concentrates on refining the LQR by employing three optimization algorithms: Artificial Bee Colony Optimization (ABC), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). These algorithms aim to overcome the reliance on empirical methods and enable a data-driven approach to LQR calibration. By comparing the outcomes of these optimization algorithms to the manual LQR controller within an offline multibody simulation as testing platform, the study highlights the superiority of the best-performing optimization approach. Following this, the optimal algorithm is implemented on a real-time system for the full vehicle level, revealing the Model-in-the-Loop and the Hardware-in-the-Loop gap up to 78,89% with lateral velocity when we use Relative Error Criterion (REC) method to validate and 2.35m with lateral displacement when considering by maximum absolute value method.

Keywords: linear quadratic regulator; calibration optimization; virtual simulation; automated driving

1. Introduction

The advancement of highly automated vehicles holds a critical position in the field of automotive engineering. Key advantages of highly automated driving include improving road safety, particularly by minimizing driver errors and making efficient use of commuters' travel time[1,2]. Perception, trajectory planning and control are three main parts in the structure of automated vehicles, where the control part allows the car to follow the trajectory, which has been determined in the trajectory planning parts. Because of the complex interconnections between the vehicle's lateral and longitudinal dynamics, designing a controller needs to be considered carefully [3] and it continues to remain a challenge. There are various control techniques that have been used for trajectory tracking in automated vehicles, such as: PID [4,5], Linear Quadratic Regulator (LQR) [6–9], Sliding Mode Control (SMC) [10,11], Robust Control [12], Model Predictive Control (MPC) [8,9,12–15] and Reinforcement Learning [16,17]. However, most of them are used to control longitudinal and lateral dynamics separately. Recently, controlling both longitudinal and lateral dynamics has been applied. Nada et al. [18] designed a multi-input–multioutput (MIMO) linear MPC with some constraints in the vehicle dynamics, in which the reference path is tracked based on the steering angle and angular velocity. In [4], Zhou et.al adopted the MPC and Nonliner Model Predictive Control (NMPC) method to cope with the nonlinear MIMO problem for the vehicle's lateral stability. Although MPC controller can use for nonlinear dynamics to solve with constraints, MPC prevents conducting an experiment on a real-time system because of the increasing of model complexity and constraints as well as the huge amount of computation. SMC technique is proposed to control by combining lateral and longitudinal dynamic [19]. However, when applying SMC technique the chattering phenomenon often appears when acquiring robustness. To

 doi:10.20944/preprints202311.0567.v1

overcome this obstacle, in [20] the authors applied a new sliding mode that reduces chattering and makes state variables converge faster. However, rough road surfaces can lead to compromised path tracking accuracy and overall system stability.

LQR approach is used popularly in automated vehicle controllers because of algorithmic simplicity, high-precision performance and satisfaction with dynamic constraints. Zhang [21] compared the robust controller with LQR and MPC in terms of performance as well as computation. In the parking scenario, the LQR controller has better performance. Regarding computation, LQR requires less in comparison with MPC while it has the same performance indicators in some cases. However, LQR methods usually do not consider disturbances which can contribute to system errors. To cope with this problem, Kapania et al. [22] designed a controller by combining feedback and feedforward steering. The aim is to keep the stability of the vehicle under hard maneuvering conditions as well as minimize path deviation. However, the steady-state path deviation drastically increases at high velocity. In [23] the authors aslo proposed a LQR controller using feedforward and predictive steering for lateral dynamic, which helps vehicle driving in complex conditions. Although LQR controllers have been widely applied to automated vehicles, most of LQR choose weight factors based on empirical consideration. To get rid of empiricism, some studies added algorithms to optimize LQR controller in order to improve performance. In [24–26] the authors used GA, fuzzy control and PSO to choose the optimize weight factors for LQR controller. The results show the effectiveness of these algorithms regarding tracking accuracy and stability of vehicle. Although promising results are shown in these studies, there is still potential for further improvement in the accuracy of the controller.

On the other hand, because the requirement for automated vehicles level is higher, the scenarios for testing become complex. Consequently, novel challenges to the reliability of automated vehicles emerge in testing and validation. Additionally, the growing need for testing and validation in terms of Advanced Driver Assistance Systems (ADAS) as well as Automated Driving technologies arises because real-world road scenarios have covered more and more [27,28]. Consequently, X-in-the-Loop (XiL) has emerged and become a predominant approach for scenario-driven simulation testing of automated vehicles. The "X" expressed the various development focuses: model, software, processor, hardware, vehicle, and driver. Hye Young An [29] proposed a path planning algorithm and Pure Pursuit in real-time to control vehicle by considering detected lanes and some constraints. The authors considered only a single lane and the error is approximately 1.147m when the car exits the roundabout. Taekgyu Lee el at. [30] employed a DNN-based controller to control the car. This method reduces the computational load in comparison with the previous NMPC method. In [31,32], the authors also used MPC and NMPC to control automated vehicles in real-time system, the results show the efficiency, robustness as well as feasibility of these methods. In [33] the authors used a low-level MPC to control the small-scale race cars. Based on the simulation and experimental findings, it appears that opting for a more cautious approximation with the discriminating kernel leads to a safer driving style. Most of the previous works did not compare the gap between office simulation and real-time system. This happened because of the differences in purpose, complexity, resources, and constraints.
The key contributions of this study are:

- Three optimization algorithms, namely Artificial Bee Colony, Particle Swarm Optimization, and Genetic Algorithm are implemented to find the best coefficient of the LQR controller. The primary objective is to eliminate the external disturbances arising from the desired trajectory. The algorithm optimizations is simulated on CarMaker and Matlab/Simulink software. The effectiveness of three algorithmic enhancements is compared to LQR controller performance without using them. Afterward, the results obtained from three algorithm optimizations are compared together to choose the best algorithm for the Model-in-the-Loop (MiL) simulation.
- The best algorithm for MiL simulation is used to simulate on a real-time system to assess performance. The chosen algorithm is simulated on MiL and Hardware-in-the-Loop (HiL). The outcomes reveal the gap between the MiL simulation and HiL for the vehicle model under consideration.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

The remainder of this study is structured as follows: Section 2 contains the vehicle dynamic of the model, the framework of Apollo as well as the optimization. In Section 3, the model and controllers are simulated on an office environment and real-time system using the multibody software CarMarker. Section 4 discusses about results and the Section 5 summarizes the research.

2. Vehicle Dynamics and Control

Apollo is a widely recognized open-source SAE Level 4 AD platform that was introduced by Baidu company [34]. It includes a comprehensive suite of hardware and software solutions for various aspects of AD, including perception, planning, and control.

2.1. Vehicle Model

Figure 1 depicts the vehicle dynamic model as:

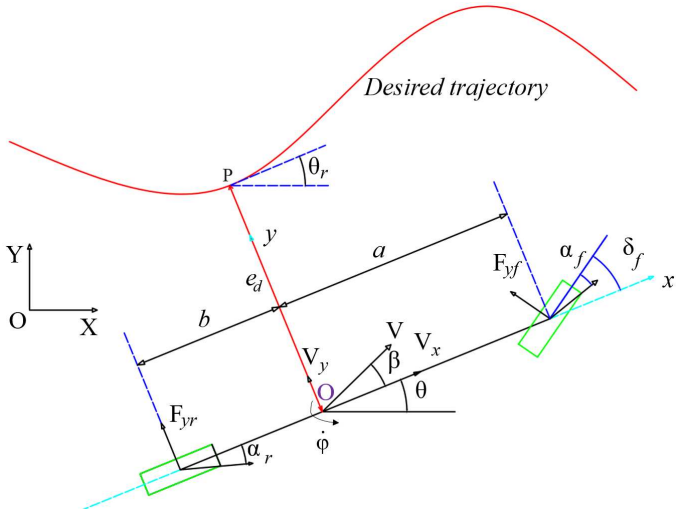

Figure 1. Vehicle dynamic model.

In this model, we assume that the model is simplified as the lateral dynamic model with two degrees of freedom, vertical movement is ignored since the vehicle drives in a single plane. α_f , α_r are steering angles in the front and rear wheels, respectively. The front and rear steering angle are small and noted by δ_f and δ_r, so the lateral force and side slip angle has a linear relationship, which is acceptable for tire slip angles about 3 degree. The lateral load transfer effects which occur due to lateral acceleration is neglected and the suspension system's effect is not considered.
O is mass center point, a and b are the distances from O to the front and rear axles, φ is the yaw angle, $\dot{\varphi}$ is yaw rate. C_f and C_r are the lateral stiffness of the tires in the front and rear wheels, F_{y_f} is the lateral force of the front wheel and F_{y_r} is the lateral force of the rear wheel. v_x, v_y are vehicle's longitudinal and lateral velocity, respectively.
The dynamic equations of this model are noted as [35]:

$$ma_y = F_{y_f} + F_{y_r} \tag{1}$$

$$I_z\ddot{\varphi} = aF_{y_f} - bF_{y_r} \tag{2}$$

$$\dot{v_x} = v_y\dot{\varphi} + a_x \tag{3}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

where m is the mass of vehicle, I_z is the rotation inertia of vehicle around vertical axis.
Because the front wheel steering angle is small, so the lateral force and side slip angle has a linear relationship, the dynamic equations (1) (2) is obtained as follows:

$$\begin{cases} \dot{v_y} = \dfrac{C_f + C_r}{mv_x} v_y + (\dfrac{aC_f - bC_r}{mv_x} - v_x)\dot{\varphi} - \dfrac{C_f}{m}\delta \\ \ddot{\varphi} = \dfrac{aC_f - bC_r}{I_z v_x} v_y + \dfrac{a^2 C_f + b^2 C_r}{I_z v_x}\dot{\varphi} + \dfrac{aC_f}{I_z}\delta \\ \dot{v_x} = \dot{y}\dot{\varphi} + a_x \end{cases} \tag{4}$$

When vehicle follows a reference path, lateral error and heading angle error will occur. Illustrated in Figure 1, the lateral error, denoted as e_d represents the shortest distance between the point O and projected point P on the reference path. Meanwhile, the heading angle error, referred to as e_θ is the difference between the actual heading angle of the vehicle θ and the reference heading angle θ_r. For the sake of simplicity, the side slip angle is assumed $\beta = 0$ at the point O, and the heading angle error is calculated as $e_\theta = \theta - \theta_r$. In practical control, the controller's task is to promptly eliminate these two errors in real-time to ensure the vehicle stays on track with the planned path. With these errors in mind, it becomes possible to compute the first-order derivatives of the lateral error $\dot{e_d}$ and the heading angle error $\dot{e_\varphi}$:

$$\dot{e_d} = v_x e_\varphi + v_y \tag{5}$$

$$\dot{e_\varphi} = \dot{\varphi} - \dot{\varphi_r} \tag{6}$$

The Equations (5) and (6) are substituted in Equation (4), can be obtained:

$$\ddot{e_d} = \frac{C_f + C_r}{mv_x}\dot{e_d} - \frac{C_f + C_r}{m}e_\varphi + \frac{aC_f - bC_r}{mv_x}\dot{e_\varphi} + (\frac{aC_f - bC_r}{mv_x} - v_x)\dot{\theta_r} - \frac{C_f}{m}\delta_f \tag{7}$$

$$\ddot{e_\varphi} = \frac{aC_f - bC_r}{I_z v_x}\dot{e_d} - \frac{aC_f - bC_r}{I_z}e_\varphi + \frac{a^2 C_f + b^2 C_r}{I_z v_x}\dot{e_\varphi} + \frac{a^2 C_f + b^2 C_r}{I_z v_x}\dot{\theta_r} - \frac{aC_f}{I_z}\delta_f \tag{8}$$

The equations (7) to (8) can be rewritten as:

$$\begin{Bmatrix} \dot{e_d} \\ \ddot{e_d} \\ \dot{e_\varphi} \\ \ddot{e_\varphi} \end{Bmatrix} = \begin{Bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{C_f + C_r}{mv_x} & -\dfrac{C_f + C_r}{m} & \dfrac{aC_f - bC_r}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{aC_f - bC_r}{I_z v_x} & -\dfrac{aC_f - bC_r}{I_z} & \dfrac{a^2 C_f + b^2 C_r}{I_z v_x} \end{Bmatrix} \begin{Bmatrix} e_d \\ \dot{e_d} \\ e_\varphi \\ \dot{e_\varphi} \end{Bmatrix} + \begin{Bmatrix} 0 \\ -\dfrac{C_f}{m} \\ 0 \\ \dfrac{aC_f}{I_z} \end{Bmatrix} \delta_f + \begin{Bmatrix} 0 \\ \dfrac{aC_f - bC_r}{mv_x} - v_x \\ 0 \\ \dfrac{a^2 C_f + b^2 C_r}{I_z v_x} \end{Bmatrix} \dot{\theta_r} \tag{9}$$

Equation (9) can be rewritten in the state - space representations as:

$$\dot{X} = AX + BU + C\dot{\theta} \tag{10}$$

with: X=[$\dot{ed}, \ddot{ed}, \dot{e\varphi}, \ddot{e\varphi}$] is the state vector; U = [δ_f] is the control input.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

2.2. Lateral Controller

In this study, the path planning process involves a series of reference points. This discretization of data is necessary for practical implementation. To control the vehicle along this discrete trajectory, a discrete Linear-Quadratic Regulator (dLQR) controller is used. The Equation (10) is discretized to design dLQR controller, which governs the vehicle's dynamics. During this process, we neglect the effect of C_{θ_r} and apply the midpoint Euler and the forward Euler approach to clarify the model while preserving essential characteristics. As a result, we obtain the equation that describes the discrete tracking errors as:

$$\dot{X}_{k+1} = \bar{A}X + \bar{B}U \tag{11}$$

with:

$$\bar{A}_k = (I - \frac{Adt}{2})^{-1}(I + \frac{Adt}{2})^{-1}; \bar{B} = Bdt \tag{12}$$

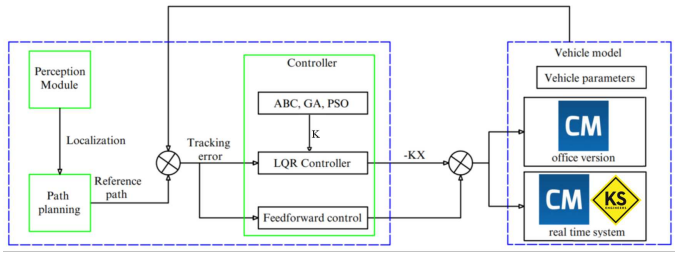

Figure 2. Lateral control structure.

Figure 2 shows the entire structure of the LQR controller for an automated vehicle, which includes four main parts: perception, path planning, controller and vehicle model. In path planning, the EM planner [36] is used to generate reference path and path tracking errors. As the path planning module is not the focus of our work, the exact algorithm can be referenced in the GitHub project [37].
The LQR controller is the main center of the second part. In this study, we use three algorithm optimizations (ABC, GA, PSO) to find the optimized matrix K for the LQR controller as well as calculate δ_f f in the Feedforward control step. Then, the steering angle δ_f is sent to the vehicle model to control a car. We use the BMW5 car model and IPG Carmaker software as a simulation environment. After comparing the K value from three optimization algorithms, we find the best algorithm for the car. Finally, we use this algorithm to simulate on the real-time system and compare the gap results between the MiL and HiL.
The cost function of LQR controller is defined as:

$$\sum_{0}^{\infty}(X_k^T Q X_k + U_k^T R U_k)dt \tag{13}$$

$$Q = diag(q1, q2, q3, q4) \tag{14}$$

$$R = [q5] \tag{15}$$

Where Q weighting matrices of the state error and R is weighting matrices of the control signal. q_1, q_2, q_3, q_4, q_5 are the weight factors of lateral error, lateral error rate, heading error, heading error rate and the steering angle in the front wheel, respectively. Substituting equation (11) into (13), the Lagrange multiplier approach is employed to build the constraints as follows:

$$J = \sum_{k=0}^{n-1}[X_k^T Q X_k + U_k^T R U_k + \lambda_{k+1}^T(\overline{A_k}X_k + \overline{B}U_k) - \lambda_{k+1}^T X_{k+1}] + X_n^T Q X_n \tag{16}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

The Hamiltonian function is defined as:

$$H_k = X_k^T Q X_k + U_k^T R U_k + \lambda_{k+1}^T (\overline{A_k} X_k + \overline{B} U_k) \tag{17}$$

From equation (17) and (16) obtain:

$$J = \sum_{k=0}^{n-1} [H_k - \lambda_k^T X_k] + X_n^T Q X_n + \lambda_0^T X_0 - \lambda_n^T X_n \tag{18}$$

The extreme value of equation (18) is achieved:

$$U_k = -(R + \overline{B}^T P_{k+1} \overline{B})^{-1} \overline{B}^T P_{k+1} \overline{A_k} X_k \tag{19}$$

where $K = (R + \overline{B}^T P_{k+1} \overline{B})^{-1} \overline{B}^T P_{k+1} \overline{A_k}$
P_{k+1} is solved by Riccati equation: P = Q + $\overline{A_k}^T P \overline{A_k}$ - $\overline{A_k}^T P \overline{B} (R + \overline{B}^T P \overline{B})^{-1} \overline{B}^T P \overline{A_k}$
Equation (19) can be rewritten as:

$$U_k = -K X_k \tag{20}$$

where K = $[K_1, K_2, K_3, K_4]$ is the gain of the LQR controller. Substituting equation (20) into equation (10) obtains:

$$\dot{X} = (A_k - BK) X_k + C_k \dot{\theta} \tag{21}$$

According to equation (21), irrespective of the specific value assigned to the gain K, the distance error and heading error of an automated vehicle can not be zero during the control process, indicating the presence of a steady-state error in the system. Consequently, the influence of $C\dot{delta}_r$ is removed and use feedforward control $\delta_{f\,f}$ follow as:

$$U = -KX + \delta_{f\,f} \tag{22}$$

Substitute equation (22) into (10), so that when $\dot{X}$ =0, the formula for the state variable without steady-state error is as follows:

$$X = -(A_k - BK)^{-1} (B\delta_{f\,f} + C_k \dot{\theta}_r) \tag{23}$$

By solving and simplifying equation (23), the following is obtained:

$$\begin{Bmatrix} \dot{e}_d \\ \ddot{e}_d \\ \dot{e}_\varphi \\ \ddot{e}_\varphi \end{Bmatrix} = \begin{Bmatrix} \frac{1}{k_1} (\delta_{f\,f} - \frac{\dot{\theta}_r}{v_x} (a + b - bK_3 - \frac{m v_x^2}{a+b} (\frac{b}{C_f} + \frac{a}{C_r} K_3 - \frac{a}{C_r})) \\ 0 \\ -\frac{\dot{\theta}_r}{v_x} (b + \frac{a}{a+b} \frac{m v_x^2}{C_r}) \\ 0 \end{Bmatrix} \tag{24}$$

Following the Equation (24), when e_d = 0, the feedforward control is:

$$\delta_{f\,f} = -\frac{\dot{\theta}_r}{v_x} [a + b - bK_3 - \frac{m v_x^2}{a+b} (\frac{b}{C_f} + \frac{a}{C_r} K_3 - \frac{a}{C_r})] \tag{25}$$

In Equation (6), we assume that the real heading error can be calculated as $e_\varphi = \varphi - \theta_r$. Reducing the heading error to zero, as well as ensuring that $e_\varphi = \varphi - \theta_r = -\beta$ is crucial when the vehicle reaches a stable state. Consequently, there is no need to devise a feedforward controller for eliminating the steady-state error in e_φ. Additionally, Kapania et al. [22] demonstrated that achieving steady-state equilibrium is still possible even when there are non-zero values for lateral error as well as heading angle error.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

2.3. Calibration Optimization

- Calibration solution in the state of the art
  Matrix K is the gain of the LQR controller and can be calculated from equation (19). The key to the LQR controller lies in the choose of the weight factors of matrices Q and R in equation (14), (15). In the previous study [38], Li el at. chose these weight factors based on the empirical method. First, Q is optimized by setting to R an intermediate value (arbitrarily chosen to be in the order of 10^5) and Q is considered diagonal to simply tuning. Q is tested for a small value Q_{min} (close to 0) and later for a large value Q_{max} (in the order of 10^6). The two results are compared, then an intermediate value Q_{avg} is tested and the process is repeated by considering as new "small" and "large" limits Q_{avg} and the value between Q_{min} and Q_{max} that yielded the best result. The process is then repeated by choosing a new intermediate value between the two limits until an acceptable result is produced. For R optimization the same approach is followed setting Q to the optimized value just found.
- Artificial Bee Colony Algorithm
  The flowchart of the ABC algorithm and LQR controller is shown in Figure 3. The employed bees actively seek K values in the vicinity of their remembered food source, all the while communicating information about these K values to the onlooker bees. The onlooker bees are likely to select good K values from those based on the fitness function and also evaluate K values using the cost function. A few employed bees translate into the scout bees and search for new food sources until satisfy the condition.

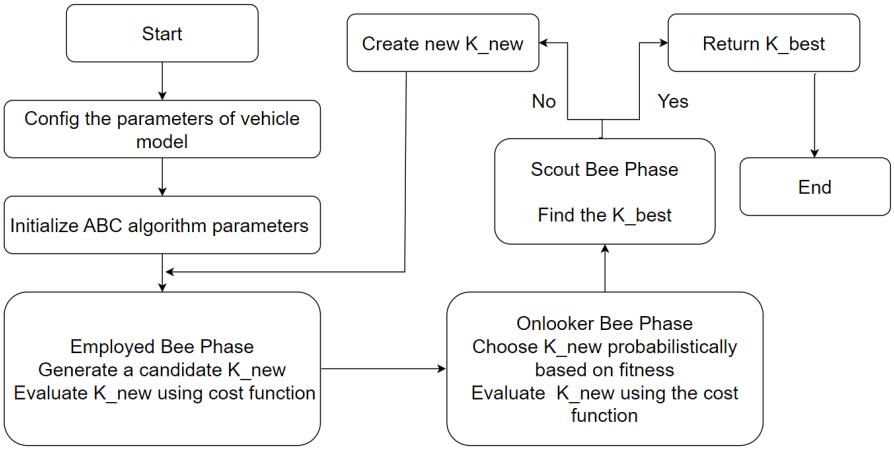

Figure 3. Flow chart of ABC and LQR controller.

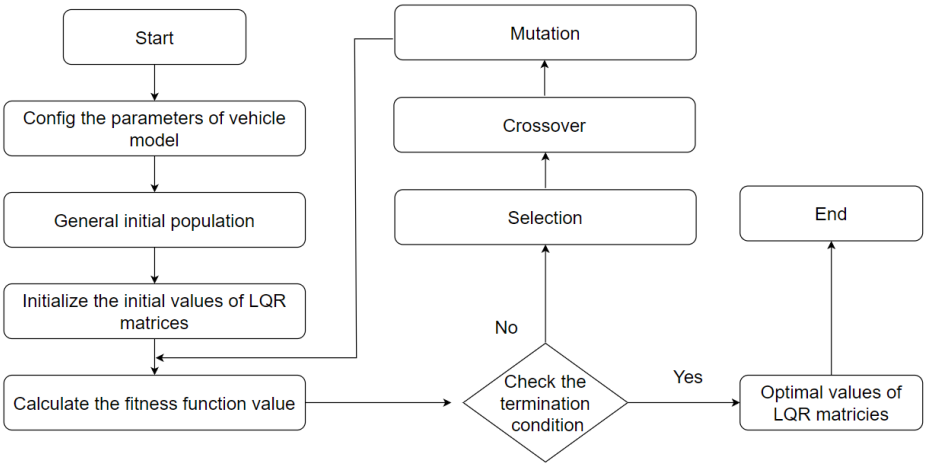

Figure 4. Flow chart of GA and LQR controller.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

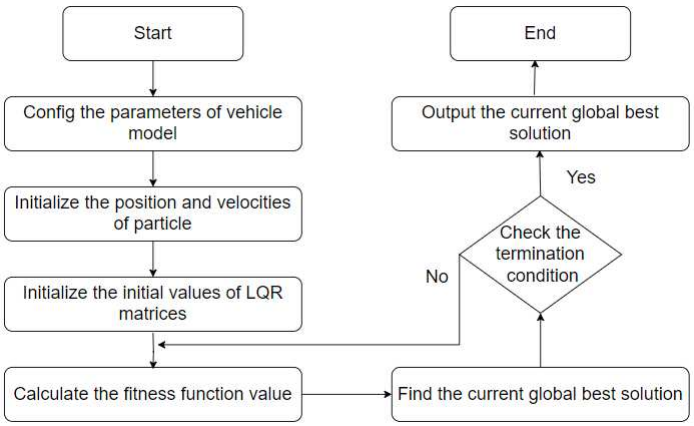

Figure 5. Flow chart of PSO and LQR controller.

- Genetic algorithm
  The GA drawing inspiration from biological principles such as mutation, crossover, and selection. The GA commences with config the parameters of BMW car, initial K values as well as randomly generate individuals, initiating the evolutionary journey. In each generation, the fitness of every individual is evaluated, typically by assessing the value of the fitness function. Stochastic selection is employed to favor more fit individuals from the current population. These selected individuals' genomes are then subject to modifications, such as recombination and possibly random mutations, to generate a new generation of candidate solutions. This cyclic process continues, with the newly formed generation becoming the basis for the subsequent iteration. The GA advances through iterations until it reaches a termination condition. The flowchart of GA-LQR is shown in Figure 4.
- Particle swarm optimization
  Figure 5 illustrates the flowchart of PSO algorithm and LQR controller. The optimal weight factor is found by searching on the global in order to enhance the LQR controller performance. Firstly, the parameters of BMW car model are configured. Then, we set the initial particles's position and velocities as well as the initial values of LQR matrices. After that, the algorithm will calculate the fitness function value and find the global best solution. If the condition is satisfied, the algorithm is stopped, otherwise, the algorithm continues to run until satisfies the condition.

3. Simulation and Results

3.1. Model-in-the-Loop Simulation

In this study, a BMW5 model is used for simulation. The model was calibrated with experiments done in the Institute of Automotive Engineering laboratory and on a proving ground. Firstly, test the BMW 5 car on the real road to measure parameters and save it into the datasheet. Then, a simple model with parameters adjustment from the datasheet is created. After that, the car was simulated on the test road, compared the parameters were measured with the simulation. Finally, calibrate the steering model and stabilizer model for good fitting of the curves and choose the best tire model.The results of the model calibration refer to [39]. The BMW5 model is simulated in the CarMaker environment on a 50-meter circular road, as depicted in Figure 6. Lateral position error was evaluated during a total driving duration of 48 seconds, with a maximum speed is 50km/h.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

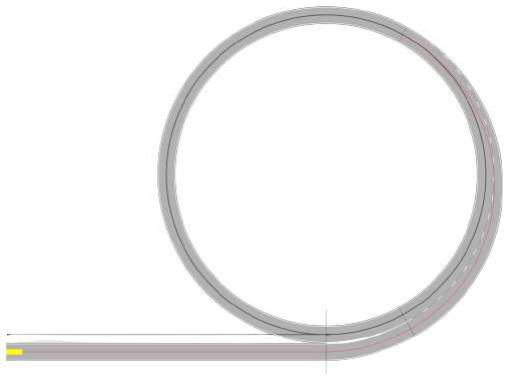

Figure 6. Road test for automated vehicle.

Figure 7 illustrates the vehicle's steering angle under four control schemes: the manual LQR controller, the PSO based on LQR (PSO-LQR) controller, the ABC based on LQR (ABC-LQR) and GA based on LQR controller (GA-LQR).

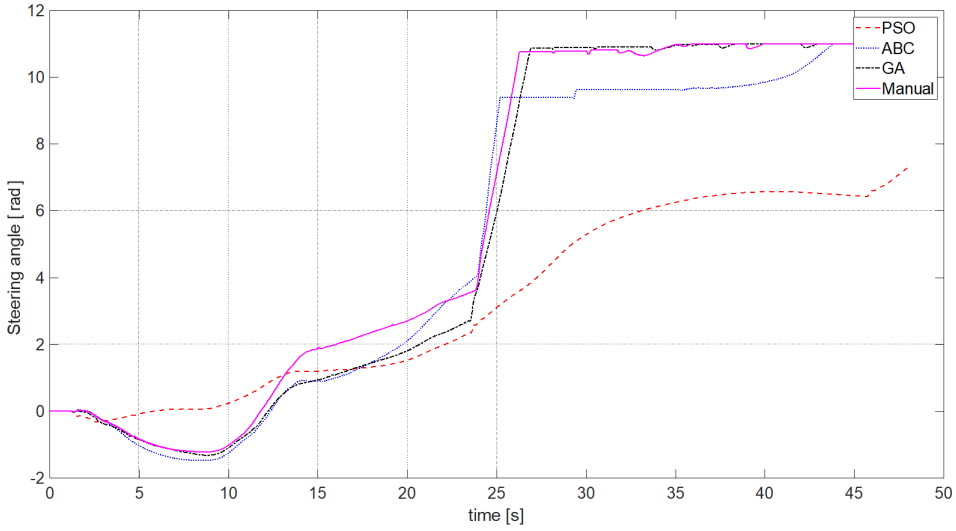

Figure 7. Steering angle of automated vehicle.

Table 1. Lateral error comparison between optimization algorithms.

Compare	Manual	ABC	GA	PSO
Absolute max values dL	1.2026	2.6069	1.5611	0.0576
Absolute mean values dL	0.0249	0.0242	0.0207	0.0061
Absolute max values L	3.5481	4.4860	3.5151	0.7532
Absolute mean values L	0.5854	0.7696	0.5549	0.1582

3.2. Hardware-in-the-Loop Simulation

In the previous works, Li et al. [27,28,40] the co-simulation framework for Virtual Vehicle Test Bench is introduced. CarMaker and Matlab/Simulink are used together to build a co-simulation software system. The Multi-Body Simulation (MBS) environment is provided by CarMaker software, including vehicle dynamics, sensor models as well as virtual environment. The CarMaker is implemented on a real-time processing unit called "Xpack4". Both CarMaker and Xpack 4 are from IPG Automotive GmbH [41]. Moreover, the whole vehicle test bench is controlled by the automation-system software Tornado from KS Engineers [42]. The users has capacity to configure Virtual Vehicle Test Bench using Tornado. In adddition, ADAC (5 kHz) from KS Engineers [42], which is real-time

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 November 2023 doi:10.20944/preprints202311.0567.v1

testing, controlling and monitoring system which developed for the Virtual Vehicle Test Bench in real-time. EtherCAT topology protocol enables real-time synchronous simulation communication between Tornado, ADAC and XPack4 real-time systems. In this study, the same framework is used, which comprises a virtual environment and a Virtual Vehicle Test Bench. This framework (RT-ADF) allows multiple software are integrated and executed on real-time hardware platforms. This work demonstrates an offline Virtual Vehicle Test Bench. If the simulation model is successfully executed and validated offline, it can be transferred to the real test bench. Therefore, this is a good approach to improve efficiency and optimize the work. Table 2 depicts the results of the vehicle model in a real-time system.

Table 2. Lateral error comparison between MiL and HiL.

Compare	Absolute max values dL	Absolute mean values dL	Absolute max values L	Absolute mean values L
MiL	0.0576	0.0061	0.7532	0.1582
HiL	0.2894	0.0098	2.5857	0.5339

In [43], a quantitative comparison method is the Relative Error Criterion (REC) is used to validate results as:

$$Error_L = \frac{|Peak_{test} - Peak_{sim}|}{|Peak_{test}|} * 100\% = 53.23\% \tag{26}$$

$$Error_dL = \frac{|Peak_{test} - Peak_{sim}|}{|Peak_{test}|} * 100\% = 78.89\% \tag{27}$$

We use the calculate the maximum absolute value of the error as:

$$Error_L = max(|L_{test} - L_{sim}|) = 2.35 \tag{28}$$

$$Error_dL = max(|d_{Ltest} - d_{Lsim}|) = 0.28 \tag{29}$$

4. Discussion

Figure 7 shows the steering angle of the vehicle in four cases using manual calibration, ABC, GA and PSO algorithms. In the interval spanning from the 4th to the 10th second, alterations in the steering angle were observed for the Manual calibration, GA, and ABC algorithms, registering an approximate variation of 1.9 rad. Conversely, the steering angle adjustment remained negligible when employing the PSO algorithm. Moreover, at the 24th second, a sudden soar in the steering angle occurred within a span of 3 seconds, for the Manual calibration, GA, and ABC algorithms. This value then stabilized thereafter at approximately 11 rad. In contrast, the steering angle change generated by the PSO algorithm exhibited a gradual increment, culminating at 8 rad by the conclusion of the aforementioned timeframe. Therefore, PSO algorithms can prevent the centrifugal acceleration experienced in the vehicle. Consequently, this incremental approach has the capacity to enhance both the precision of lateral displacement and lateral velocity measurements.

Table 1 compares the lateral error and lateral velocity using three algorithms as well as manual calibration. Regarding root mean square values for lateral displacement, the PSO algorithm exhibits a value of 0.2274 meters, outperforming manual calibration (1.124 meters), ABC (1.5092 meters), and GA (1.0747 meters). Moreover, the results across various values in the PSO algorithm consistently demonstrate superior performance compared to the other algorithms under consideration.

The lateral displacement and lateral velocity differences between the MiL and HiL are evident in Table 2. Notably, simulation values obtained from the HiL are larger than those generated by the MiL. Similarly, there are errors between the results obtained from the MiL and the HiL, particularly up to 53.23% with lateral displacement and 78,89% lateral velocity when we use REC method to validate. Because the value of lateral velocity is too small, this REC value is too much. When we use the maximum absolute value method to validate the difference between the results obtained from

the MiL and the HiL, the figure for lateral displacement and lateral velocity are 2.35m and 0.28m/s, respectively. This divergence can be caused by the cyclic delay existing between Xpack and ADAS and the computer.

5. Conclusions

This study presents three optimization algorithms ABC, GA and PSO to fine-tune LQR parameters based on open source software for automated vehicle Apollo frameworks. These algorithms aim to overcome the reliance on empirical methods and enable a data-driven approach to LQR calibration. The study highlights the superiority of the PSO algorithm by comparing the outcomes of this optimization algorithm to the manual, ABC, GA LQR controller within the MiL simulation. Finally, the PSO algorithm is simulated on a HiL, the result shows the gap between the MiL and the HiL simulation. In the future, our works will continue to focus on optimizing and enhancing our system, especially focusing on the controllers and path planning, where MPC and Reinforcement learning can be applied. Moreover, the more complex traffic scenarios will be used to test and validate the reality of Apollo on the test bench and real car.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, D.B. and H.L.; methodology, D.B. and H.L.; software, D.B. and H.L.; validation, D.B., H.L. and F.D.; formal analysis, D.B. and H.L.; investigation, D.B. and H.L.; resources, D.B. and H.L.; data curation, D.B. and H.L.; writing—original draft preparation, D.B. and H.L.; writing—review and editing, D.B., H.L., F.D. and A.E.; visualization, D.B. and H.L.; supervision, A.E.; project administration, A.E.; funding acquisition, A.E. All authors have read and agreed to the published version of the manuscript.

Funding: Funding by the Graz University of Technology. This activity is part of the research project InVADE (FFG nr. 889349) and has received funding from the program Mobility of the Future, operated by the Austrian research funding agency FFG. Mobility of the Future is a mission-oriented research and development program to help Austria create a transport system designed to meet future mobility and social challenges.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We extend our gratitude to Zhengguo Gu for his valuable internship support, which greatly contributed to the success of this project.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. Additionally, the authors declare that this study received funding from the Austrian research funding agency FFG within the scope of the research project InVADE. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

References

1. Singh, S. *Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey*, 2018.
2. Paden, B.; Čáp, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles* **2016**, *1*, 33–55.
3. Chebly, A.; Talj, R.; Charara, A. Coupled longitudinal/lateral controllers for autonomous vehicles navigation, with experimental validation. *Control Engineering Practice* **2019**, *88*, 79–96.
4. Zhou, H.; Jia, F.; Jing, H.; Liu, Z.; Güvenç, L. Coordinated longitudinal and lateral motion control for four wheel independent motor-drive electric vehicle. *IEEE transactions on Vehicular Technology* **2018**, *67*, 3782–3790.
5. Han, Y.; Zhu, Q.; Xiao, Y. Data-driven control of autonomous vehicle using recurrent fuzzy neural network combined with PID method. 2018 37th chinese control conference (ccc). IEEE, 2018, pp. 5239–5244.
6. Yang, T.; Bai, Z.; Li, Z.; Feng, N.; Chen, L. Intelligent vehicle lateral control method based on feedforward+ predictive LQR algorithm. Actuators. MDPI, 2021, Vol. 10, p. 228.
7. Piao, C.; Liu, X.; Lu, C. Lateral control using parameter self-tuning LQR on autonomous vehicle. 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS). IEEE, 2019, pp. 913–917.

 doi:10.20944/preprints202311.0567.v1

8. Xu, S.; Peng, H. Design, analysis, and experiments of preview path tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems* **2019**, *21*, 48–58.
9. Varma, B.; Swamy, N.; Mukherjee, S. Trajectory tracking of autonomous vehicles using different control techniques (pid vs lqr vs mpc). 2020 International conference on smart technologies in computing, electrical and electronics (ICSTCEE). IEEE, 2020, pp. 84–89.
10. Tagne, G.; Talj, R.; Charara, A. Design and comparison of robust nonlinear controllers for the lateral dynamics of intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems* **2015**, *17*, 796–809.
11. Li, A.; Niu, C.; Li, S.; Huang, X.; Xu, C.; Liu, G. Research on intelligent vehicle trajectory planning and control based on an improved terminal sliding mode. *Applied Sciences* **2022**, *12*, 2446.
12. Yang, K.; Tang, X.; Qin, Y.; Huang, Y.; Wang, H.; Pu, H. Comparative study of trajectory tracking control for automated vehicles via model predictive control and robust H-infinity state feedback control. *Chinese Journal of Mechanical Engineering* **2021**, *34*, 1–14.
13. Alcalá, E.; Puig, V.; Quevedo, J. LPV-MPC control for autonomous vehicles. *IFAC-PapersOnLine* **2019**, *52*, 106–113.
14. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology* **2007**, *15*, 566–580.
15. Abbas, M.A.; Milman, R.; Eklund, J.M. Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. *Canadian journal of electrical and computer engineering* **2017**, *40*, 12–22.
16. Folkers, A.; Rick, M.; Büskens, C. Controlling an autonomous vehicle with deep reinforcement learning. 2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019, pp. 2025–2031.
17. Dai, X.; Li, C.K.; Rad, A.B. An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems* **2005**, *6*, 285–293.
18. Awad, N.; Lasheen, A.; Elnaggar, M.; Kamel, A. Model predictive control with fuzzy logic switching for path tracking of autonomous vehicles. *ISA transactions* **2022**, *129*, 193–205.
19. Ohara, H.; Murakami, T. A stability control by active angle control of front-wheel in a vehicle system. *IEEE Transactions on Industrial Electronics* **2008**, *55*, 1277–1285.
20. Wang, H.; Han, J.; Zhang, H. Lateral Stability Analysis of 4WID Electric Vehicle Based on Sliding Mode Control and Optimal Distribution Torque Strategy. Actuators. MDPI, 2022, Vol. 11, p. 244.
21. Zhang, W. A robust lateral tracking control strategy for autonomous driving vehicles. *Mechanical Systems and Signal Processing* **2021**, *150*, 107238.
22. Kapania, N.R.; Gerdes, J.C. Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling. *Vehicle System Dynamics* **2015**, *53*, 1687–1704.
23. Yuan, T.; Zhao, R. LQR-MPC-Based Trajectory-Tracking Controller of Autonomous Vehicle Subject to Coupling Effects and Driving State Uncertainties. *Sensors* **2022**, *22*, 5556.
24. Lu, A.; Lu, Z.; Li, R.; Tian, G. Adaptive LQR Path Tracking Control for 4WS Electric Vehicles Based on Genetic Algorithm. 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI). IEEE, 2022, pp. 1–6.
25. Wang, Z.; Sun, K.; Ma, S.; Sun, L.; Gao, W.; Dong, Z. Improved Linear Quadratic Regulator Lateral Path Tracking Approach Based on a Real-Time Updated Algorithm with Fuzzy Control and Cosine Similarity for Autonomous Vehicles. *Electronics* **2022**, *11*, 3703.
26. Zhang, Y.; Gao, F.; Zhao, F. Research on Path Planning and Tracking Control of Autonomous Vehicles Based on Improved RRT* and PSO-LQR. *Processes* **2023**, *11*, 1841.
27. Li, H.; Nalic, D.; Makkapati, V.; Eichberger, A.; Fang, X.; Tettamanti, T. A real-time co-simulation framework for virtual test and validation on a high dynamic vehicle test bed. 2021 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2021, pp. 1132–1137.
28. Li, H.; Makkapati, V.P.; Wan, L.; Tomasch, E.; Hoschopf, H.; Eichberger, A. Validation of Automated Driving Function Based on the Apollo Platform: A Milestone for Simulation with Vehicle-in-the-Loop Testbed. *Vehicles* **2023**, *5*, 718–731.
29. An, H.Y.; Choi, W.S.; Choi, S.G. Real-Time Path Planning for Trajectory Control in Autonomous Driving. 2022 24th International Conference on Advanced Communication Technology (ICACT). IEEE, 2022, pp. 154–159.
30. Lee, T.; Seo, D.; Lee, J.; Kang, Y. Real-Time Drift-Driving Control for an Autonomous Vehicle: Learning from Nonlinear Model Predictive Control via a Deep Neural Network. *Electronics* **2022**, *11*, 2651.

31. Liang, Z.; Chen, Y.; Zhao, J. Real-time Parameter Updating and Path-following Control for Autonomous Vehicles on Slope Roads. *International Journal of Control, Automation and Systems* **2022**, *20*, 2178–2190.
32. Collado-Gonzalez, I.; Gonzalez-Garcia, A.; Sotelo, C.; Sotelo, D.; Castañeda, H. A Real-Time NMPC Guidance Law and Robust Control for an Autonomous Surface Vehicle. *IFAC-PapersOnLine* **2021**, *54*, 252–257.
33. Liniger, A.; Lygeros, J. Real-time control for autonomous racing based on viability theory. *IEEE Transactions on Control Systems Technology* **2017**, *27*, 464–478.
34. https://developer.apollo.auto/index.html.
35. Jazar, R.N. *Vehicle dynamics*; Vol. 1, Springer, 2008.
36. Fan, H.; Zhu, F.; Liu, C.; Zhang, L.; Zhuang, L.; Li, D.; Zhu, W.; Hu, J.; Li, H.; Kong, Q. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048* **2018**.
37. https://github.com/ApolloAuto/apollo.git **2022**.
38. Li, H.; De Cristofaro, F.; Orucevic, F.; Zhengguo, G.; Eichberger, A. Quantitative Analysis of the Impact of Baidu Apollo Parameterization on Trajectory Planning in a Critical Scenario. ZIRP 2023, 2023.
39. Erik, R.B. *Aufbau von validierten Fahrzeugmodellen*; Master thesis. submitted to Graz University of Technology, 2023.
40. T. Kanuric BSc, H. Li MSc, A.P.D.I.D.t.A.E. Advanced Lane Detection Model for the Virtual Development of Highly Automated Functions. *International ViennaMotor Symposium 2023* **2023**.
41. https://ipg-automotive.com/en/products-solutions/software/carmaker/. Technical report.
42. https://www.ksengineers.com/. Technical report.
43. Proposed CM-S-014 Modelling & Simulation – for consultation. Technical report, European Union Aviation Safety Agency, 2021.