Proceedings of the Canadian Society for Mechanical Engineering International Congress 2021
CSME Congress 2021
June 27-30, 2021, Charlottetown, PE, Canada

Design Optimization of Autonomous Steering Control Schemes for Articulated Vehicles

Jiangtao Yu, Tarun Sharma*, Yuping He
Department of Automotive and Mechatronics Engineering, University of Ontario Institute of Technology, Oshawa, Canada
*Corresponding Author, e-mail address: tarun.sharma@ontariotechu.net

Abstract— This paper presents a design synthesis approach for the development of autonomous steering control schemes for articulated vehicles. To design the autonomous steering controller, a 3 degrees of freedom (DOF) yaw-plane model is generated to represent a car-trailer combination, and a model predictive control (MPC) algorithm is used for lateral position and yaw motion control of the articulated vehicle. For enhancing the performance of the self-steering articulated vehicle, the design synthesis of the autonomous driving control schemes is formulated as a design optimization problem. Two optimization algorithms, namely Particle Swarm Optimization (PSO) and Differential Evolution (DE), are introduced and tested for the design optimization. In the design synthesis, the design variables may include passive vehicle design variables, e.g., geometric. To demonstrate the effectiveness of the proposed design synthesis approach, selected simulation results are presented and analyzed. The insightful findings attained from the study may be used as guidelines for developing autonomous driving control systems of articulated vehicles.

Keywords-autonomous driving control; automated steering; articualted vehicles; design optimization; model predictive control; numerical simulation

I. INTRODUCTION

An articulated vehicle consists of a towing unit, e.g., car or tractor, and a towed unit, namely trailer. Compared with single-unit vehicles, such as cars and trucks, articulated vehicles exhibit unique dynamic behaviors, e.g., 'Jack-knife' and 'Fishtailing', which may cause fatal traffic accidents [1-3]. To increase the safety of articulated vehicles, such as car-trailer combinations, attempts have been made to propose, design, and evaluate various active safety systems, including active trailer steering [4-5], trailer differential braking [6-9], active roll control or the coordination of the three systems [10]. These active safety technologies are categorized as 'reactive safety systems' (RSSs), designed to react to the current vehicles state. These systems are effective, but do not consider the effect of driver error [11]. The main cause of traffic accidents is linked to human driver errors. Approximately 93% of severe crashes are due to human driver mistakes [12].

A resolution to the problem is autonomous driving, which removes human from the driving control loop. Since the late 1990s, advanced driver assistance systems, e.g., lane departure prevention, have been developed. These systems are classified as 'predictive safety system' (PSSs) [13]. The last two decades have witnessed extensive research of semi-autonomous vehicles, which are human driven vehicles with autonomous driving capabilities [14]. These vehicles are level 2/3 automated vehicles [15]. To date, the research activities in autonomous driving have mainly been dedicated to passenger cars [11]. Articulated vehicles represent much higher risk than passenger cars in highway operations. However, much less attention has been paid to exploring these PSSs for articulated vehicles [16].

Recently, few studies tackled autonomous driving for articulated construction vehicles [17], and articulated vehicles with automated reverse parking [18]. These autonomous systems were designed only considering low-speed trajectory planning and tracking based on kinematic control, neglecting the high-speed dynamic behaviors of articulated vehicles, e.g., trailer sway, jackknifing, and rollover.

This study proposes a design synthesis approach to the development of automated steering control schemes for articulated vehicles. To design the automated steering controller, a 3-DOF yaw-plane model is generated to represent a car-trailer combination, and a MPC algorithm is used for lateral position and yaw motion control of the car-trailer. The design synthesis of the automated steering control schemes is formulated as a bi-level optimization problem. In the lower level, given a set of design variables, the MPC algorithm determines the desired steering decision to make the vehicle track the target path identified by the navigation system; in the upper level, an optimizer (a search algorithm, such as Particle Swarm Optimization or Differential Evolution) will search the best design variable set in the design space to satisfy the design objective and constraints. Numerical simulation demonstrates the effectiveness of the proposed approach.

The rest of the paper is organized as follows. Section II proposes the bi-level design synthesis method. Section III presents the car-trailer modelling. Section IV introduces the MPC controller. Selected simulation results are analyzed and discussed in Section V. Finally, conclusions are drawn in Section VI.

II. Design Synthesis Approach

The proposed design synthesis approach to the design optimization of autonomous steering control schemes is illustrated in Figure 1. Actually, the proposed approach is a by-level optimization method. At the lower level, based on the data from forward-looking sensors and a higher-level motion planner of the navigation system, motion and path planning is conducted. The predicted path boundaries, which consider various road features and hazard analysis, establish constraints on projected vehicle location and orientation. Given the constraints and the design variable set $\boldsymbol{X}_d$ (including vehicle and controller design variables) from the upper level, the vehicle dynamic model and the MPC algorithm are combined, and an MPC-based optimization problem is formulated. The MPC-based optimization determines a desired sequence of steering inputs. Then, the resulting fitness values and constraints are output to the upper level. The acquired fitness values (i.e., $F_i(\boldsymbol{X}_D)$ and constraints (e.g., $h_i(\boldsymbol{X}_D)$ and $g_i(X_D)$ are treated as a vector optimization problem. By means of a scalarization technique, the vector optimization problem is converted to a scalar optimization problem with a utility function taking the form of $\sum m_i F_i(X_D)$. Note that m_i is a weighting factor. A global search algorithm, such as particle swarm optimization (PSO) or differential evolution (DE), is used as the optimizer to resolve trade-off relations among various design criteria at the upper level, and seeks better design solutions in terms of a new set of design variables $\boldsymbol{X}_d$. The above design process will continue until an optimal design variable set $\boldsymbol{X}_{d\_opt}$ is found. It is expected that given the optimal design variable set $\boldsymbol{X}_{d\_opt}$ determined offline, with the desired steering inputs and tracking the optimal trajectory, the vehicle will be exposed with the minimum threat and the least safety risk.

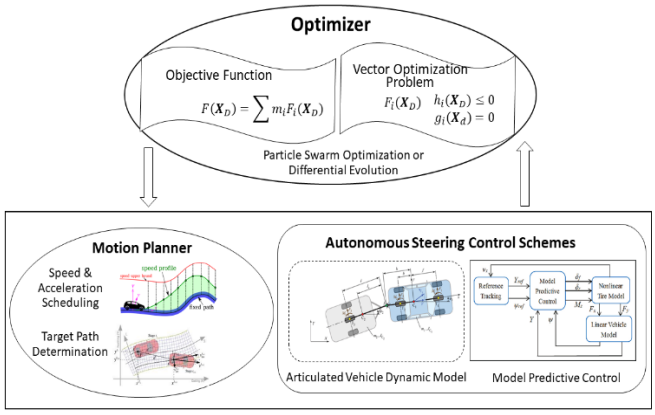

Figure 1. Proposed design synthesis approach to the design optimization of autonomous steering control schemes.

III. Articulated Vehicle Modelling

To design the MPC controller and simulate the lateral dynamics of the car-trailer, a 3-DOF yaw-plane model is generated. Figure 2 shows the single-track car-trailer model considering three motions, including lateral (v) and yaw (r) motions of the car and yaw (r') motion of the trailer. It is assumed that vehicle forward speed (u) is constant, steering angle (δ_f) of car front wheel is small, articulation angle (γ) between the car and trailer is small, the product of variables is small and neglected, and tire cornering forces are the linear function of tire slip angles [19].

Given the above assumptions, the linear governing equations of motion for the car and trailer can be derived as

$$\begin{bmatrix} 1 & 1 & -1 \\ a & -b & d \end{bmatrix} \begin{bmatrix} Y_f \\ Y_r \\ Y_h \end{bmatrix} = \begin{bmatrix} m(\dot{v} + ru) \\ I_{zz}\dot{r} \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & e & -h \end{bmatrix} \begin{bmatrix} X_h \\ Yh \\ Yt \end{bmatrix} = \begin{bmatrix} m'(\dot{u}' + r'v') \\ I'_{zz}\dot{r}' \end{bmatrix} \tag{2}$$

where I_{zz} and I'_{zz} denote the yaw inertia of momentum of the car, respectively, and m and m' the mass of the car and trailer, accordingly, and the rest of symbols are defined and shown in Figure 2.

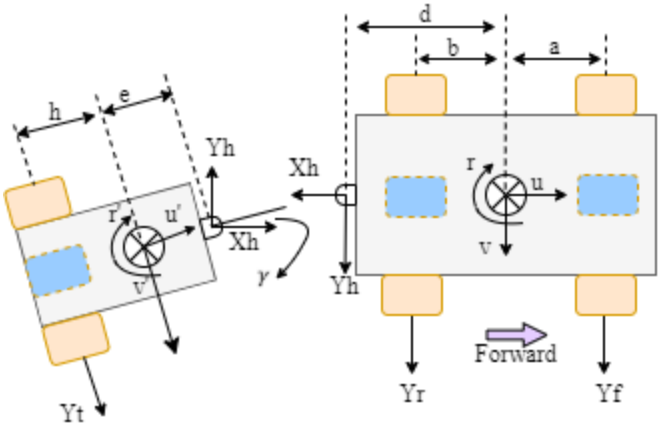

Figure 2. Schematic representation of the 3-DOF linear yaw-plane model.

The tire cornering forces and the respective tire slip angles have the following linear relations,

$$\begin{bmatrix} Y_f \\ Y_r \\ Y_t \end{bmatrix} = -\begin{bmatrix} c_f & 0 & 0 \\ 0 & c_r & 0 \\ 0 & 0 & c_t \end{bmatrix} \begin{bmatrix} \alpha_f \\ \alpha_r \\ \alpha_t \end{bmatrix} \tag{3}$$

where the subscript *f*, *r*, and *t* denote car front axle, car rear axle, and trailer axle, respectively, α represents tire slip angle, and c tire cornering stiffness. The car and trailer are connected by the hitch, and the following kinetic relation holds,

$$v' = u\gamma + (v - dr) - er' \tag{4}$$

Combining Equations (1), (2) and (3), eliminating the reaction forces at the hitch, and considering the kinetic constraint in Equation (4), we have three independent linear governing equations of motion. Then, the 3-DOF linear yaw-plane model can be formulated in the state-space representation as,

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}\dot{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{aligned} \tag{5}$$

where $\boldsymbol{x}$, $\boldsymbol{u}$, and $\boldsymbol{y}$ denote state, control input, and output vector, respectively, matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are defined and offered in Appendix A. The state and input vectors are defined as,

$$\boldsymbol{x} = [\gamma \quad v \quad r \quad r' \quad l \quad l' \quad \varphi \quad \varphi']^T;$$
$$\boldsymbol{u} = [\delta_f] \tag{6}$$

where l and l' represent the lateral position of the car and trailer center of gravity (CG) in the global coordinate system, respectively, and φ and φ' the yaw angle of the car and trailer, accordingly.

The vehicle parameter description and nominal values used in the study are given in Table 1.

Table 1. Vehicle parameter notation and nominal/optimal values

Notation	Nominal value	Lower bound	Upper bound	DE Optimal	PSO Optimal
$m\ /kg$	1,730				
$m'\ /kg$	2,000				
$I_{zz}\ /kgm^2$	3,508				
$I'_{zz}\ /kgm^2$	3,000				
$c_f/N(rad)^{-1}$	80,000				
$c_r/N(rad)^{-1}$	80,000				
$c_t/N(rad)^{-1}$	80,000				
$a\ /m$	1.5	1.35	1.65	1.65	1.6059
$b\ /m$	1.5	1.35	1.65	1.65	1.5805
$d\ /m$	2.7	2.43	2.97	2.7784	2.4452
$e\ /m$	3.0	0	6.0	4.6558	4.1317
$h\ /m$	3.0	0	6.0	1.3442	1.8683

IV. Model Predictive Controller

Model predictive control (MPC) is a feedback control algorithm that uses a model to make predictions about future outputs of a process. MPC uses the model of the plant to make predictions about the future plant output behavior. It also uses an optimizer which ensures that the predicted future plant output tracks the desired reference [20]. After the prediction is made over a finite time, then certain control steps are obtained to minimize the objective performance index. Only the first control step is considered at the current time step and all other control steps are discarded to get predicted state variables at the next time step, the whole process repeats itself in order to get the new control action for the coming time steps. Car trailer model is discretized (matrix A_d is the discretized matrix A and B_d is the discretized matrix B) having sample time Ts = 0.1 is shown below,

$$x_{(k+1)} = A_d x_{(k)} + B_d u_{(k)}; \qquad y_{(k)} = C_{x(k)} \tag{7}$$

1) Performance Index

Performance index is a way to show model predictive control in the form of an optimization problem and the ultimate target of this optimization strategy is to minimize the error between the reference output and the predicted output. J_{MPC} reflects the cost function over a finite prediction horizon (PH) and control horizon (CH), and the details about PH and CH are shown in Appendix.

$$J_{MPC} = \sum_{k=0}^{PH-1} ||y_{d(k)} - y_{(k)}||^2{}_Q + \sum_{k=0}^{CH-1} ||\Delta u_{(k)}||^2{}_R + \sum_{k=0}^{PH-1} ||u_{(k)}||_S^2 \tag{8}$$

Where, $y_{d(k)}$ is the reference output at current time step which is desired lateral position and desired yaw angle, $y_{(k)}$ is the predicted output at current time step which is lateral position and yaw angle, PH is the prediction horizon which is considered at 10-time steps, CH is the control horizon which is considered as 7-time steps. $u_{(k)}$ is the current input which is steering angle of car. Q, R, and S are the weighting factor matrices which shows the importance of control actions. Constraints of steering angle are considered as -0.5 and 0.5, respectively.

2) Reference and Error

As shown in Fig.4, reference defines the desired trajectory which is to be followed by the center of gravity of both the car and trailer. Reference consists of desired values of lateral position of the desired trajectory in global coordinate system. This trajectory is designed based on ISO-14791 standard. Longitudinal speed is considered as constant and the value is equal to 30m/sec. A 2.437 m lane change is designed having a peak lateral acceleration of 2.5m/sec^2 and a time period of 2.5 sec.

V. Experiment Setup

A. Overall simulation diagram

One of the experiment goals is to find the best geometry design ($a, b, d, e, and\ h$) of the car-trailer system, which particularly, are to optimize the geometry dimensions shown in Table.1. Another goal is to find the optimal weighting factors ($m_1, m_2, m_3, and\ m_4$) used in MPC fitness function. The overview co-simulation is presented as the flow chart shown in the Fig3. The detailed Simulink model could be found in the Appendix.

First, the experiment parameters are to be defined by the users, which are mainly defining the population size and termination criteria. The values used in this experiment could be found in Table2, where the termination criteria are defined as number of cost function calls (NFC) is equal to 300.

After setting up the algorithms, the relative parameters are used in PSO or DE, the algorithms are initializing the first generation of population that are defined as 10 in this experiment. All the population are vectors containing all five optimizing geometry variables, which are generated randomly but constrained by the defined limitations.

Each population generated from the algorithms are sent to a vehicle model builder, which build the vehicle dynamics model based on the generated geometries ($a, b, d, e, and\ h$)). Then, vehicle model is sent and used in the MPC controlling algorithm to conduct proper controlling command.

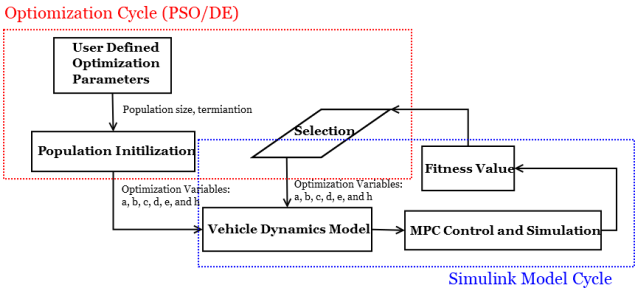

Figure 3 Flow chart of overall simulation diagram. Two main cycles are contained in this process, which are optimization and Simulink dynamic model.

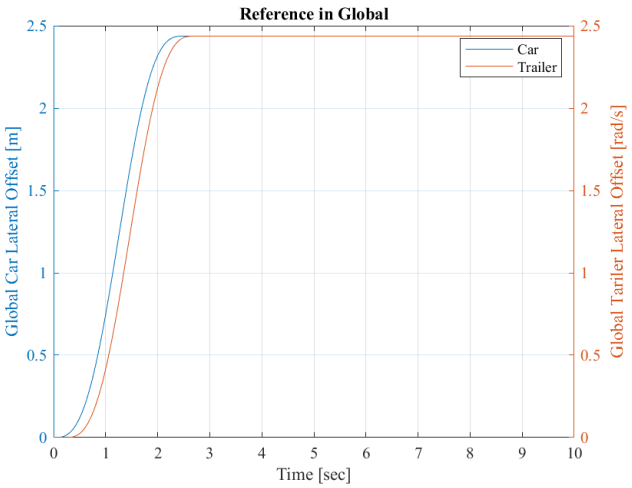

Figure 4 Reference, where the left vertical axis is corresponding to the lateral position of car and the right axis is corresponding to that of trailer. The gap is created due to the distance between the car center of gravity and trailer center of gravity.

The MPC controller is conducting the controlling based on the predefined single-lane-change path as shown in Fig.4. The actual vehicle performance (lateral position, yaw angle) is used to calculate the fitness value for optimization cycle to do selection. Finally, the optimization cycle makes proper mutation and selection based on the fitness value received. Then the next generation of population is generated and send to the vehicle dynamics model builder again to update the old plant model.

Fitness functions for these two algorithms are identical, which is the error between car and trailer lateral positions against reference lateral position plus the error between the car and lateral yaw angles against reference yaw angle. Shown in equation (9), as the units of yaw angle error and the lateral position error are different (rad and meter), standardization of the errors have been conducted, which means the normalized error of each point in the path is equal to the actual error divided by the maximum error among all trajectory points in this path, m_1/m_2, and m_3/m_4 are representing the weighting factors for lateral position, yaw angle errors respectively, and in this experiment, and the range of the parameters are shown in Table2.

Table 2 Optimization Algorithms Parameters Settings

Parameters	**PSO**	**DE**
Population	10	10
Number of Generation	30	30
Termination Criteria	NFC = 300	NFC = 300
Personal Train Weight	0.25	X
Global Train Weight	0.25	X
Inertia Weighting	0.5 to 0.2	X
Crossover Possibility	X	0.8
Weight 1	5 to 10	5 to 10
Weight 2	0 to 10	0 to 10
Weight 3	0 to 5	0 to 5
Weight 4	0 to 5	0 to 5
Number of Runs	2	2

$$F_{(l_i,\,\varphi_i,\,l_i')} = Mean.\left[\sum_{i=1}^{i=k} m_1\left(\frac{l_i - l_{ref_i}}{MAX(l - l_{ref})}\right)^2 + m_2\left(\frac{\varphi_i - \varphi_{ref_i}}{MAX(\varphi - \varphi_{ref})}\right)^2 + m_3\left(\frac{l'_i - l_{ref_i}}{MAX(l' - l_{ref})}\right)^2 + m_4\left(\frac{\varphi'_i - \varphi_{ref_i}}{MAX(\varphi' - \varphi_{ref})}\right)^2\right] \quad (9)$$

B. *Two optimization algorithms*

Both PSO and DE are popular algorithms for optimization tasks, but one of the major differences between DE and PSO is in the mechanism to produce a new population of solutions via perturbation of solutions from the old population. As discussed in [21], the population diversification of DE is better than PSO because the best solution in the population is independent of the other solutions in the population. Another comparison between DE and PSO is the clustering of particles. PSO has a higher chance to cluster rapidly and the swarm may quickly become stagnant but clustering in DE is least and re-initialization has the least effect for DE.

1) *Differential Evolution*

Differential evolution (DE) is a population-based meta-heuristic search algorithm which optimizes a problem by iteratively improving a candidate solution based on an evolutionary process. Such algorithms make few or no assumptions about the underlying optimization problem and can quickly explore very large design spaces. In differential Evolution, each solution is known as Chromosome. Each chromosome undergoes mutation followed by recombination. A target vector is the solution which undergoes evolution. Target vector is used in mutation to generate the donor vector and the donor vector undergoes recombination to obtain that trial vector. A greedy selection is employed between target vector and trial vector and the better solution among these two vectors survives for the next generation. The selection of better solutions is performed only after the generation of all trial vectors. Donor Vector (V) of a chromosome (Xi) is created as:

$$V = X_{r_1} + F(X_{r_2} - X_{r_3}) \quad (10)$$

Where F is the mutation constant and r_1, r_2, r_3 are the three random solutions from the population. Trial Vector is created as:

$$k' = \begin{cases} h^j & if \quad n \le p_c \; OR \; j = \delta \\ m^j & if \; n > p_c \; \; AND \; \; j \ne \delta \end{cases} \quad (11)$$

Where p_c is the crossover probability, n is a random number between 0 and 1 and δ is randomly selected variable location which ensures that at least one variable is obtained from the donor vector.

Differential Evolution is implemented to minimize the difference between the resultant lateral position of the car trailer and the reference lateral position. The population size is kept as 10. The dimension of the problem is 5 and the generation was set as 20. Crossover possibility and scale factor were set as 0.8 and 0.8 respectively. More detail about DE could be found form the DE flow chart in the Appendix.

2) *Particle Swarm Optimization*

Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart in 1995, which is one of the stochastic population-based metaheuristics inspired from swarm intelligence [22]. This algorithm mimics social behavior like bird flocking and fish schooling. The first step in Particle swarm optimization is to initialize the position and velocity randomly within the search space and initial velocities are set as zero. Fitness function (shown in equation 9) called to find the path error, and if the error is less than the former, the personal best record is updated, if not, the personal best will not be updated. Each personal best value is compared within this generation in order to find the global best and record it. In the next generation, velocities will be updated based on the global best and personal best from the former generation.

As a result, two training weights (personal and global) were defined as 0.25 (as shown in Table. 2), inertia weighting was set decaying from 0.5 at the first generation to 0.2 at the last generation, since the generation increases the population is getting closer to the optimized solution and the velocity of search should be reduced. Besides, regular parameters like population size, termination criteria and number of runs were specified as in Table.2 and identical with DE algorithm. More detail about PSO could be found form the PSO flow chart in the Appendix.

VI. EXPERIMENT RESULTS

A. *Lateral Position*

The first simulation result is the car and trailer trajectories, where the car and trailer should follow with the reference lateral position. Therefore, it is important to find the difference between the lateral position reference and the lateral position of car's mass center and trailer's mass center. The results of car and trailer are shown in fig.5 and fig.6, respectively. Both the optimized results from PSO and DE are shown and compared with the nominal design performance. First, it is obvious that the PSO-optimized design has the largest delay against the reference lateral position. The optimized geometries are listed in the Table. 3, where the dimension e of the PSO-optimized design is the largest, which means the trailer center of gravity is closest to the rea end. This is the reason leading to this large lateral position delay. It is obvious in the plot of trailer trajectories, DE-optimized design has efficiently decreased the deviation of the trailer from the target path, where the peak offset is lower than the nominal design. It can be realized that DE algorithm has moved the car center of gravity forward compared to the nominal design and increased the controlling weight on the trailer path-following error, which is the reason to the performance improvement.

B. *Lateral Acceleration*

The lateral acceleration of single lane change maneuver should be a single cycle of sine wave. The following results (Fig.7 and Fig.8) are presenting the lateral acceleration histories of both car and trailer. Three designs (PSO, DE, and Nominal) are plotted together for convenience of comparison.

Identify applicable sponsor/s here. *(sponsors)*

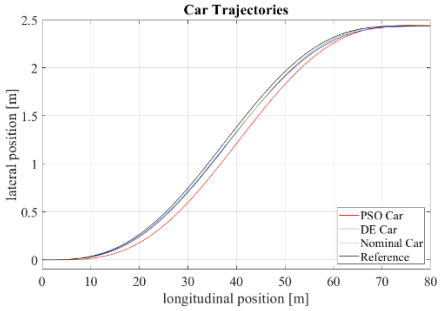

Figure 5, car trajectories, where the target path is represented by the black color. DE-optimized design (blue) slightly reduced the error between trajectory and target path. PSO-optimized design has a larger error compared to DE and nominal designs.

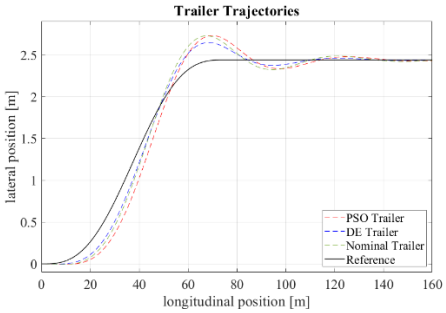

Figure 6 trailer trajectories, the color legends here are the same as the ones used in fig.4. One could find the DE-optimized design reduced the response delay compared to the nominal design, and the peak value (overshoot) has been reduced as well.

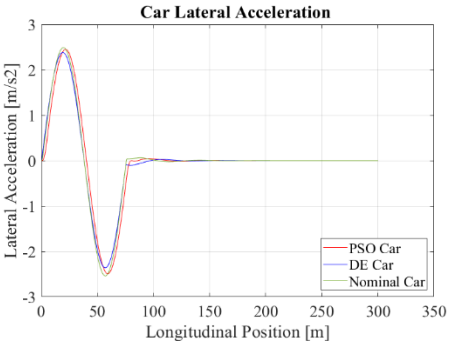

Figure 7 Car lateral acceleration. The PSO-optimized design has similar peak lateral acceleration as nominal design. The DE (blue) has lower peak lateral acceleration than that of the other two designs, which illustrates the lower offset shown in trajectories plots.

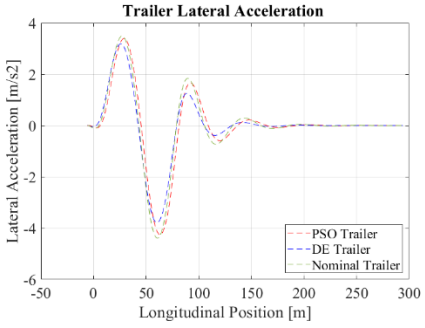

Figure 8 Trailer lateral acceleration. More obvious improvement could be found in this plot regarding DE optimized design. Although all trailer peak lateral accelerations are higher than that of cars, the DE still achieves the lowest peak trailer lateral acceleration.

Table 3 Lateral Acceleration Peak Value

	Car		Trailer	
	Peak	Improvement	Peak	Improvement
Nominal	2.539	0	4.39	0
PSO	2.477	2.40%	4.233	3.58%
DE	2.36	8.98%	3.787	13.74%

The peak lateral accelerations of both car and trailer are summarized in the Table.3. It shows that both PSO and DE have successfully improved the lateral stability compared to nominal design. Especially for the trailer, PSO has reduced the peak lateral acceleration by 3.58% and the DE has reduced by 13.74% compared to the nominal design.

VII. Discussion

A. *Optimization Process*

As discussed in the experiment design section, the population of both algorithms are set as 10, and the termination criteria is the maximum number of function calls equaling to 300 (30 generations). The fitness logs of both algorithms have been shown in Fig.9. PSO has shown a relative higher fitness value compared to the DE, while both algorithms were reaching to the optimized design after about 15 generations.

B. *Optimized design summary*

It has been shown in the Table. 4 that DE have achieved optimal fitness value at 0.1911 while the nominal design is 0.2211. For e and h, both PSO and DE have reduced h and increased e, DE has moved the car center of gravity forward compared with nominal and PSO designs.

VIII. Conclusion

This paper presents a design synthesis approach to the development of autonomous steering control schemes for articulated vehicles. To examine the effectiveness of the proposed design synthesis approach, numerical simulation is conducted. The co-simulation combines optimization algorithms, vehicle dynamic modeling, and controller designing in order to analyze the influence from geometry design on the car-trailer system path-following performance, where the performance is evaluated based on the lateral position error and yaw angle error.

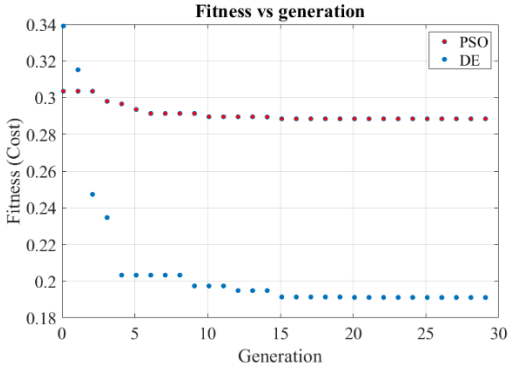

Figure 9 Fitness value log. This plot directly compared the fitness history of PSO and DE optimization algorithm. It is obvious that compared with DE, PSO presents poorer optimization results, and its fitness was trapped around 0.29. However, the DE has further reduced the fitness value to about 0.19, which is also identical to the results shown in trajectories and lateral acceleration plots.

Table 4 Optimized results summary

	Nominal	**PSO**	**DE**
e_trailer	3	3.6668	3.433
h_tariler	3	2.3332	2.567
dimension_a	1.5	1.4536	1.35
dimension_b	1.5	1.5408	1.65
dimension_d	2.7	2.6409	2.43
Weight1	10	6.8858	6.1817
Weight2	5	2.8492	4.3304
Weight3	0	0.3798	0
Weight4	0.1	0.2643	0
max car lateral position error [m]	0.1407	0.1791	0.0527
max car yaw angle error [rad]	0.0686	0.0652	0.0615
max trailer lateral position error [m]	0.3028	0.4095	0.2292
max trailer yaw angle error [rad]	0.0183	0.0449	0.0353
average car lateral position error [m]	0.0153	0.0218	0.0066
average car yaw angle error [rad]	0	0	0
average trailer lateral position error [m]	0.0175	0.0211	0.0046
average trailer yaw angle error [rad]	0	0	0
Fitness	0.2211	0.2901	0.1911

Verified car-trailer model was implemented in Simulink, which was accompanied with hand-tuning model predictive controller. The co-simulation has been designed as a whole simulation/optimization cycle, and the process experiment data was collected from 30 generations of optimization and 10 individuals in each generation. However, the results have shown that the two algorithms successfully achieved the designing optimization task after 15 generations. The DE algorithm successfully generate the geometry of the car-trailer system that could minimize the path-following error during a single lane change maneuver (normalized error decreased from 0.2211 to 0.1911).

Comparison between these two algorithms were conducted and the results were shown in this article as well. The results have shown that for this 9-dimension optimization problem, DE outperforms PSO in level of error.

IX. Futuren work

This paper presents the preliminary results in developing the design synthesis approach, where car-trailer system could be designed by the optimization algorithms in order to improve the safety features and ability to stay stable under various maneuvers. The next step of this project will be focusing on adding more design parameters of car-trailer system as the optimization variables such as mass, height, and wheelbase, so the algorithms could optimize at a larger picture. The future work of this project could be using a better vehicle model showing more details. For example, including the roll motion of the car-trailer system, which is important for emergent situation as well. Besides, safety issue 'Fishtailing' and 'Jack-knife' should be considered as the constraints of the optimization target to avoid these situations or reduce the possibility of these two dangerous behaviors.

X. REFERENCES

[1] S. Vempaty, Y. He, and L. Zhao, "An overview of control scheme for improving the lateral stability of car-trailer combinations," Int. J. Vehicle Performance, Vol. 6, No. 2, pp. 151-199, 2020.

[2] Y. He, H. Elmaraghy, and W. Elmaraghy, "A design analysis approach for improving the stability of dynamic systems with application to the design of car-trailer systems," Journal of Vibration and Control, Vol. 11, No. 12, pp. 1487-1509, 2005.

[3] Y. He, and J. Ren, "A comparative study of car-trailer dynamics modelts," SAE Int. J. Passeng. Cars-mech. Syst. Vol. 6, No. 1, 2013, doi:10.4271/2013-01-0695.

[4] S. Vempaty, E. Lee, and Y. He, "Model-reference based adaptive control for enhancing lateral stability of car-trailer systems," Proceedings of ASME International Mechanical Engineering Congress and Exposition, Paper No: INECE2016-65090, V012T16021; 8 pages, 2016.

[5] S. Vempaty, T. Sun, and Y. He, "Enhanced lateral stability of car-trailer systems using model reference adaptive control," Dynamics of Vehicles on Roads and Tracks Vol 1: Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017), 14-18 August 2017, Rockhampton, Queensland, Australia.

[6] E. Lee, S. Kapoor, T. Sikder, and Y. He, "An optimal robust controller for active trailer differential braking systems of car-trailer combinations," Int. J. Vehicle Systems Modelling and Testing, Vol. 12, Nos. ½, pp. 72-93, 2017.

[7] T. Sun, E. Lee, and Y. He, "Non-linear bifurecation stability analysis for articulated vehicles with active trailer differential braking systems," SAE Int. J. Mater. Manf., Vol. 9, No. 6, doi:10.4271/2016-01-0433, 2016.

[8] T. Sun, Y. He, and J. Ren, "Dynamics analysis of car-trailer systems with active trailer differential braking strategies," SAE Int. J. Passeng. Cars – Mech. Syst., Vol. 7, No. 1, doi:10.4271/2014-01-0143, 2014.

[9] X. Ding, Y. He, J. Ren and T. Sun, "A comparative study of control algorithms for active trailer steering systems of articulated heavy vehicles," *2012 American Control Conference (ACC)*, Montreal, QC, Canada, 2012, pp. 3617-3622, doi: 10.1109/ACC.2012.6315139.

[10] Y. He, M. M. Islam, S. Zhu, and T. Hu, "A design synthesis framework for directional performance optimization of multi-trailer articulated heavy vehicles with trailer lateral dynamic control systems," Proc IMechE Part D: J Automobile Engineering 2017, Vol. 231(8) 1096–1125.

[11] A. Rahimi, and Y. He, "A review of essential technologies for autonomous and semi-autonomous articulated heavy vehicles," Proceedings of the Canadian Society for Mechanical Engineering International Congress 2020 (CSME Congress 2020), June 21-24, 2020, Charlettetown, PE, Canada, 8 pages.

[12] "U.S. Federal Motor Carrier Safety Administration. Large Trucks and bus crash facts 2016." [Online]. Available: https://cms8.fmcsa.dot.gov/safety/data-and-statistics/large-truck-and-bus-crash-facts-2016#A8.

[13] S. J. Anderson, J. M. Walker, S. B. Karumanchi, et al. "The intelligent copilot: A constraint-based approach to shared-adaptive of ground vehicles," IEEE Trans Intell Transp Syst Magaz. 2013;5(2):45-54.

[14] A. Gray, Y. Gao, J. K. Hedrick, et al. "Robust predictive control for semi-autonomous vehicles with an uncertain driver model," IEEE Intelligent Vehicles Symposium (IV); 2013; Gold Coast, Australia.

[15] SAE International. Surface vehicle recommended practice J3016: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Society of Automotive Engineering; 2014.

[16] N. Andrén, A. G. Martín, et al. "Predictive control for autonomous articulated vehicles." Bachelor thesis, Department of Computer Science, Chalmers University of Technology, Sweden; 2017.

[17] B. J. Alshaer, T. T. Darabseh, M. A. Alhanouti, "Path planning, modeling and simulation of an autonomous articulated heavy construction machine performing a loading cycle," Applied Mathematical Modelling, 2012;37:5315-5325.

[18] A. Elhassan, "Autonomous driving system for reversing an articulated vehicle," Master's thesis, Department of Automatic Control, the Royal Institute of Technology, Stockholm, Sweden; 2015.

[19] L. Zhao, and Y. He, "An investigation of active safety control strategies for improving the lateral stability of car-trailer systems," Int. J. Vehicle Systems Modelling and Testing, Vol. 13, No. 4, pp. 295-318.

[20] Abroshan, M., Hajiloo, R., Hashemi, E., & Khajepour, A. (2020). Model predictive-based tractor-trailer stabilisation using differential braking with experimental verification. Vehicle System Dynamics, 1–24. https://doi.org/10.1080/00423114.2020.1744024

[21] Goudarzi, S., Hassan, W., Anisi, M., & Soleymani, S. (2016). Comparison between hybridized algorithm of GA–SA and ABC, GA, DE and PSO for vertical-handover in heterogeneous wireless networks. Sadhana (Bangalore), 41(7), 727–753. https://doi.org/10.1007/s12046-016-0509-4

[22] Deb, A., Roy, J., & Gupta, B. (2014). Performance Comparison of Differential Evolution, Particle Swarm Optimization and Genetic Algorithm in the Design of Circularly Polarized Microstrip Antennas. IEEE Transactions on Antennas and Propagation, 62(8), 3920–3928. https://doi.org/10.1109/TAP.2014.2322880

XI. APPENDIX

A. State Space Representation Matrices Defination

$$\boldsymbol{A} = \begin{bmatrix} -inv(\boldsymbol{M}) * \boldsymbol{L} & 0 & 0 & 0 & 0 \\ 0,1,0,0 & 0 & 0 & u & 0 \\ u,1,-d,-e & 0 & 0 & 0 & u \\ 0,0,1,0 & 0 & 0 & 0 & 0 \\ 0,0,0,1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} inv(\boldsymbol{M})\boldsymbol{F} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & m+m' & -m'd & -m'e \\ 0 & -m'd & I_{zz}+m'd^2 & m'ed \\ 0 & -m'e & m'ed & I'_{zz}+m'e^2 \end{bmatrix}$$

$$\boldsymbol{L} = \frac{1}{u}\begin{bmatrix} 0 & 0 & -u & u \\ uc_t & c_f+c_r+c_t & (m+m')u^2+ac_f-bc_r-dc_t & -(e+h)c_t \\ -duc_t & ac_f-bc_r-dc_t & -m'du^2+a^2c_f+b^2c_r+d^2c_t & (e+h)dc_t \\ -(e+h)uc_t & -(e+h)c_t & -m'eu^2+(e+h)dc_t & (e+h)^2c_t \end{bmatrix}$$

$$\boldsymbol{F} = \begin{bmatrix} 0 \\ c_f \\ ac_f \\ 0 \end{bmatrix}$$

B. *Prediction Horizon and Control Horizon*

Explained in the following figure, Prediction Horizon is a parameter which shows how far a controller can predict the future. If it is too big, the controller will not be able to control the incidents that will happen during the period which lies in the prediction horizon time and if it is too small, the controller will not be able to cover a safe predicted distance which is required to respond to the coming hurdles instantly at high speeds.

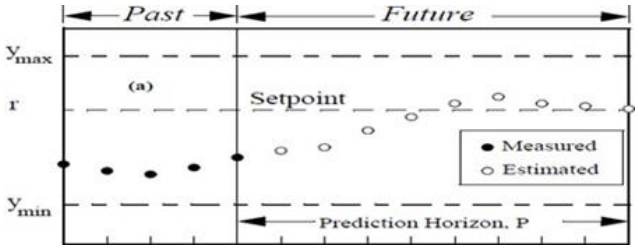

Figure 1 Prediction Horizon Illustration

As shown in following figure CH is the number of time steps which are controlled by the controller or the total computations run by the controller. Rest of the time steps which are only predicted and do not fall in the controlled horizon are considered as constant because they do not have a significant effect on the output Recommended Control Horizon is between 10 to 20 percent of the prediction Horizon.

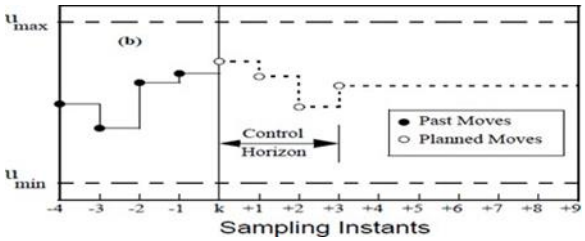

Figure 2 Control Horizon Illustration

C. *PSO Flowchart*

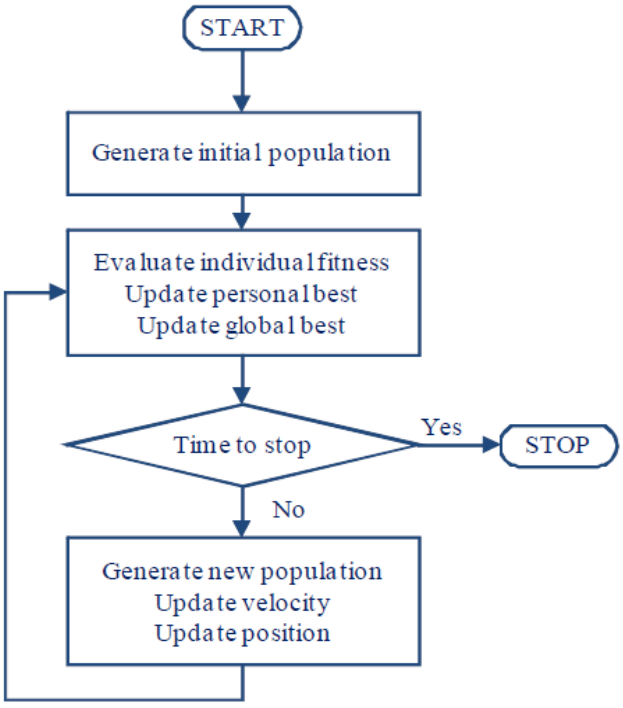

Figure 3 PSO algorithm flowchart

D. DE Flowchart

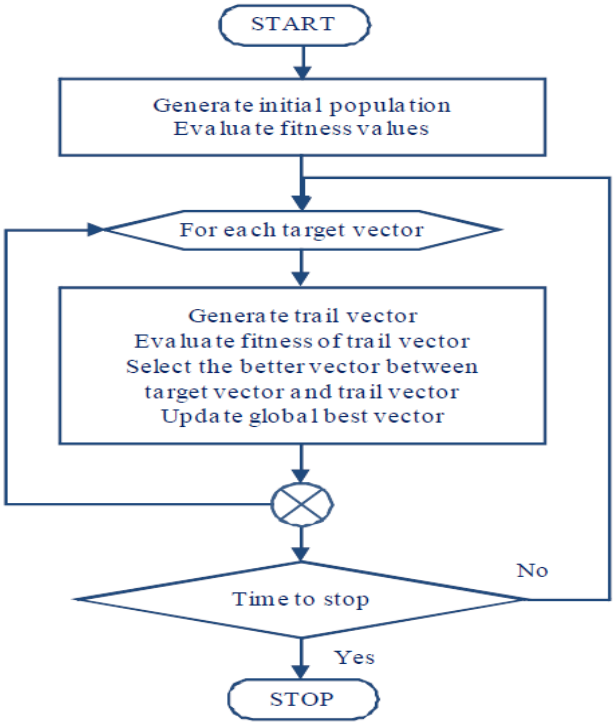

Figure 4 DE algorithm flowchart.

E. Simulink Diagram

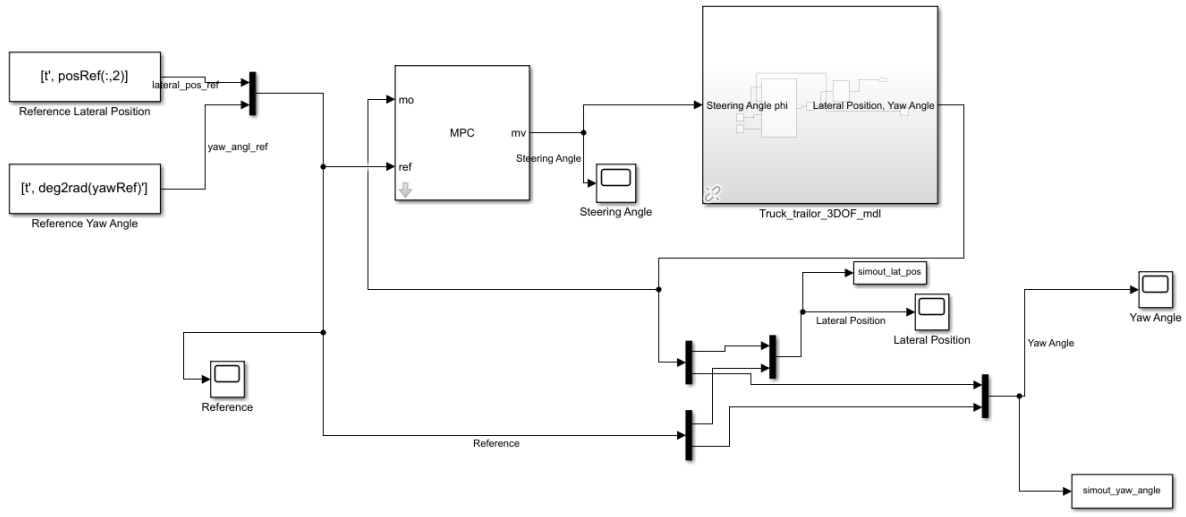

Figure 5 Screenshot of the Simulink model.