Robust Controller Design for Active Trailer Steering Systems of Articulated Vehicles Using Multi-objective

by
Khizar Qureshi

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Faculty of Engineering and Applied Science
University of Ontario Institute of Technology
Oshawa, Ontario, Canada

August 2019

THESIS EXAMINATION INFORMATION

Submitted by: **Khizar Ahmad Qureshi**

Masters of Applied Science in Electrical and Computer Engineering

Thesis Title: Robust Controller Design for Active Trailer Steering System of Articulated Vehicles Using Multi-Objective Optimization

An oral defense of this thesis took place on August 6, 2019 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Shahryar Rahnamayan
Research Supervisor	Dr. Ramiro Liscano
Research Co Supervisor	Dr. Yuping He
Examining Committee Member	Dr. Akramul Azim
External Examiner	Dr. Jaho Seo

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

STATEMENT OF CONTRIBUTIONS

Part of this work has been published in conferences as,

1. Mutaz Keldani, Khizar Qureshi, Yuping He, Ramiro Liscano. Design Optimization of a Robust Active Trailer Steering System for Car-Trailer Combinations. Society of Automotive Engineers (SAE) WCX Congress 2019

2. Mohammed Hammad, Khizar Qureshi, Yuping He. Safety and Lateral Dynamics Improvement of a Race Car Using Active Rear Wing Control. Society of Automotive Engineers (SAE) WCX Congress 2019.

3. Khizar Qureshi, Shahryar Rahnamayan, Yuping He, Ramiro Liscano. Enhancing LQR Controller Using Optimized Real-time System by GDE3 and NSGA-II Algorithms and Comparing with Conventional Method. IEEE Congress on Evolutionary Computation 2019.

ABSTRACT

This thesis presents and evaluates an approach to the robust controller design for active trailer steering (ATS) systems to increase the safety of articulated vehicles. By applying a multi-objective evolutionary algorithm (MOEA) to the design optimization of the robust ATS controller, a series of optimal gain values can be obtained in a single run. This allows for posteriori decision making along with flexibility to select appropriate gain for different operating conditions. The algorithm creates Pareto optimal gain values for various speeds, thereby resulting in the robust ATS controller with an optimized gain scheduling scheme. The research elucidates the advantages of multi-objective algorithms over mono-objective or single-objective algorithms. For the design optimization of the ATS controller, a benchmark investigation is conducted to select an effective algorithm from the multi-objective algorithms, including GDE3, NSGA-II, NSGA-III, SPEA2 and MOPSO. A modular framework is introduced for co-simulations conducted in the CarSim-Simulink/Matlab environment, with which the vehicle and controller parameters can be optimized. The method ensures that a robust ATS controller with optimized feedback control gains, as well as satisfaction of design criteria and constraints. This research proposes a framework to generate a multi-dimensional look-up table using the multi-objective evolutionary algorithm for a general dynamic system controlled by a feedback controller. The optimized look-up system can be used to improve the robustness of control systems in real-world applications.

AUTHORS DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Khizar Ahmad Qureshi

ACKNOWLEDGEMENTS

I want to extend my utmost gratitude to Dr. Ramiro Liscano and Dr. Yuping He, for giving me the opportunity to pursue my masters at UOIT. You ensured I was on the right track with my research and provided me with continued financial support to guarantee I could concentrate completely on my studies. It has been an absolute privilege to work with you both. In the ever changing path of life, the constant, for me, will be my appreciation and remembrance of you two. Thank You.

I want to make sure I thank my wonderful wife, Ayesha, who supported me through my education. She always believed in me and created a wonderful environment at home to ensure success. She put her life on hold to push me forward, and for that I would be eternally grateful. My life would be incomplete without you.

Contents

Thesis Examination Information i

Statement of Contributions ii

Abstract iii

AUTHORS DECLARATION iv

Acknowledgements v

List of Figures ix

List of Tables xiii

Abbreviations xiv

1 Introduction 1
- 1.1 Problem with Conventional Tuning Methods 2
  - 1.1.1 Motivation 2
- 1.2 Problem Statement 4
- 1.3 Thesis Contributions and Novelty 5
  - 1.3.1 MOEA for ATS Controllers 5
  - 1.3.2 MOEA Optimized Gain Scheduled ATS Controllers . . . 6
  - 1.3.3 Modular Approach for Design Optimization of Control Systems Using CarSim-MATLAB/Simulink Co-Simulations 6
  - 1.3.4 Adaptive Driver Model with Varying Reaction Time and Vehicle Forward Speed 7
  - 1.3.5 Performance Analysis of MOEAs for Control System Tuning 7
- 1.4 Thesis Organization 7

2 Literature Review 10
- 2.1 Evolutionary Algorithms Applied to Control Systems 10
- 2.2 Linear Quadratic Regulator Technique 11

2.2.1 Design Optimization of LQR Controller 12
2.2.2 Difference in Approach 13
2.3 Unstable Modes of Car-Trailer Combinations 14
2.4 Stabilizing Articulated Vehicles 15
2.4.1 Passive Steering System 16
2.4.2 ATS Systems 16
2.4.3 Multi-Mode Steering Control 17
2.4.4 Differential Braking 17
2.4.5 Teslas Approach to Jackknifing Prevention 18
2.5 Differential Evolution 18
2.6 Optimal Pareto-front 20
2.7 Knee-point 21
2.8 Non-dominated Sorting Genetic Algorithm 21
2.9 Generalized Differential Evolution 23
2.9.1 GDE3 Algorithm 24

3 Testing the Base Algorithm: Differential Evolution 26
3.1 PID Controller 27
3.2 Tuning methods for PID controllers 28
3.3 DE Algorithm 30
3.3.1 Island distribution of DE 31
3.4 Objectives 33
3.5 Steady State Error 34
3.6 Second Order System 35
3.7 Performance Comparison 35
3.7.1 Results and Analysis 36
3.8 Convergence Speed Comparison 39
3.8.1 Results and Analysis 40
3.9 Summary 40

4 GDE3 and NSGA-II Comparison on LQR Based Aircraft Pitch Control 42
4.1 Introduction 43
4.2 System Model 44
4.3 Multiple Objectives 45
4.4 Ensuring Stability and Controllability 47
4.5 Optimization Parameters and Pre-compensation 48
4.6 Complexity of Case Study 49
4.7 LQR Controller for Aircraft Pitch Control 49
4.7.1 Experimental Results 51
4.7.2 Analysis 53
4.8 Summary 54

5 Design Optimization of LQR-based ATS Controller Using GDE3 56
5.1 Introduction 56
5.2 Rearward Amplification (RWA) 57

5.3 Path-Following Off-Tracking 59
5.4 Design ATS Controller 59
5.4.1 3-DOF Yaw-plane Car-trailer Model 60
5.5 Single-lane Change Testing Maneuver 63
5.6 Testing Methodology 64
5.7 Results and Discussions 65
5.7.1 Critical Speed Analysis 65
5.8 Forward Speed Variation 66
5.9 Summary 70

6 Design Optimization of Gain Scheduling Controllers Using GDE3 and Closed-loop Co-Simulation 72
6.1 Introduction 73
6.2 CarSim Model and Methodology 74
6.2.1 Built-in Driver Model Offered from CarSim 75
6.2.2 Evaluation Criteria 77
6.3 Gain Scheduling Controller 77
6.4 Modular Design Methodology 79
6.5 Overcoming System Limitation with Innovization 81
6.6 Algorithm Comparison 83
6.7 Results: GDE3 Generated GSC vs Passive 85
6.8 Results: Gain Scheduling 92
6.9 Important Observation 96

7 Conclusions 98
7.1 Contributions 98
7.2 Recommendations for Future Studies 99

Bibliography 101

List of Figures

2.1 Point B is the knee-point. This image is taken from [51] 22

3.1 Responses for DE1 and MATLAB Tuner and Random Value . . . 38
3.2 Responses for DE2 and MATLAB Tuner and Random Value . . . 39

4.1 Feedback system model with LQR and decision support system. Open-loop plant is the aircraft pitch model. Controller gains are obtained by the algorithm, which can be dynamically changed by the decision system based on environmental factors. 46
4.2 Feedback system model for aircraft pitch control with GDE3 . . 51
4.3 Pareto-front obtained by GDE3 and NSGA-II on aircraft pitch con- trol model. (Note: The designer can choose any LQR gain matrix value from the Pareto-front, based on ST and RT requirements.) 52
4.4 Response of the aircraft pitch control system with ST prioritized. 52
4.5 Response of the aircraft pitch control system with RT prioritized. 52
4.6 Response of the aircraft pitch control system with knee-point. . 53
4.7 Response of the aircraft pitch control system obtained from [66]. 53

5.1 Schematic representation of the 3-DOF yaw-plane car-trailer model 61
5.2 A representation of an SLC steering input 64
5.3 Lateral acceleration of the car at the critical speed 79.2km/h (without ATS controller) 66
5.4 Lateral acceleration of the trailer at the critical speed 79.2km/h (without ATS controller) 66
5.5 Optimal Pareto-front at 80km/h 67
5.6 Optimal Pareto-front at 90km/h 67
5.7 Optimal Pareto-front at 100km/h 67
5.8 Optimal Pareto-front at 110km/h 67
5.9 Lateral acceleration of the car at 80km/h 68
5.10 Lateral acceleration of the trailer at 80km/h 68
5.11 Lateral acceleration of the car at 90km/h 68
5.12 Lateral acceleration of the trailer at 90km/h 68
5.13 Lateral acceleration of the car at 100km/h 68
5.14 Lateral acceleration of the trailer at 100km/h 68
5.15 Lateral acceleration of the car at 110km/h 69
5.16 Lateral acceleration of the trailer at 110km/h 69

6.1 Car sub-model developed in CarSim 74
6.2 Trailer sub-model developed in CarSim 75
6.3 Car-trailer model with the built-in driver model. 75
6.4 Predefined trajectory for the closed-loop single lane-change testing maneuver 76
6.5 System model using modular blocks. State Information from top to bottom: Yaw-rate of car, Yaw-rate of trailer, lateral speed of car and lateral speed of trailer. 80
6.6 Algorithm comparison based on two design PFOT. 84
6.7 Trajectory of Car at 120km/h for Algorithm Comparison 85
6.8 Trajectory of Trailer at 120km/h for Algorithm Comparison . . . 85
6.9 Lateral Acceleration of Car at 120km/h for Algorithm Comparison 85
6.10 Lateral Acceleration of Trailer at 120km/h for Algorithm Comparison 85
6.11 Optimal Pareto-front for 80 km/h and 0(s) driver model reaction time 86
6.12 Optimal Pareto-front for 90 km/h and 0(s) driver model reaction time 86
6.13 Optimal Pareto-front for 100 km/h and 0(s) driver model reaction time 86
6.14 Optimal Pareto-front for 110 km/h and 0(s) driver model reaction time 86
6.15 Optimal Pareto-front for 120 km/h and 0(s) driver model reaction time 87
6.16 Optimal Pareto-front for 80 km/h and 0.1(s) driver model reaction time 87
6.17 Optimal Pareto-front for 90 km/h and 0.1(s) driver model reaction time 87
6.18 Optimal Pareto-front for 100 km/h and 0.1(s) driver model reaction time 87
6.19 Optimal Pareto-front for 110 km/h and 0.1(s) driver model reaction time 87
6.20 Optimal Pareto-front for 120 km/h and 0.1(s) driver model reaction time 87
6.21 Trajectory of the car at 80 km/h and 0(s) driver model reaction time. 88
6.22 Trajectory of the trailer at 80 km/h and 0(s) driver model reaction time. 88
6.23 Trajectory of the car at 90 km/h and 0(s) driver model reaction time. 88
6.24 Trajectory of the trailer at 90 km/h and 0(s) driver model reaction time. 88
6.25 Trajectory of the car at 100 km/h and 0(s) driver model reaction time. 89

6.26 Trajectory of the trailer at 100 km/h and 0(s) driver model reaction time. 89

6.27 Trajectory of the car at 110 km/h and 0(s) driver model reaction time. 89

6.28 Trajectory of the trailer at 110 km/h and 0(s) driver model reaction time. 89

6.29 Trajectory of the car at 120 km/h and 0(s) driver model reaction time. 89

6.30 Trajectory of the trailer at 120 km/h and 0(s) driver model reaction time. 89

6.31 Trajectory of the car at 80 km/h and 0.1(s) driver model reaction time. 90

6.32 Trajectory of the trailer at 80 km/h and 0.1(s) driver model reaction time. 90

6.33 Trajectory of the car at 90 km/h and 0.1(s) driver model reaction time. 90

6.34 Trajectory of the trailer at 90 km/h and 0.1(s) driver model reaction time. 90

6.35 Trajectory of the car at 100 km/h and 0.1(s) driver model reaction time. 90

6.36 Trajectory of the trailer at 100 km/h and 0.1(s) driver model reaction time. 90

6.37 Trajectory of the car at 110 km/h and 0.1(s) driver model reaction time. 91

6.38 Trajectory of the trailer at 110 km/h and 0.1(s) driver model reaction time. 91

6.39 Trajectory of the car at 120 km/h and 0.1(s) driver model reaction time. 91

6.40 Trajectory of the trailer at 120 km/h and 0.1(s) driver model reaction time. 91

6.41 GSC Trajectory of the car at 80 km/h and 0.0(s) driver model reaction time. 92

6.42 GSC Trajectory of the trailer at 80 km/h and 0.0(s) driver model reaction time. 92

6.43 GSC Trajectory of the car at 90 km/h and 0.0(s) driver model reaction time. 92

6.44 GSC Trajectory of the trailer at 90 km/h and 0.0(s) driver model reaction time. 92

6.45 GSC Trajectory of the car at 110 km/h and 0.0(s) driver model reaction time. 93

6.46 GSC Trajectory of the trailer at 110 km/h and 0.0(s) driver model reaction time. 93

6.47 GSC Trajectory of the car at 120 km/h and 0.0(s) driver model reaction time. 93

6.48 GSC Trajectory of the trailer at 120 km/h and 0.0(s) driver model reaction time. 93
6.49 GSC Trajectory of the car at 80 km/h and 0.1(s) driver model reaction time. 93
6.50 GSC Trajectory of the trailer at 80 km/h and 0.1(s) driver model reaction time. 93
6.51 GSC Trajectory of the car at 90 km/h and 0.1(s) driver model reaction time. 94
6.52 GSC Trajectory of the trailer at 90 km/h and 0.1(s) driver model reaction time. 94
6.53 GSC Trajectory of the car at 100 km/h and 0.1(s) driver model reaction time. 94
6.54 GSC Trajectory of the trailer at 100 km/h and 0.1(s) driver model reaction time. 94
6.55 GSC Trajectory of the car at 110 km/h and 0.1(s) driver model reaction time. 94
6.56 GSC Trajectory of the trailer at 110 km/h and 0.1(s) driver model reaction time. 94
6.57 GSC Trajectory of the car at 120 km/h and 0.1(s) driver model reaction time. 95
6.58 GSC Trajectory of the trailer at 120 km/h and 0.1(s) driver model reaction time. 95

List of Tables

3.1 The table lists the assigned labels of the variants of DE based on mutation schemes. 32
3.2 The experimental results for DE1 to DE6 and the average RT, ST and OS. 36
3.3 The experimental results for DE7 to DE12 and the average RT, ST and OS. 37
3.4 MATLAB PID tuner and random value results 37
3.5 Rank of DE Variants in Convergence Comparison 40

4.1 Control parameters of Optimization 48
4.2 Comparison Aircraft Pitch Control 54

5.1 Car-trailer combination parameters for ATS controller design . . 63
5.2 RWA Result Comparison. 69
5.3 Peak Lateral Acceleration (g) of Trailer Result Comparison 69

6.1 Vehicle parameter values. 74
6.2 Possibilities of GSC Look-up Table 79
6.3 RWA Comparison GS vs W/O GS 95

Abbreviations

EA	Evolutionary Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
GA	Genetic Algorithm
DE	Differential Evolution
PSO	Particle Swarm Optimization
GDE	Generalized Differential Evolution
AHV	Articulated Heavy Vehicles
RWA	Rearward Amplification
LQR	Linear Quadratic Regulator
DM	Decision Maker
MOEA	Multi Objective Evolutionary Algorithm
PAES	Pareto-archived Evolutionary Strategy
SPEA	Strength Pareto Evolutionary Algorithm
MOGA	Multi Objective Genetic Algorithm
ATS	Active Trailer Steering

Chapter 1

Introduction

Control systems are essential in the modern world to tackle ever-changing environmental disturbances while meeting adequate safety standards. Control systems must be robust and appropriately tuned for each application to ensure reliability and safety. Evolutionary algorithms are proven to be effective in offline tuning of control systems for various applications [1]. Depending on the complexity of a problem, evolutionary algorithms can even take days to compute an optimal result. This approach is logical for systems with no environmental dependencies. If a system is subject to external disturbances, its behaviour changes and an optimal controller may not be able to stabilize the system, resulting in a failure. To tackle this problem, this thesis proposes an approach to use evolutionary multi-objective optimization, for offline tuning of a control system at multiple dependencies, creating a comprehensive optimized look-up table to increase robustness of the system. This method will also provide posteriori decision making capabilities for multiple objectives.

1.1 Problem with Conventional Tuning Methods

The need for complicated and robust control algorithms is ever increasing for real-time systems. Ranging from autonomous cars to fully-automated intelligent manufacturing, all require complex controllers to increase efficiency, safety and reduce costs. The essence of a control system is to measure the output to provide corrective feedback to ensure that the system operates with the desired performance even though many environmental changes occur. With the shift towards Internet of Things and smart cities, the need for effective control methods gets very demanding. Optimal control techniques have a high potential to be enhanced by optimization. Some conventional strategies, to tune optimization techniques, are used in [2]. The obtained values are mainly system specific and tailored by adjusting the weights of gain matrices for the control method. These methods often require a deep understanding of the model and in-depth knowledge of the technique and the system, which is under enhancement. The thesis deviates from conventional techniques, and employs population-based meta-heuristics to tune weighting matrices to achieve optimal solutions.

1.1.1 Motivation

Car-Trailer combinations are a type of Articulated Vehicles (AV), which is a combination of multiple vehicle units [3]. An AV can carry a greater payload, compared to a single unit vehicle, due to an additional trailer while using the same engine [3]. This allows for a reduction in pollution and increases cost-effectiveness for customers. The trailers, on an AV, are joined with a mechanical coupling, called hitch [4]. Because of this coupling, some new problems arise.

The first challenge with an AV is the size of the vehicle, an AV is much larger than a normal car. Any collision or roll-over of an AV can result in damage not only to the AV but neighboring/surrounding vehicles and pedestrians. Thus, ensuring the safety of an AV is paramount but rather difficult.

The second challenge, with car-trailer combinations, is the nonlinear dynamic behaviors. The nonlinear dynamics of car-trailers may lead to severe traffic accidents, such as trailer sway, jackknifing, and rollover. A car-trailer combination can be linearized by assumptions, for example, fixing the forward speed to be a constant and the steering angle to be small [5].

The first challenge is tackled by using an ATS system and optimizing it using multi-objective algorithms. The second challenge is addressed by using a multi-objective optimized gain scheduling controller, with a two-dimensional look-up table. The similar problems with car-trailer combinations also plague the trucking industry. The amazon e-Commerce effect [6], discussed by Gwen Mortiz, has greatly impacted the trucking industry. Users can purchase goods online and get them shipped to their houses, workplaces, PO Boxes etc. These products have increased the already burdened trucking industry, which now must make several trips daily, for express shipping services required by consumers. Increasing the number of trucks to carry payload increases emissions, requires more drivers and have more cost of operation. Transport trucks are also involved in 20% of road crashes in Ontario [7]. The way to increase the safety of transport vehicles is to apply advanced optimized control systems, which is the motivation behind this thesis. Within the limitation of time and resources, this thesis only focuses on the advanced optimized control systems for car-trailer combinations. The proposed methodology can be extended to articulated heavy vehicles (AHVs), since both car-trailer combinations and AHVs share similar lateral dynamics.

1.2 Problem Statement

There are no perfect, noise and uncertainty free environments for a vehicle systems to operate. An optimal control system provides a way to tackle the ever-changing environments of real-time systems. In the design and modeling stage, the systems are functioning using a set of optimal parameter values. The design expectations have to be constrained to allow for that single set of parameter values to meet all requirements. The tuning methods (state space exploration) [2] are not fast enough to be implemented in safety-critical real-time systems. This is why designers have to make a tradeoff. Each solution also requires comprehensive testing to ensure that this solution results in a stable system, to avoid any catastrophic failure. In this thesis, an approach is proposed to tackle both of these problems. This method produces a set of solutions for multiple objectives without doing weighted aggregation for multiple objectives. The designer only needs to choose one of those solutions, depending on the operating condition of a real-time system, instead of re-running the optimizer and finding a new solution. This approach allows for more flexibility in designing controllers for real-time systems. These solutions are calculated for varying environmental factors to create multi-dimensional look-up tables, which is used with a gain scheduling controller.

1.3 Thesis Contributions and Novelty

Design of a control system is usually based on more than one objective. A simple example can be rise-time and settling-time of an inverted pendulum [8]. When evolutionary algorithms are used, all objectives are aggregated into a single fitness function or a single objective function [9]. The aggregated objective function is then evaluated to ascertain the results.

For two objectives, if a MOEA is used, a set of solutions is obtained where each solution has two fitness value, one for each objective. The optimal set of such values is called the optimal Pareto-front. This approach has not been utilized in the design of ATS systems for car-trailer combinations. The following subsections introduce the thesis contributions, which are based on the control system tuning for ATS systems of car-trailer combinations.

1.3.1 MOEA for ATS Controllers

Evolutionary algorithms have been used to tune ATS systems [10, 11]. Mono-objective algorithms are used for the tuning of the ATS controllers, e.g. genetic algorithms. Using mono-objective algorithm eliminates any possibility of choice, this is further discussed in section 4.3.

The first contribution of the thesis is to tune an ATS controller for car-trailer combination, and to provide the system designer with a choice from an optimal set of solutions. Ensuring each solution is stable and applicable within design goals. This is done by using MOEAs. The MOEA studied in this thesis is known as generalized differential evolution (GDE3), which is outlined in section 2.9. GDE3 is used for the design of ATS controllers in Chapters 5 and 6.

1.3.2 MOEA Optimized Gain Scheduled ATS Controllers

The car-trailer combination is a non-linear system. It is linearized using assumptions to generate a state space representation, which is then used in controllers design, e.g., linear quadratic regulator (LQR), to develop a feedback control system. This is the approach used in [11]. The second contribution of the thesis is the provision of an ATS system using an optimized gain scheduling controller (GSC) for car-trailer combinations. The LQR-based ATS controller is tuned using MOEA at various speeds to provide a lookup table for the system designer to implement in the control strategy. The GSC makes the system more robust. This is discussed in detail in Chapter 6.3.

1.3.3 Modular Approach for Design Optimization of Control Systems Using CarSim-MATLAB/Simulink Co-Simulations

Control system tuning requires system-specific knowledge, especially if a linearized model is to be generated. An example is the 3 degrees of freedom (DOF) car-trailer model that is derived in Chapter 5. The steps involve: 1) generating a linearized model to represent the non-linear system by using certain assumptions, and 2) testing to ensure that the linearized model derived captures the essential dynamic features if the non-linear system. This process is cumbersome and requires knowledge of the mechanical system to a great extent. In Chapter 6 of this thesis, an approach is used, which eliminates the use of linear control strategies and replace it with an evolutionary strategy, which can work without the requisite of a set of linear equations of the system.

1.3.4 Adaptive Driver Model with Varying Reaction Time and Vehicle Forward Speed

The thesis explores two parameters for gain scheduling for the built-in driver model offered in CarSim software: 1) driver model reaction time, and 2) vehicle forward speed. The built-in driver model in CarSim can be used to drive the virtual car-trailer combination to follow a prescribed single lane-change (SLC) trajectory. GDE3 is used to create a gain scheduling scheme, which considers the reaction time of the driver model and the varying vehicle forward speed. This allows for adjusting not only the control gain matrix of ATS controller, but also the essential parameter of the driver model.

1.3.5 Performance Analysis of MOEAs for Control System Tuning

The MOEA has not been used for the design optimization of ATS system for car-trailer combinations. The last contribution of this thesis is to compare popular MOEAs by means of evaluating the respective ATS controllers designed car-trailer combinations.

1.4 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 provides a literature review of control system tuning using evolutionary algorithms, problems with car-trailer combinations and how they can be solved. In Chapter 2 the building blocks of the thesis are also discussed, which include the car-trailer combination, optimization algorithms, multi objective optimization techniques and the linear quadratic regulator (LQR) control technique. The verification of optimization search algorithms starts from Chapter 3 where a PID controller is tuned using differential evolution to test which variant is the best, this is later used in Chapters 5 and Chapter 6. Chapter 4 illustrates a comprehensive case study to compare genetic algorithm based MOEA against a differential evolution based MOEA. Confirming that the algorithm performing well on existing models is essential before generating the models and testing the algorithm.

In Chapter 5, a 3-DOF car-trailer model is generated using a multi-body dynamic software. Then, the linear 3-DOF car-trailer model is used to design the ATS controller using the LQR technique. The LQR-based ATS controller is optimized based on open-loop dynamic simulation using the algorithm discussed in 4. Lastly, Chapter 6 is the crux of all previous chapters. It applies everything tested before, with a co-simulation with a Car-Trailer model from CarSim mechanical simulator. It reaffirms the results and provides insights into gain scheduling for non-linear systems, tuning based on adaptive driver model with varying reaction time and vehicle forward speed and algorithm comparison. Chapter 6 also provides recommendations for future studies.

This thesis involves five major steps to conclude. The first step is to decide the base evolutionary algorithm (EA) from a large selection of database of EA and their variants. Chapter 3 focuses on comparing different variants of differential evolution (DE). The results from Chapter 3 provide the mutation scheme and algorithm which is used as the base algorithm in Chapters 4 to 6. Generalized

Differential Evolution (GDE3) is the corresponding multi-objective evolutionary algorithm (MOEA) of differential evolution. In Chapter 4 the results received from Chapter 3 are used with a multi-objective scheme and compared to a popular algorithm. The GDE3 variant tested and verified in Chapter 4 is then used in Chapters 5 and 6. In Chapter 5, an LQR controller is tuned using GDE3 variant from the previous chapter and applied to a 3-DOF car-trailer articulated vehicle. The controller, objectives and constraint design from Chapter 5 is used in Chapter 6, which concludes the thesis results.

Chapter 2

Literature Review

The literature review focuses on methods for tuning control systems and stability of articulated vehicles while providing some background information on general techniques used in this thesis.

2.1 Evolutionary Algorithms Applied to Control Systems

Evolutionary algorithms (EAs) and MOEAs are proven to be effective in offline tuning of control systems for various applications [1]. The authors in [12] studied the effectiveness genetic algorithm (GA) in tuning of PI and LQR controllers for boiler-turbine plant.

Non-dominated sorting genetic algorithm (NSGA-II) was used as MOEA to optimize HVAC control system [13]. NSGA- II achieves an improvement, within the design constraints, while considering multiple objectives [13].

A MOGA was used to generate an optimal Pareto-front, allowing for multi-criterion decision making, in an energy conservation application for building design [14].

As discussed by the authors of [15], EAs were employed in most of the LQR implementations as optimizer [16, 17]. The authors of [17] concluded that differential evolution, along with other continuous optimizers, outperforms genetic algorithms, which reinforces the selection of Differential Evolution as the core optimization method. All the above studies, collectively, show that the LQR control method is used in real-time systems and that evolutionary algorithms achieve a better solution than other conventional methods or even fuzzy logic controllers.

An ATS control system, using the LQR-based controller, for articulated vehicles is designed and optimized by GA in [18] and shows that it is superior to other studies. A GA optimized active trailer differential braking system is outlined in [19] for a car-trailer combination. The aforementioned studies along with engineering applications show that evolutionary search algorithms are ideal for tuning control systems in a wide array of applications [1].

2.2 Linear Quadratic Regulator Technique

The LQR technique was introduced by Kalman [20, 21]. The LQR controller is a feedback controller used to provide optimal control for a dynamic system to ensure operation at minimum cost. A continuous-time linear system is defined as follows:

$$\dot{x} = Ax + Bu \tag{2.1}$$

Where A represents the states of the system which is a square matrix, B is the input matrix, x is the state vector, and u is the control vector. For the optimal controller design, the continuous-time LQR cost function is defined as

$$J = \int_0^\infty (x^T Qx + u^t Ru + 2x^T Nu)dt \tag{2.2}$$

Where Q and R matrices are used to find an optimal gain matrix. J is the internal cost function of the LQR control method, whereas the optimizer will try to minimize objectives, Settling-time (ST) and Rise-time (RT), by changing Q and R matrices, which in turn change how the LQR cost function performs in a closed-loop.

The optimizer manipulates the Q and R matrices to minimize the objective function described in Eq.(2.2). These matrices, which vary in size from system to system, consist of the variables and their number corresponding to the dimension, which determines the complexity of the system. The LQR controller gives a better overall performance when compared to a PID controller, but the values are harder to attain [22]. In a PID controller, there are three control gains, i.e. K_p, K_i and K_d. In terms of complexity, the dimension of the PID problem is 3. For an aircraft pitch control, the dimension of the problem is 9, as the Q matrix is a $3*3$ matrix. In [22] authors demonstrate that optimization of the control matrix of LQR is harder than a PID controller.

2.2.1 Design Optimization of LQR Controller

The research in this thesis combines two areas: utilizing the LQR technique for ATS controller design, and using evolutionary algorithms to optimize the LQR control parameters. The LQR technique has been utilized to design the

ATS controller for an AHV [23]. The study used a genetic algorithm to optimize a LQR controller for ATS of AHVs, and demonstrates the strength of the LQR controller, and also elaborates on the importance of optimization [23]. The work in [24] uses an LQR controller to control an anti-roll bar, which prevents vehicle rollover and they demonstrate how the LQR controller can provide stability to vehicular systems.

The LQR controller can be used to mitigate the risks in many safety-critical systems, for instance ATS Systems to reduce the risk of rollover and jackknifing in AHV [3, 25, 26]. It also has been used in Active Suspension System [27] to provide a smoother and safer driving. This work also shows that a LQR controller performed better than comparative controllers, in active suspension, and reached the performance of industry implemented suspension systems in some situations. A comparative analysis of multiple control strategies for active steering system has been conducted and found that the LQR outperforms fuzzy logic controllers (FLC) and improves low-speed maneuverability and high-speed stability [28]. An LQR controller provides a faster and stable response, in real-time magnetic levitation performance, as compared to FLC and PID controllers [29, 30]. It is presented in [15] that an LQR controller has been successfully used to control aircraft pitch.

2.2.2 Difference in Approach

Tuning an LQR controller is a big part of this thesis, as the LQR controller is used in ATS system for the car-trailer combination to ensure stability. Similarly an LQR controller is designed for the control system for aircraft pitch control.

The difference in approach to both applications is the use of MOEA, GDE3 for tuning. GDE3 is proven to be better than the counterpart genetic algorithms.

2.3 Unstable Modes of Car-Trailer Combinations

A typical car-trailer combination consists of a leading vehicle unit and a trailing unit, which are connected with a mechanical hitch [4]. The hitch connection between the vehicle units generates several mechanical constraints, making the dynamics and kinematics of this combination more complex to investigate compared to a single-unit vehicle, e.g., car or truck. Due to the unique dynamics of the combination and the mechanical constraints, the car-trailer combination usually faces unstable states, which could lead to fatal accidents. Typical unstable motion modes lead to crash of car-trailer combinations are trailer-sway, jackknifing and rollover [31]. Several reasons could lead to each of the aforementioned unstable motion modes, e.g., high-speed evasive maneuvers, crosswinds, road conditions, and payload variation of the trailer [32].

Trailer sway is a dynamic phenomenon where the trailer exhibits a fishtailing angular motion around the hitch. The low yaw-damping ratio of the trailer is the primary cause of the trailer sway. The forward speed of the combination directly influences this damping ratio; when traveling on a straight line at a critical speed, the corresponding yaw damping ratio decreases to zero [33]; while traveling above the critical speed, and the car-trailer combination loses its lateral stability with increasing amplitude of angular oscillation [34]. The damping ratio is an important criterion for determining the yaw-stability of car-trailer combinations. If the combination is exposed to external disturbances, e.g., crosswind, it may cause the trailer to lose traction, resulting in swinging and skidding of the combination [32].

The second is Roll-over, as the name suggests, this is when the rear trailer, due to undue forces [35, 36] and shift in center of gravity, might roll-over, causing the entire vehicle to rollover on the road.

The third unstable motion mode is known as Jackknifing. It is heavily dependent on the relative braking force distribution between the car and the trailer. In the event of panic breaking, if the articulated angle reached a certain limit, it could result in jackknifing where the driver has no control over it [37]. Jackknifing usually occurs during curve negotiation with heavy braking and external disturbances, for example crosswind [38].

2.4 Stabilizing Articulated Vehicles

The detrimental effect of the three aforementioned unstable motion modes is reduced by using various strategies. These strategies can be passive and active. Passive strategies require no additional energy or complicated implementation whereas active strategies do require additional energy to provide stabilization. The strategy considered in this thesis is the ATS system, which is an active control strategy. This system requires a steering actuator to steer the wheels of the trailer of a car-trailer combination. The distinguished feature of the thesiss approach is the use of GDE3 for tuning the LQR controller, instead of the mono-objective based method for tuning the controller. Other control strategies are reviewed in this section.

2.4.1 Passive Steering System

Passive steering systems (PSSs) have been used to improve low-speed maneuverability. There are various types of PSSs: Self-Steering, Command Steering and Pivotal Bogie Systems, etc. A comparison of these is given in [39]. The common element among all the systems is how they deal with maneuverability. Curved path turning, e.g., 90-degree turning, frequently occur at low speeds. During a turn, the trailer is required to follow the trajectory of the car. When the car turns, a PSS uses geometric the relationship between front axles and the trailer axles to ensure the trailer to follow the same path of the car [39, 40]. A PSS may promote a greater tail-swing. Tail-swing is a dynamic phenomenon of a car-trailer combination, where the trailer oscillates angularly around the hitch at the rear end of the car. At high speeds, the lateral stability is crucial and should be enhanced, while PSSs performs poorly in terms of high-speed lateral stability [41]. For articulated vehicles, there exists a distinguished dynamic phenomenon, called rearward amplification (RWA), implying that the trailer shows larger lateral motion than that of the car. Generally, to ensure the high-speed lateral stability of an articulated vehicle, the PSS should be locked or disabled above a high speed, e.g., 70 km/h.

2.4.2 ATS Systems

An ATS system may allow an articulated vehicle to operate with a fix steer ratio between the leading vehicle unit steering angle in relation to the trailer axle steering angle [42]. Using an ATS, the trade-off between the low-speed maneuverability and the high-speed lateral stability may be mitigated. An ATS system may improve the RWA measure at high speeds and the path-following off-tracking (PFOT) measure at lower speeds, without having to use multiple modes of operation and worrying about switching between them. An ATS system is intended to manipulate the trajectory of the trailer and the forces/-torques generated between trailer tires and the road surface, thereby leading to improving the trailers path-following capability and enhancing the lateral stability. To implement an ATS system, a controller should be designed. The LQR control technique has been extensively used for the design of ATS controllers. To achieve desired performance, the control parameters of the LQR-based ATS controller may be optimized.

2.4.3 Multi-Mode Steering Control

As discussed earlier, reducing PFOT increases RWA, and vice versa. Using both active and passive steering systems together is one of the acceptable solutions to the problem [42, 43]. Depending on the operation of the vehicle, the steering system switches between Active and Passive Steering. This in turn grants maneuverability at lower speeds and stability at higher speeds. The disadvantage of the multi-mode system is the great increase in steering system complexity. Implementing a dual-steering system on existing trucks/AHVs is very difficult and costly.

2.4.4 Differential Braking

Fancher discussed the use of differential braking for improving the lateral stability of AHVs. Differential breaking helps reduce RWA [44]. In [44] the results showed that using their differential breaking model, RWA was reduced to 1.7 from 2.3 at high speeds. This is achieved by applying different pressure at the braking actuators of the right and left wheels on an axle when the differential braking system is actuated. Although the ideal RWA value of 1.0 was not achieved, a great reduction was still achieved. The downside of using differential braking is the toll on the braking system. The brakes are more prone to heating up. Hence, this system must be managed carefully where it is only turned on when necessary

2.4.5 Teslas Approach to Jackknifing Prevention

Tesla unveiled their electric semi-trucks in 2017 [45]. One of the important safety features they unveiled was Jackknifing prevention. The approach by Tesla is to use electric motors on individual tires and a dynamic control system. The control system can get feedback from each motor and various sensors around the vehicle. Whenever a danger of Jackknifing occurs, the motors activate to counteract the action, thus preventing jackknifing. This approach is expensive and impractical for current powertrains. None of these methods, individually, solve the entire problem, under all operating conditions. The most promising method to use is the ATS technology. The research for this thesis leads to the opinion that an ATS system may be optimized to provide an acceptable RWA and PFOT values, when used with an LQR controller.

2.5 Differential Evolution

Storn and Price introduced Differential Evolution (DE) [46] in 1995. It is a population-based metaheuristic (P-metaheuristic) evolutionary algorithm (EA), which has been inspired by a genetic algorithm (GA). DE consists of mutation, crossover, and selection; the algorithm performs best for continuous-valued problems. The working principle of DE is alike GA. It starts with population initialization. During initialization all members are assigned fitness values according to fitness functions, these values correspond to how good these population members are for solving the problem. Following initialization, the selection is made at random from the population space, anywhere between 2 to 4 members are chosen, depending on the utilized mutation scheme. There are more than ten mutation variants in DE [47]. The fitness of the population member, once chosen, is evaluated after it undergoes mutation and recombination. After applying the crossover and calculating the resulted offspring's fitness value, the best parent and offspring is selected to the next generation, in fact, the selection follows a greedy strategy. DE has a specific set of control parameters: Population Size (Np), Crossover rate (Cr), mutation scaling factors, and mutation schemes [47]. Seven different mutation schemes are listed as follows,

$$V = X_1 + F * (X_2 - X_3) \tag{2.3}$$

$$V = X_1 + F * (X_2 - X_3 + X_4 - X_5) \tag{2.4}$$

$$V = X_{best} + F * (X_1 - X_2 + X_3 - X_4) \tag{2.5}$$

$$V = X_3 + G * (X_{best} - X_3) + G * (X_1 - X_2) \tag{2.6}$$

$$V = X_{best} + F * (X_1 - X_2) \tag{2.7}$$

$$V = X_i + G * (X_3 - X_i) + F * (X_1 - X_2) \tag{2.8}$$

$$V = X_i + G * (X_{best} - X_i) + G * (X_i - X_2) \tag{2.9}$$

where V is the mutant vector, F and G is the scaling factor with values between 0 and 1, X_i are randomly selected individuals from the population. The crossover or recombination method is as follows, regardless of the mutation method.

$$U(j) = \begin{cases} V(j), \text{if } (rand(0,1) < Cr) \cup (j = j_{rand}) \\ X_i(j), \text{otherwise} \end{cases} \tag{2.10}$$

where U is the population member created after recombination. DE may be an effective algorithm for control system optimization because control system weighting matrices are real-valued vectors instead of discrete-valued. For better final results and faster convergence, the mutation schemes is varied and tested to see which mutation scheme provides a better final result or finds an optimal value faster by running system-specific comparisons. A study conduction in [48] sheds light on the strength of DE compared to GA and PSO.

2.6 Optimal Pareto-front

For multi-objective optimization, with conflicting objectives, there is no single solution rather a set of solutions. Each solution consists of costs equal to the number of objectives. One solution is said to strongly dominate another solution if and only if all the costs of one solution are better than the other or if and only if all costs are no worse, and at least one is better. If a solution is not strongly dominated by any other solution then it is a Pareto solution [49]. An optimal Pareto-front is the set of all such solutions, which are not dominated by other solutions [50].

2.7 Knee-point

Knee-point is an optimal trade-off point in a set of Pareto-optimal solution. There are many different methods of calculating the knee-point, and these methods are studied in detail in [51]. The approach used in this thesis, to determine knee-point, is the trade-off approach [51]. All the solutions are studied and a point is chosen, from the knee-region, which has a similar deviation from both extreme points for each of its costs. The knee-point is discussed and defined in great detail in [51]. An example of a knee-point is given in Fig. 2.1 where the point B is the knee-point [51].

2.8 Non-dominated Sorting Genetic Algorithm

NSGA-II (Non-Dominated Sorting Genetic Algorithm) [9] is an evolutionary algorithm to solve multi-objective optimization problems and get an optimal Pareto-front solution. Optimal Pareto solutions can be used by the user/researcher to achieve their desired outputs. NSGA-II's elitist preservation never discards best solutions [9]. The process of selection mutation and recombination continues until preset conditions are satisfied and a desired solution is achieved. The second part of NSGA-II ensures diversity between candidate solutions. At the end of the entire procedure, an optimal Pareto-front set of solutions is found.

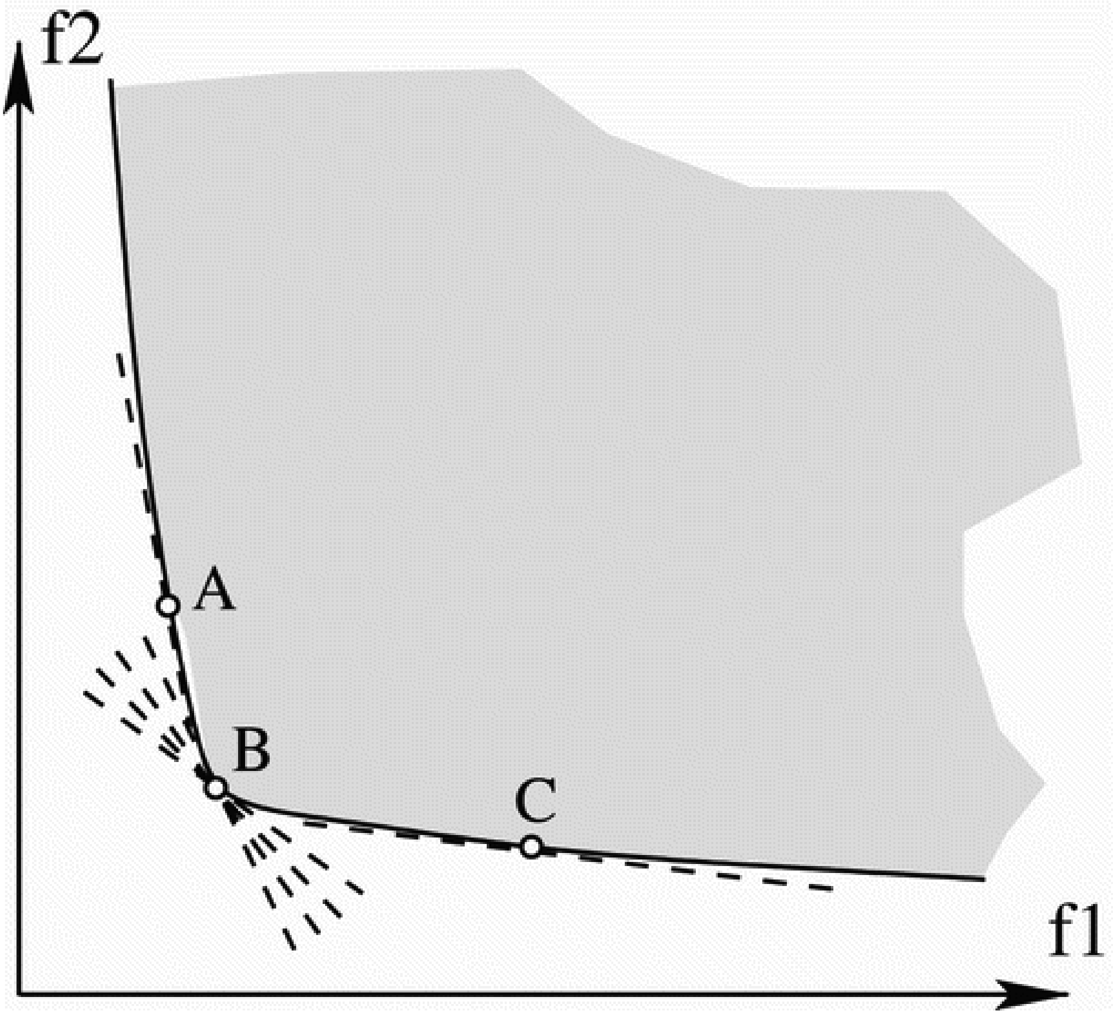

Figure 2.1: Point B is the knee-point. This image is taken from [51]

Each population member is assigned two values, i.e. η_p, the number of other solutions which dominate this solution, and S_p, set of solution that the solution dominates [9]. The process continues until the termination conditions is reached. Sorting is done based on η_p and S_p, and an optimal Pareto-front is occupied by the solutions, which has not been dominated by any other solution [9]. Each solution is visited $N-1$ times and complexity of the problem is $O(N^2)$ [9], where N is the population size. The complexity would be $O(MN^2)$, where M is the number of objectives [9]. Detailed description of the algorithm can be found in [9].

Crowding Distance, according [9], is defined as a diverse set of solutions provide a better optimized result. NSGA-II maintains solutions diversity by using the crowding distance concept. Crowding Distance is calculated by selecting a solution and then calculating the distance between the point and the two adjacent points in the Pareto-front. Crowding operator sets the desired diversity among the candidate solutions (crowding distance). Each population member is assigned a crowding distance and a rank, which corresponds to the rank of the Pareto-front.

2.9 Generalized Differential Evolution

Non-dominated Sorting Genetic Algorithm (NSGA) is the base of Generalized Differential Evolution (GDE3). Apart from these two algorithms, there are other MOEAs, e.g. Pareto Evolution Strategy (PAES) [52] and Strength-Pareto Evolutionary Algorithm (SPEA) [53]. GDE3 is chosen based on a comprehensive comparison done between these algorithms in [9, 54, 55]. With this information, GDE3 is compared to NSGA and conventional tuning rather than re-perform established experiments to gain similar results just to include other MOEA.

GDE3 removes the GA part of NSGA and replaces it with DE's method of selection, mutation and recombination. Bit string representation makes GA great for scheduling and discrete problem, but DE exceeds GA for real-valued problems [56], e.g. optimization of the LQR controllers. All other aspects of NSGA are unchanged.

2.9.1 GDE3 Algorithm

In the algorithm 1, D represents number of decision variables, max_{it} represents generational iterations, x_{lb} and x_{ub} are lower and upper bounds for the search space respectively, F is the scaling factor, F_{min} and F_{max} are the lower and upper bounds of the scaling factor, Np is the population size, V is the mutant vector. For non dominated sorting, p is the selected individual, S_p are the sets dominated by p, and n_p are the number of individuals that dominate p. For crowding distance, Pf_i is the ith pareto front, TPf represents total pareto fronts, n_i represents number of individuals in that pareto front, O_1 is value of objective function 1, and O_2 is the value of objective function 2. The GDE3 method is outlined in algorithm 1.

Algorithm 1 GDE3 Algorithm [57],[9]

```
$INPUT : D, max_{it}, Np, x_{lb}, x_{ub}, F \epsilon (F_{min}, F_{max})$
Initialize population uniform randomly
**Non Dominated Sorting**
**for** $p = 1, p \leq Np, p{+}{+}$ **do**
    $S(p) = \phi$
    **for** $q = 1, q \leq Np, q{+}{+}$ **do**
        **if** $q == p$ **then**
            Skip and check next member
        **end if**
        **if** $Np(p)$dominates$Np(q)$ **then**
            $S_p = S_p \cup q$
        **else**$Np(q)$dominates$Np(p)$
            $n_p + +$
        **end if**
    **end for**
**end for**
Pareto-optimal members are such that $n_p = 0$
**Crowding distance**
Initialize distance between all members of front to be 0
**for** $i = 1, i < TPf, i + +$ **do**
    Sort population based on each objectives' cost
    Assign inf distance to boundary values
    Assign crowding distance based on
    **for** $k = 1, k \leq$ No. of objectives, $k + +$ **do**
        **for** $j = 2, k \leq (n_i - 1), k + +$ **do** $CD(k,j) = \frac{O_j(k+1) - O_j(k-1)}{O_j(n_i) - O_j(1)}$
        **end for**
    **end for**
**end for**
**Main loop**
**while** $it < max_{it}$ **do**
    $Selection$
    $X_1, X_2, X_3 \epsilon \{1, 2, ... Np\}$
    $j_{rand} \epsilon 1, 2, ..., D$
    $V = X_1 + F * (X_2 - X_3)$, Creating mutant vector V
    $x_{lb} \leq V(i) \leq x_{ub}$,ensure V is within bounds
    Mutation and Crossover:
    $U(j) = \begin{cases} V(j), \text{ if } (rand(0,1) < Cr) OR (j = j_{rand}) \\ X_i(j), \text{ otherwise} \end{cases}$
    Selection $X_i = \begin{cases} U, & \text{, if } U \text{ dominates } X_i \\ X_i, & \text{, otherwise} \end{cases}$
    $it = it + 1$
    **Non-Dominated Sorting**
    **Calculate Crowding Distance**
    $it = it + 1$
**end while**
```

Chapter 3

Testing the Base Algorithm: Differential Evolution

DE has many variations, mainly due to the difference in mutation schemes. The effect of changing mutation schemes on control system tuning is the topic of focus in this chapter. The best mutation scheme variant is found to be DE5, which makes use of the mutation scheme described in Eq.(2.7).

This chapter also clarifies the effect of a distributed framework on control system tuning. It will be examined whether splitting the population into multiple small islands allowing for information exchange and individual island evolution has any effect on control system tuning or not. The achieved results show that distributed framework improves the final results.

It is very challenging and time consuming to test 12 variations of DE on ATS tuning, for car-trailer combination. This chapter makes use of a PID controller example to give an overview of the performance difference of DE variants in control system tuning. There is no guarantee that the best variant in PID controller tuning is the best variant for ATS tuning. The results in further chapters show that the variant selected can tune, within design guidelines.

3.1 PID Controller

We have come a long way in increasing the computational efficiency for computers to run optimization algorithms, which once required highly specialized equipment. Applications of optimization are in every field of technology, and every system can be optimized to be better. This section concentrates on tuning of PID controllers using DE algorithms to obtain better control gain K_p, K_i, K_d values. DE is at the core of GDE3, which will be utilized as a MOEA to tune the ATS controllers for car-trailer combinations in later chapters of this thesis. To benchmark the algorithms, a general second order system is considered.

Various methods are proposed for tuning PID controllers. In the view of optimization, the goal is to find an optimum solution. Initially, classical optimization techniques were developed to find optimum solutions. Fermat and Lagrange found calculus-based formulae for identifying optima, while Newton and Gauss proposed iterative methods for moving towards an optimum. The term linear programming for certain optimization cases was due to Dantzig, although much of the theory had been introduced by Kantorovich in 1939 [58]. The classical optimization techniques are beneficial in finding the solution or unconstrained maxima or minima of continuous and differentiable functions. Differential calculus is used to reach up to the solution in these types of problems. As many practical problems are not continuous or differentiable, so classical methods could not be used for the solution to these types of problems.

However, classical methods are the base for the development of linear programming techniques and many modern optimization methods. Numerical methods, e.g. Linear Programming, Integer Programming, Quadratic Programming and Non- linear programming, are used for the solution to many problems.

In this section, the convergence rate and performance of 12 DE variants over 5 different fitness functions (varied by Steady State error calculations) are compared. The objectives are to minimize the following performance measures of PID controllers: Settling Time, Rise Time, Overshoot and Steady State Error. The four performance measures make the PID controller turning a multi-objective optimization problem. In this section, a single objective algorithm is used with an aggregated fitness function.

3.2 Tuning methods for PID controllers

Conventional PID controllers are categorically the most generally used control algorithms in various applications due to their practicality. Their comparatively simple structures, which can be simply executed, and the accessibility of well-established rules for tuning the parameters of the controllers are the main reasons for various real-time applications.

A PID controller is a feedback controller, which continuously calculates the appropriate adjustment whenever the actual condition of a plant differs from the desired value or set point. It consists of a proportional control term, an integral control term, and a derivative control term. To get a satisfactory control performance, one only needs to adjust three parameters of the controller gain, the integral or rest time, and the rate or derivative time in a PID controller. The adjustment of these parameters is called tuning, and many experimental methods have been developed for this purpose, e.g. Ziegler-Nichols and Astrom-Hagglund [59]. The term P is proportional to the current value of the $SP - PV$ error, i.e., $e(t)$. Here SP is the set-point and PV is the proposed value or Process variable. Term I accounts for the past values of the $SP - PV$ error and integrates them over time to produce the I term. Term D is the best estimate for the future trend of the $SP - PV$ error.

Various tuning methods have been proposed from 1942 up to now for gaining better and more acceptable control system response based on desirable control objectives, e.g., percent of overshoot, integral of the absolute value of the error (IAE), settling time, manipulated variable behavior, etc. Some of these tuning methods have considered only one of these objectives as a criterion for their tuning algorithm, and some of them have developed their algorithms by considering more than one of the mentioned criteria [60].

The goal of tuning a PID controller is to make it stable, responsive and to minimize overshoot. These goals - especially the last two - conflict with each other. You must find a compromise between the goals, which acceptably satisfies them all. Process requirements and physical limitations will determine the balance between the amount of acceptable overshoot as well as the demand for responsiveness. A PID controller is essentially a generic closed-loop feedback mechanism. The auto-tuning of PID control started by Ziegler and Nichols. This method is a trial and error test, in which one must produce oscillations with constant amplitude. Ziegler and Nichols is an effective conventional technique for tuning of PID Controllers. However, sometimes it does not provide a good solution and tends to produce large overshoots.

3.3 DE Algorithm

DE has been benchmarked to be faster and more accurate than genetic algorithms in various research papers [61, 62]. DE uses continuous vectors instead of chromosome based on bits [63]. Systems, which require population members to be continuous, e.g., a second order system, greatly benefit from this [62].

A PID controller designed for a second order system has the control gains, K_p, K_i and K_d, whose values belong to real continuous numbers. Each population member of the DE algorithm has three values corresponding to each K_p, K_i, and K_d. GA differs by incorporation of various crossover strategies to achieve better more optimized results, for DE the reliance is on mutation as can be seen from Eqs.(2.3)-(2.9).

DE algorithm starts with the initialization of the population within the lower and upper bound defined. Each population member created is evaluated and a cost is assigned based on the fitness function. The main loop of DE begins after population initialization. In each iteration of the main loop, three to five population members are chosen at random, depending on the mutation scheme, and the best member may also be used. Upon these members, the mutation is performed, according to the selected mutation scheme. Eq.(2.10) shows that the crossover between current population member and the mutated population member is performed. Once crossover is completed and a new vector is created, which is evaluated against the population member currently selected, if it is better, it is kept, otherwise discarded. The global best value is always kept, ensuring elitist preservation. A detailed algorithm is outlined in [46].

As any evolutionary algorithms, DE has a set of parameters. These parameters include Population Size (N_p), Crossover probability (C_r), Scaling Factors ($K \& F$), definition of a Search Space (upper and lower bounds), mutation and crossover schemes. Apart from that, it can have one or more fitness functions to evaluate how good a population member is and termination conditions, in case termination condition is based on the number of generations (NFC).

3.3.1 Island distribution of DE

Genetic Algorithm is inspired from real life. Real life is full of information exchange and individual or collective evolution. Families evolve, which evolve cities, which evolves countries, and then there is global evolution. Resources, information, and technology are exchanged as well. This similar concept can be seen in the variation of DE, also discussed in [47]. It is called Distributed Differential Evolution. In this case, the total population is split into islands, each island has its own individual evolution, and it also sends migrants to adjacent islands as information exchange. This is called Island Based Distributed Framework with Information Exchange, which is part of a Low- Level Relay Based Method as well. The same distribution is followed. The Island-based DE has the following parameters: Number of Islands (ni), Number of Migrants (nm), Migration Frequency (mf), The Migration Topology (mt), the Selection Policy (sp) and lastly the replacement policy (rp).

In this experiment, Number of Islands (ni) is 4, Number of Migrants (nm) is 1, and Migration Frequency (mf) is every $45th$ generation. Migration topology (mt) is migration between adjacent islands with the first island not accepting any migrants. The Selection Policy (sp) for the migrant is selecting the best member of the island to send to the adjacent island. Replacement policy (rp) is defined that the migrant will replace any random member, which is not the best member of the island. Each island can have different mutation schemes, which allows for high variation. In this thesis, five such variations are considered: 1) each island uses mutation scheme described in Eq.(2.3); 2) each island uses mutation scheme defined in Eq.(2.4); 3) each island uses mutation scheme specified in Eq(2.5); 4) each island uses mutation scheme expressed in Eq.(2.6); and 5) island 1 uses Eq.(2.3), island 2 uses Eq.(2.4), island 3 uses Eq.(2.5), and island 4 uses Eq.(2.6). Table 3.1 lists all DE labels and their definitions.

TABLE 3.1: The table lists the assigned labels of the variants of DE based on mutation schemes.[†]

Given Label	Mutation Strategy	Distributed (Yes/no)
DE1	Eq.(2.3)	No
DE2	Eq.(2.4)	No
DE3	Eq.(2.5)	No
DE4	Eq.(2.6)	No
DE5	Eq.(2.7)	No
DE6	Eq.(2.8)	No
DE7	Eq.(2.9)	No
DE8	Eq.(2.3), on all Islands	Yes
DE9	Eq.(2.4), on all Islands	Yes
DE10	Eq.(2.5), on all Islands	Yes
DE11	Eq.(2.6), on all Islands	Yes
DE12	Eq. 2.3, on Island 1 Eq.(2.4), on Island 2 Eq.(2.5), on Island 3 Eq.(2.6), on Island 4	Yes

[†] For example DE4 is the label assigned to a variant of DE which uses Eq.(2.6) as the mutation scheme. These labels are used in the analysis section.

3.4 Objectives

When tuning a system, many things must be considered. Settling Time, Rise Time, Overshoot are among some of them, as well as Steady State Error. If all four of these are considered as fitness criteria, then it is a multi-objective optimization problem. A fitness function, which is an aggregate of all four, can be a good approach in finding an optimal solution. The final solution needs to have acceptable values for K_p, K_i, and K_d, which will result in the minimization of all four objectives. In this section, the four objectives are dealt with by aggregating them using Eq.(3.1) [64]. The aggregation approach outlined in [64] is used. This approach utilizes a varying weighted fitness function. The simple aggregation method is adding all the objectives together and converting a multi-objective into a mono-objective or single objective problem. This did not result in a great variation when the system is tuned, nor a good set of values. The approach used is followed from [64], where Eq(3.1) is used to get better results.

$$F = \frac{1}{1 + e^{-\alpha}} * (RT + ST) + \frac{e^{-\alpha}}{a + e^{-\alpha}} * (OS + SS) \tag{3.1}$$

Where RT is the Rise Time, ST is the Settling Time, OS is the Over Shoot, SS is the Steady State Error and F is the fitness/objective function. Here α is the weighted variable. The value of α will vary in the range [-5, 5] [64].

3.5 Steady State Error

RT, ST, and OS are generated through simulation graphs directly, but steady state error must be calculated. To calculate SS, there are various techniques, which are described in Eq.(3.2)-(3.6). The methods to find SS are taken from[65] for comparison. Adding a variety of SS methods in simulation ensures a reliable comparison and a more concrete conclusion.

$$MSE = \frac{1}{t} * \int e(t)^2 \, dt \tag{3.2}$$

$$ITAE = \int t * \left| e(t) \right| dt \tag{3.3}$$

$$IAE = \int \left| e(t) \right| dt \tag{3.4}$$

$$ISE = \int e(t)^2 \, dt \tag{3.5}$$

$$ITSE = \int t * e(t)^2 \, dt \tag{3.6}$$

where MSE (Mean Squared Error), ITAE (Integral of Time Multiplied by Absolute Error), IAE (Integral of Absolute Magnitude of the Error), ISE (Integral of the Squared Error) and ITSE (Integral of the time multiplied by Squared Error) are the five methods to calculate SS listed in [65]. Eq.(3.1) is the fitness function with a steady-state error component. Five different ways to calculate steady state error results in five different variants of the fitness functions. This allows for the comparison of the performance of not only DE for tuning, but how variation in Steady State Error calculation affects DE performance as well as the final response of the second order system.

3.6 Second Order System

A system whose input-output equation is a second order differential equation is called Second Order System. There are many factors that make second-order systems important. They are simple and exhibit oscillations and overshoot. Higher order systems are based on second order systems. The second order systems being used as a test bench is defined in Eq.(3.7) and the controller in Eq.(3.8).

$$\frac{1}{s^2 + 10s + 20} \tag{3.7}$$

$$k_p + \frac{k_i}{s} + k_d * s \tag{3.8}$$

3.7 Performance Comparison

The first experiment deals with the performance of DE. Variants of DE are compared, for the given parameters. The results of the experiment show the advantages of islands and hybridization along with which mutation scheme outperforms the others. Steady State Error is omitted from Table 3.4, as it is spread over response time. Rise-Time, Settling-Time and Peak-Overshoot, listed in Table 3.4, are used to ably judge the system performance.

The DE parameters are: Np = 48, Cr = 0.95, Generations = 200, and the number of runs to average values = 5. Moreover, the search space is [1, 250]. We also tuned the same system using MATLAB built-in PID Tuner and gained values of Settling Time, Response Time and Overshoot, as well as a random value to compare DE. A random value is used to ensure that the algorithm is working using evolution and not due to randomness. These two are anchors around

Table 3.2: The experimental results for DE1 to DE6 and the average RT, ST and OS.

SS	PM	DE1	DE2	DE3	DE4	DE5	DE6
MSE	RT(s)	0.1589	0.0854	0.1695	0.1490	0.1441	0.1159
MSE	ST(s)	0.2620	0.9777	0.3896	0.4027	0.2294	0.7359
MSE	OS	0	0	0	0	0	0
ITAE	RT(s)	0.1366	0.1239	0.1510	0.1482	0.1672	0.1596
ITAE	ST(s)	0.6261	0.6795	0.2346	0.2817	0.3183	0.2525
ITAE	OS	0	0	0	0	0	0
IAE	RT(s)	0.1322	0.1530	0.1407	0.1375	0.1359	0.1482
IAE	ST(s)	0.1844	0.2444	0.2140	0.1981	0.2015	0.2095
IAE	OS	1.3798	0	0	1.1198	0.4546	1.8158
ISE	RT(s)	0.1567	0.1468	0.1724	0.1361	0.1358	0.1566
ISE	ST(s)	0.2438	0.3449	0.2851	0.2021	0.2055	0.3030
ISE	OS	0.0494	0	0	0.4220	0.0322	0
ITSE	RT(s)	0.0920	0.0912	0.0895	0.0900	0.0891	0.0916
ITSE	ST(s)	0.9034	0.9080	0.9195	0.9165	0.9222	0.9055
ITSE	OS	0	0	0	0	0	0
AVG	RT(s)	0.1353	0.1201	0.1446	0.1322	0.1344	0.1344
AVG	ST(s)	0.4447	0.6309	0.4086	0.4002	0.3754	0.4813
AVG	OS	0.2858	0	0	0.3083	0.0974	0.3631

which the algorithm performs. The experiment is repeated 51 times and the values averaged to eliminate stochastic bias in evolutionary algorithms.

3.7.1 Results and Analysis

Tables 3.2 and 3.3 offer the results of the experiments. Table 3.4 displays the results from MATLAB PID Tuner and a random value. The purpose of the random value is to emphasize the convergence of the evolutionary algorithm. A comparison of the results offered in Tables 3.2 and 3.3 with those provided in

Table 3.3: The experimental results for DE7 to DE12 and the average RT, ST and OS.

SS	PM	DE7	DE8	DE9	DE10	DE11	DE12
MSE	RT(s)	0.1489	0.1258	0.0975	0.0992	0.1343	0.1458
MSE	ST(s)	0.5434	0.6719	0.8900	0.8608	0.2463	0.2510
MSE	OS	0	0	0	0	0	0
ITAE	RT(s)	0.1652	0.1480	0.1415	0.1333	0.1310	0.1571
ITAE	ST(s)	0.3535	0.2513	0.2194	0.6408	0.5903	0.2732
ITAE	OS	0	0	0	0	0	0
IAE	RT(s)	0.334	0.1305	0.1355	0.1332	0.1560	0.1679
IAE	ST(s)	0.1957	0.2050	0.2052	0.2158	0.3084	0.2890
IAE	OS	0.6483	0	0.0703	0	0	0
ISE	RT(s)	0.1342	0.1714	0.1438	0.1398	0.1466	0.1362
ISE	ST(s)	0.1967	0.2682	0.3963	0.2101	0.2066	0.2576
ISE	OS	0.6645	0.4475	4.9219	0.2383	1.7173	0
ITSE	RT(s)	0.0893	0.0900	0.0879	0.0912	0.0895	0.0902
ITSE	ST(s)	0.9210	0.9160	0.9309	0.9083	0.9198	0.9150
ITSE	OS	0	0	0	0	0	0
AVG	RT(s)	0.1342	0.1322	0.1213	0.1193	0.1315	0.1394
AVG	ST(s)	0.4421	0.4625	0.5283	0.5672	0.4543	0.3972
AVG	OS	0.2626	0.0895	0.9984	0.0477	0.3435	0

Table 3.4: MATLAB PID tuner and random value results

Method	RT(s)	ST(s)	Overshoot
PID Tuner	0.3523	1.2112	6.3913
Random Value	.9355	2.1074	0.6770

Table 3.4 reveals that the all DE variants show better performance than the Matlab PID Tuner and Random Value for the gains. The difference is not minute. This shows that the DE variants can, regardless of the mutation scheme, perform better than conventional methods for PID tuning. In terms of the rising time, DE10 performs the best, DE5 achieves the shortest settling time, and DE2, DE3, and DE12 all can achieve 0 overshoot. From this experiment, it can be

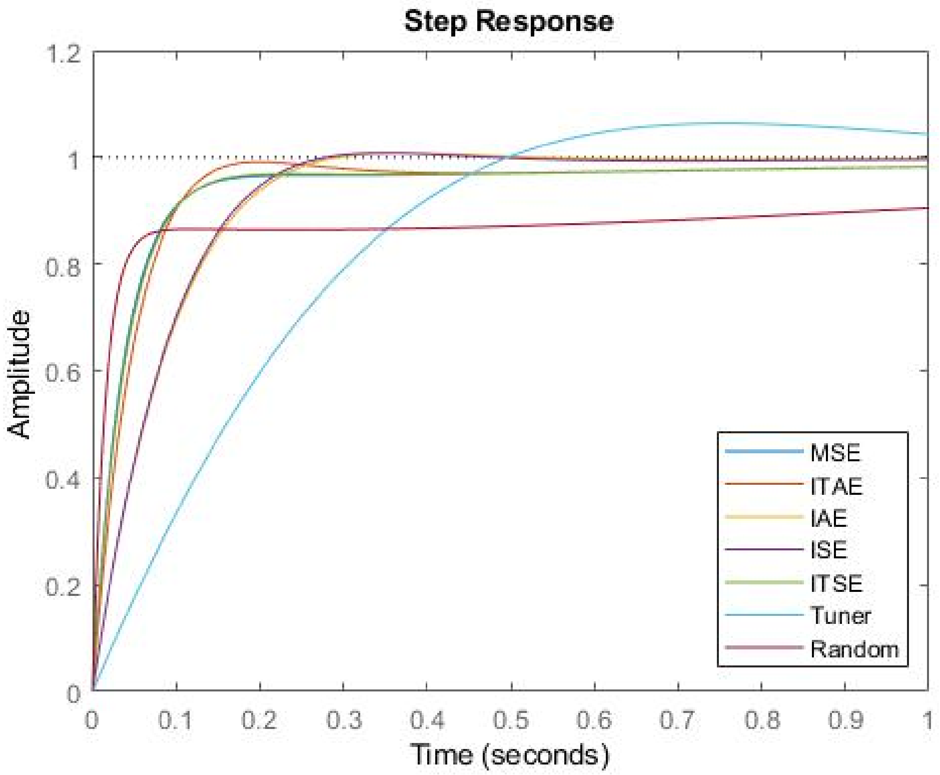

FIGURE 3.1: Responses for DE1 and MATLAB Tuner and Random Value

concluded that DE may greatly enhance the performance of the Second Order System when compared to other conventional methods. Table 3.3 displays that Island-based DE and Hybrid DE perform very well. DE10 and DE12 are Island-based and hybrid DE, respectively, and they can optimize 2/3 of the parameters better than their counter parts. With further optimization and tweaking of the Islands and Hybrid Algorithm or increasing the number of runs, better results can be achieved. Figs. 3.1-3.2 show the responses for DE1 and DE2. The responses of other DE variants are similar.

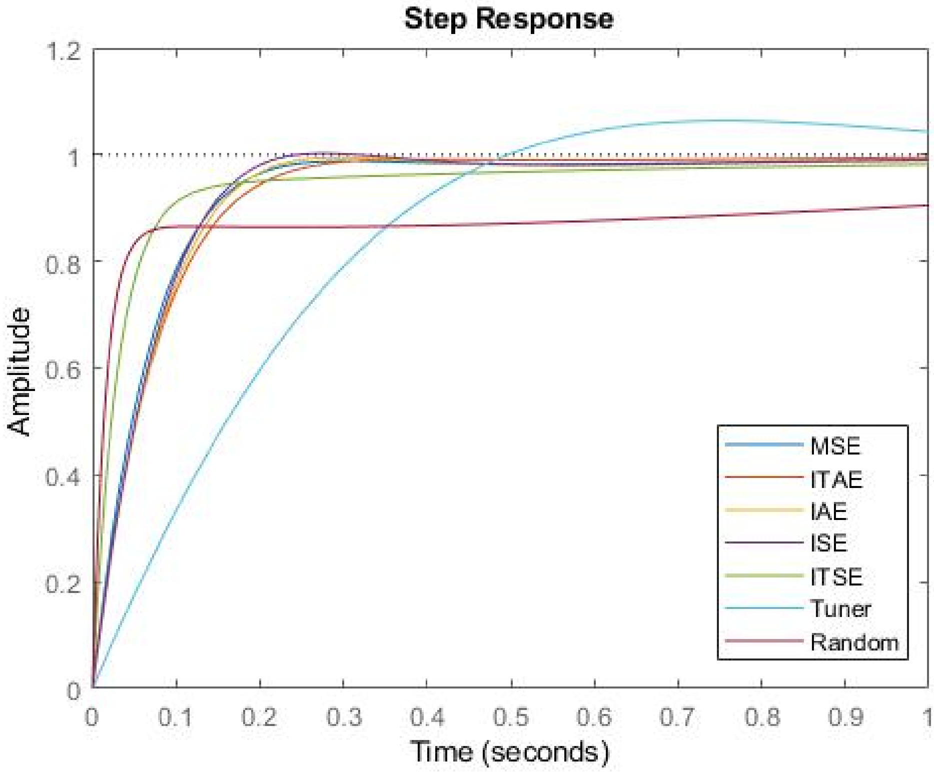

FIGURE 3.2: Responses for DE2 and MATLAB Tuner and Random Value

3.8 Convergence Speed Comparison

Convergence speed is an integral factor when tuning control systems. Convergence speed is defined by how fast does an evolutionary algorithm reach the optimal point. The best fitness value is noted every NFC/Np generations and a logarithmic plot is created. Each variant of DE is plotted and ranked based on how quickly the best fitness value is achieved. The ranks go from 1 through 12, 1 being the best and 12 being the worst, and each variant is ranked for all five SS error calculation methods.

Best rank meant that DE can find the best solution faster than any other in the competition. In case of a tie, both variants are given the same rank, and the next rank is skipped. The experiment is run 51 times, and the final averages are used to determine rank to eliminate stochastic bias.

Table 3.5: Rank of DE Variants in Convergence Comparison

Variant	IAE	ISE	ITAE	ITSE	MSE	AVG
DE1	10	10	10	9	11	10
DE2	12	9	3	7	8	7.8
DE3	11	7	5	5	5	6.6
DE4	7	5	1	6	9	7.6
DE5	1	2	2	1	1	1.4
DE6	9	8	9	12	10	9.6
DE7	2	3	1	8	2	3.2
DE8	6	6	8	4	12	7.2
DE9	8	12	12	10	4	9.2
DE10	5	11	7	3	7	6.6
DE11	3	1	4	2	3	2.6
DE12	4	4	6	11	6	6.2

3.8.1 Results and Analysis

Table 3.5 shows the average rank of each DE variant over the 5 Steady State Error methods. From Table 3.5, it is observed that DE5 is the best, followed by DE11. This comparison focuses on the convergence rate of the fitness function in Eq.(3.1) rather than individual responses (i.e., ST, RT, OS). This graph sheds great light onto the performance of various DE mutation schemes and which achieve the best results. On Average Non-distributed, DE methods reach a rank of 6.6, as compared to 6.36 of Distributed DE. One aspect which skewed result is DE5, which performs the best, is not part of the distributed mutation schemes.

3.9 Summary

It is apparent, from the results, that mutation scheme and distribution have an impact on control system tuning using DE. The overall best mutation scheme is DE5, which is described in Eq.(2.7). Though distribution brings an improvement in the final results, the improvement is not considerable enough to add a distributed framework to GDE3. The major purpose of this chapter is achieved, and the DE5 variant is henceforth used in GDE3 for control system tuning in further chapters.

Chapter 4

GDE3 and NSGA-II Comparison on LQR Based Aircraft Pitch Control

Generalized Differential Evolution (GDE) has not been used, to the best of the authors knowledge, to tune the controllers designed using the LQR technique. It is important to test a new algorithm on a well-defined benchmark case. The case study on aircraft pitch control is taken from [66] for this purpose. It has a clear definition, design goals, and implementation. GDE3 will be compared to conventional tuning methods as well as its genetic algorithm counterpart, NSGA-II. The contents of this chapter have been published in IEEE Congress on Evolutionary Computation 2019. The case study shows that GDE3 outperforms NSGA-II and other conventional tuning methods, and GDE3 outputs an optimal Pareto-front closer to the origin for this minimization problem

4.1 Introduction

A combination of two existing techniques are used to prove that the resulting algorithm performs better than the parent techniques or conventional methods. The parent optimization algorithm used in this paper is the DE [46]. For the current study, two conflicting objectives and multiple trade-off solutions are required to be found. Based on the situation, the designer, as a decision maker (DM), can choose the most appropriate solution. So enhancing LQR controller can be defined as a multi-objective optimization problem. Evolutionary computation (EC), being a powerful method, has been used to solve many real-world multi-objective optimization problems [67, 68]. The contribution of this chapter is providing a system designer with a set of optimal solutions for various conditions, which satisfy design requirements and conditions. In order to achieve an optimal Pareto-front, a variation of Non- dominated Sorting Genetic Algorithm (NSGA-II) is employed [9], which is Generalized Differential Evolution (GDE3). GDE3 uses differential evolution (DE) instead of genetic algorithm (GA).

Aircraft pitch control is used as the case study to compare the performance of GDE3 and NSGA-II, and to show the advantages of multi-objective optimization. The aircraft model and corresponding results are from a collaborative controls tutorial set-up by University of Michigan, Carnegie Mellon University, and University of Detroit Mercy [66]. GDE3 results are compared to the results obtained from [66] and NSGA-II, while keeping the same model to ensure fair comparison and obtain a concrete conclusion. The existing model from [66] is

run through the GDE3 optimizer to generate a set of optimal Pareto-front solutions. The optimizer ensures that the entire solution set is stable by ensuring that every value results in a system, which settles to 0. The dynamic responses, i.e., settling-time, rise-time and overshoot, are measured, for some values in the solution set, and compared to the response from [66] and NSGA-II.
The novelty of the proposed method is that it provides the system designer with an option when configuring an LQR controller. The method goes beyond the conventional optimization of using a weighted aggregation of multiple objectives, and results in a solution, which works but can be made much better. This case study demonstrates this hypothesis.

4.2 System Model

Evolutionary algorithms, by nature, are slow. Real-time systems, on the other hand, require faster algorithms to generate real-time results, which renders optimization unrealistic for online tuning. For change in each situation, the system has to be re-optimized after the weights are tuned, to get a new singular solution. The proposed approach provides a massive advantage against this traditional strategy. For LQR based systems, the proposed method provides the system designer an optimal set of solutions resulting from multiple design objectives. All of these solutions can be stored as a look-up table, with values based on environmental inputs. The LQR-based control system continues to function, as it would otherwise, while the decision system reads the environmental inputs. Based on the environmental inputs, the decision system selects appropriate solution to recalibrate the LQR gain in real-time.

Fig. 4.1 shows how this model works. This process is completely deterministic as all the solutions and their respective responses are already calculated offline. Similarly, all the environmental conditions, in which the gain would change, is predetermined by the system designer when creating the look-up table. The execution time analysis can be done on the system beforehand, to ensure it meets all standards. Active steering system for Articulated Heavy Vehicles (AHV) [3] is an example of a real-time system using an LQR controller [39, 69].

An active steering system is designed to improve the lateral stability at high speeds to ensure that the vehicle does not rollover [39, 69]. Similarly, it must enhance the low-speed maneuverability so that the vehicle can safely negotiate 90-degree turns at intersections. Both of these are conflicting objectives [3].Once optimal Pareto solutions are obtained, they can be loaded onto the decision system. Based on environmental requirement, the LQR gain can be changed using pre-optimized stable solutions. From Fig. 4.1, it can be ascertained that this method can be applied to any system with an LQR controller.

4.3 Multiple Objectives

Real-time systems must adhere to many restrictions, and achieve a variety of performance measures or objectives. Minimization of rise-time (RT) and settling-time (ST) are examples of two common conflicting objectives in control systems, which are incorporated in this case study. Evolutionary Algorithms, in their bare form, require a single fitness function based on one or many objectives. In other works, multiple objectives have been aggregated, using weights, into a single fitness function, which is then optimized to produce a single optimized

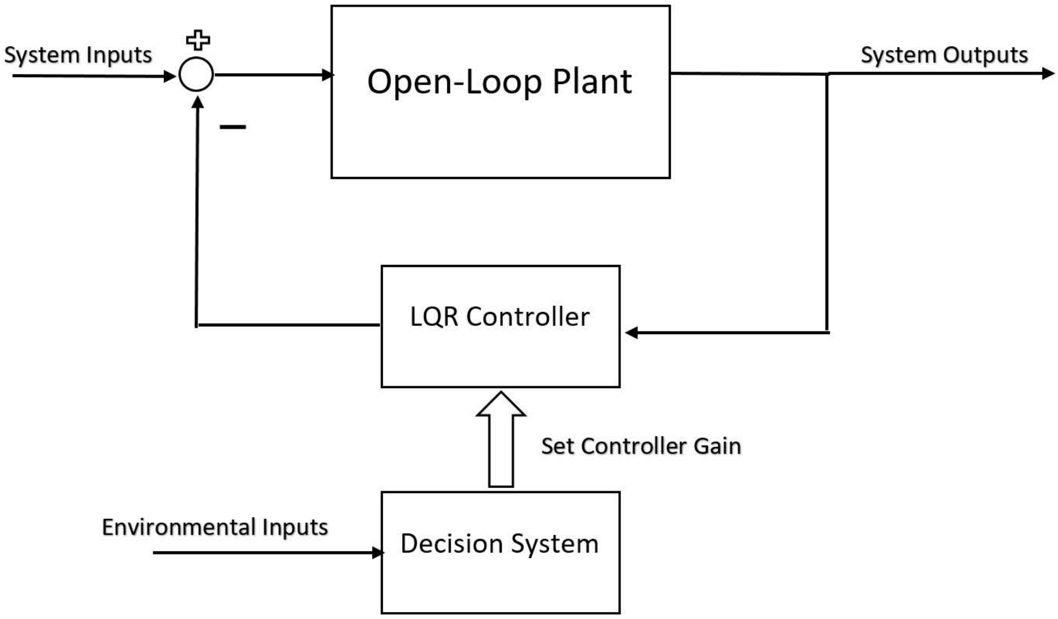

FIGURE 4.1: Feedback system model with LQR and decision support system. Open-loop plant is the aircraft pitch model. Controller gains are obtained by the algorithm, which can be dynamically changed by the decision system based on environmental factors.

solution [64]. A single aggregated weighted fitness function cannot achieve the convex part of the Pareto-front, otherwise, multiple varying weights might be required, which add a layer of complexity to it [9]. Secondly, assigning weight has to be completed before the algorithm begins. Search for a better result is not conducted after finding a satisfactory result. Posteriori decision-making can find a better solution, even though an acceptable solution is found. After completing the optimization process, both GDE3 and NSGA-II provide a set of optimal Pareto-front solutions [9]. The decision-maker can choose any solution that best fits the system, without having to rerun the experiment by adjusting weights assigned to each objective. A common example, the selection of airfare between point A and point B has two objectives: cost and time. If a ticket costs more, it has fewer connections, and less travel time will spend, and vice versa (generally). Cost and time, in the case of airfare, are conflicting objectives.

4.4 Ensuring Stability and Controllability

The Q weighting matrix (see Eq.(2.2)), in a LQR controller design, must be positive semi-definite, that is, all their eigenvalues must be greater or equal to 0 [70]. This algorithm ensures that for each iteration the generated Q matrix meets the criteria. In this experiment, the R matrix is set to 1, and the elements of the Q matrix are varied to tune the LQR controller. The optimizer ignores any selection, which cannot stabilize the system, and this ensures that all final solutions are stable.

In order to ensure that a system model is compatible with this method, it must be stable, observable, and controllable. In this case, the system model refers to the state space representation of a particular problem, including the selection of design parameters. The most common check for system stability is to check the poles; a dynamic system must have a finite number of non-negative poles to be controllable. One way to do this, in MATLAB, is to use the command $isstable$, which returns a value of 1 (true) if the system is stable and 0 (false) if the system is unstable. If the real part of the eigenvalues of matrix A is negative values, then the system can be deemed as asymptotically stable. Negative eigenvalues show that the system will tend to move back to the steady state. Controllability and observability can be checked to ensure the states of the system are controllable and observable. Controllability refers to how much the input can be controlled by manipulating the state. Observability indicates how much can be perceived from the output. In order to calculate Controllability, the rank of

TABLE 4.1: Control parameters of Optimization

Parameter	Value
Population Size (P)	60
Crossover Probability (C_r)	0.95
Scaling Factor (F)	$0.3 < F < 0.9$
Number of Objectives	2
Range of x_i	$0.01 < x < 10$

$[A...A^{n-1}B]$ must be equal to the size of the Matrix A, where the matrix B is the input-to-state matrix (see Eq.(2.1)). For calculating observability, the rank of $[C^T...CA^{n-1^T}]$ has to be equal to the size of the matrix A, where the matrix C is the state-to-output. All the states in the state space models are controllable and observable, this is ensured by using minimal realization or pole-zero cancellation [66]. Each gain matrix obtained by the multi-objective tuning algorithm is simulated and tested to ensure that the output settles to origin and is within the boundary requirements, this ensures output stability.

4.5 Optimization Parameters and Pre-compensation

For the aircraft pitch control using LQR, the same DE parameters are used, which are listed in Table 4.1. The parameters listed in Table 4.1 are obtained from [47], in which the authors do rigorous testing to show that these produc better results. The experiments has been run for 51 times to minimize the impact of stochastic effects on the reported results.

When a full-state feedback system is created, the actual output of the system and the desired output as seen in the actual model can vary and requires scaling [66]. The output received by the optimal controller is not compared to the reference, rather it compares $States*Kx$ to the reference [66]. This is done as per the full-state feedback system instructions in [66], to ensure a fair comparison. After the results are obtained, the pre-compensation procedure listed in [66] is followed to scale the output of the control system to match the desired output. The scaling factor does not affect the objectives, by using pre-compensation, and fair comparison among competing algorithms is ensured.

4.6 Complexity of Case Study

The Aircraft Pitch Control is a complex case study. The case study is chosen as it is similar to the ATS controller design problem tackled in the thesis. Two aspects determine the complexity of the system: size of the system matrix A and decision variables in gain matrices Q and R. The system matrix in the case study is $3 * 3$ and decision variables in Q and R are nine. An adaptive cruise control has been modeled by a $3*3$ system matrix [71] with nine decision variables and a simple cruise control can be modeled as a $1 * 1$ system matrix with 1 decision variable [66]. An active steering system for an articulated heavy vehicle can be modeled by a $4 * 4$ system matrix with 4 decision variables [25, 72]. A two-wheel self-balancing robot is modeled by a $4 * 4$ system matrix [73] with 2 decision variables. A quarter car suspension system can be modelled by a $4 * 4$ matrix with 4 decision variables [17, 66]. This shows that the complexity of the selected systems matches or exceeds many other comparable systems.

4.7 LQR Controller for Aircraft Pitch Control

An aircraft in flight is a complex dynamic system and may exhibit three translation and three rotational motions [74]. The case study focuses on one of these rotational motions of the aircraft, i.e., pitch around the lateral axis [74]. To have an effective comparison, the aircraft model given in [66] is used to optimize and investigate the method's performance. The state space model for the aircraft is defined by Eqs.(4.1) and (4.2). The model used in [66] assumes that the aircraft is in steady-cruise at constant altitude and velocity, the thrust, drag, weight and lift forces balance each other in x and y directions, pitch angle does not affect the speed of the aircraft. Implementing an LQR as a feedback controller does not skew comparison results as it is same for the both cases. The input is a step reference of 0.2 rad, where the following criteria must be met: overshoot less than 10%, rise-time less than 2 seconds, settling time less than 10 seconds [66]. The minimization objectives are rise time and settling time. Fig. 4.2 shows the full states-space feedback control for aircraft pitch control used in the case study [66] with the decision system, where EI corresponds to environmental inputs that affect the decision-making process, for example considering the aircraft speed. Where $x = [\alpha, q, \theta]'$ is the state vector, θ_{des} is the reference (r), $\delta = (\theta_{des} - Kx)$ is the control input (u), θ is the output (y) and K is the control gain matrix. K is obtained by the LQR method in the case study.

$$f'(x) = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix} \begin{bmatrix} \delta \end{bmatrix} \tag{4.1}$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} \delta \end{bmatrix} \tag{4.2}$$

$$Q = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} \tag{4.3}$$

4.7.1 Experimental Results

Fig. 4.3 shows the optimal Pareto-front obtained by GDE3 and NSGA-II. Each point is a candidate solution with two objective values, ST and RT. Each solution corresponds to a control gain matrix of the LQR controller. The decision maker can pick any solution from the optimal Pareto set, each solution is stable and optimized and gained in a single optimization run. Figs. 4.4, 4.5, 4.6 the show responses with least settling time, least rise time and knee-point, respectively. All solutions are well fitted within the design requirements. Fig. 4.7 is the response, after conducting optimization process from [66].

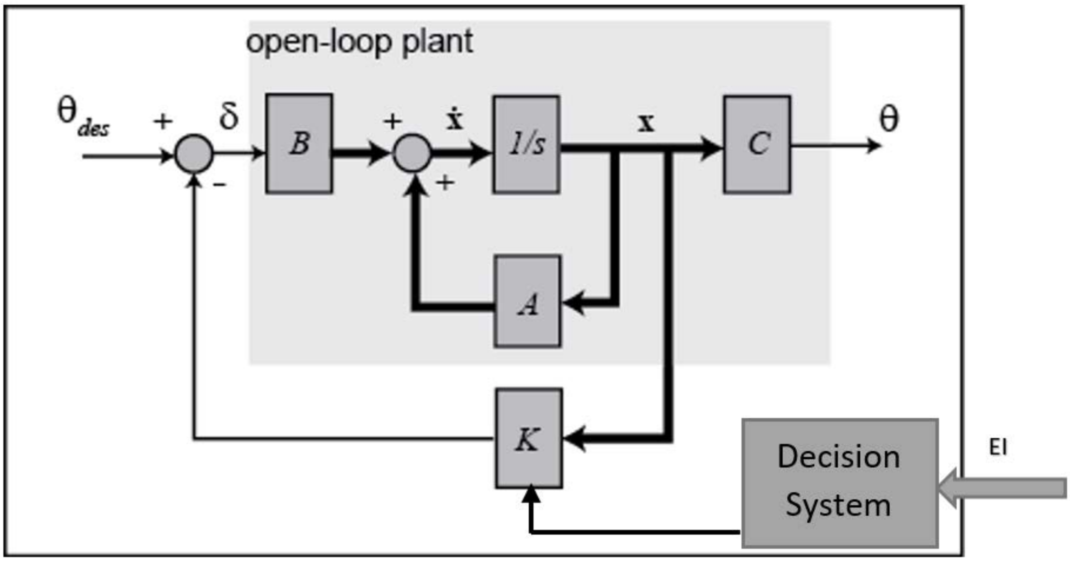

Figure 4.2: Feedback system model for aircraft pitch control with GDE3

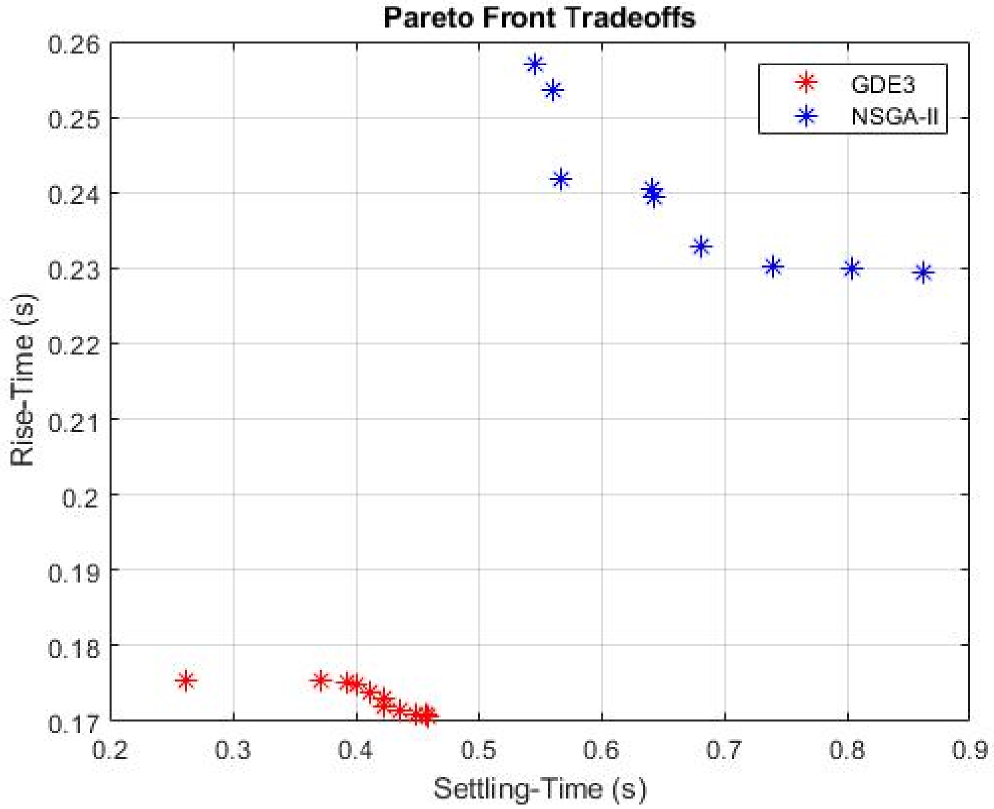

Figure 4.3: Pareto-front obtained by GDE3 and NSGA-II on aircraft pitch control model. (Note: The designer can choose any LQR gain matrix value from the Pareto-front, based on ST and RT requirements.)

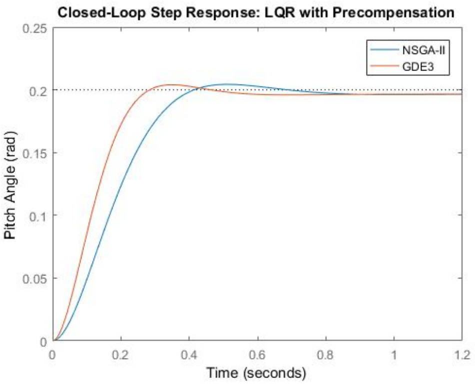

Figure 4.4: Response of the aircraft pitch control system with ST prioritized.

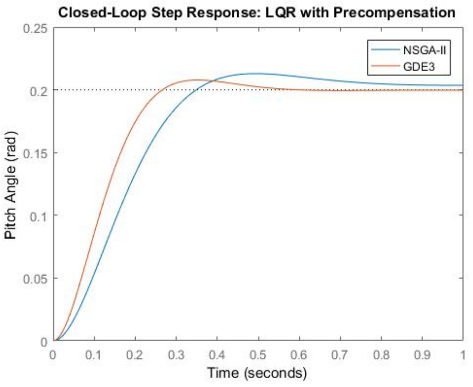

Figure 4.5: Response of the aircraft pitch control system with RT prioritized.

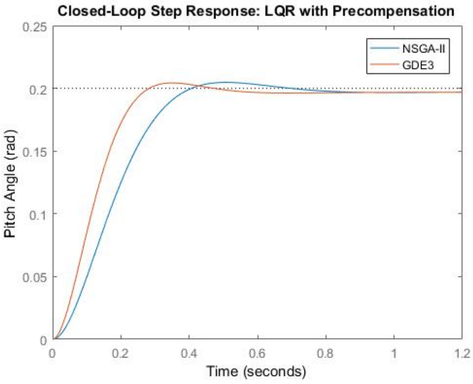

Figure 4.6: Response of the aircraft pitch control system with knee-point.

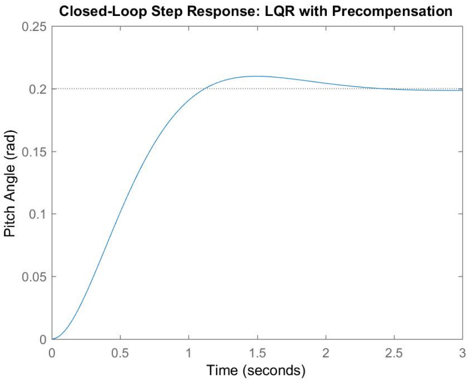

Figure 4.7: Response of the aircraft pitch control system obtained from [66].

4.7.2 Analysis

Table 4.2 proves the initial hypothesis. NSGA-II outperforms conventional tuning methods but GDE3 outperforms both NSGA-II and conventional tuning in RT, ST, and Overshoot at both extremes and knee points. Table 4.2 also shows that none of the points violated the design constraints. GDE3 is 75.9% better than conventional and 30.9% better than NSGA-II for RT at knee-point. GDE3 is 81.6% better than conventional and 33.7% better than NSGA-II for ST at knee-point. The same pattern continues for the extreme points as well. The best RT achieved by GDE3 is 76.6% better than conventional and 25.7% better than NSGA-II. The best ST achieved by GDE3 is 87% better than conventional and 52% better than NSGA-II. The best overshoot achieved by GDE3 is 59.3% better than conventional and 7.3% better than NSGA-II. The knee-point is chosen in the comparison, as it reflects an aggregated fitness function with equal weights assigned to both objectives, which is a common scenario. As observed, both evolutionary algorithms perform better than conventional method. From Fig. 4.3, it is apparent that all non-dominated solutions of GDE3 dominate the non-dominated solutions of NSGA-II.

Table 4.2: Comparison Aircraft Pitch Control

Method	RT(s)	ST(s)	Overshoot (%)
Conventional Tuning [66]	0.7283	2.0181	4.9100
NSGA-II Knee-point	0.2537	0.5605	2.3425
NSGA-II ST	0.2570	0.5467	2.1578
NSGA-II RT	0.2295	0.8621	6.4460
GDE3 Knee-point	0.1753	0.3718	2.1369
GDE3 ST	0.1754	0.2625	2.0000
GDE3 RT	0.1706	0.4595	3.8928

4.8 Summary

This is an important case study before using the algorithm in expensive optimization of the ATS controller for car-trailer combination as this proves the viability of GDE3. In Chapter 5, the same algorithm is utilized to optimize the ATS controller for a car-trailer combination, but there is no reference design available for comparison to gauge the improvement that is brought by the GDE3 algorithm. Using an existing well-established case study, with complete experimental detail available in [66], this section proves the practicality of MOEA algorithms, e.g. GDE3 and NSGA-II.

The GDE3 provides both an optimized result and a solution set to allow for online decision-making in real-time systems. The experimental results for aircraft pitch control prove that multi-objective optimization of an LQR controller is possible, providing both choice and better results than conventional methods and NSGA-II. This method can be applied to any physical model, which can be mathematically represented and can be controlled by LQR integration. As long as the designer can provide an accurate state space representation and conflicting objectives, this scheme can be used to tune the LQR controller by manipulating Q and R matrices. These matrices will change with the physical model as would the complexity of the problem.

Stability and controllability of the physical model are ensured. The algorithm is a not model oriented scheme and hence, by inference, should provide similar results for well defined, real-time systems. The aircraft pitch control provids a real-time example. This chapter of the thesis concentrates on proving GDE3 as a viable method for offline optimization for LQR controller in safety-critical systems, which not only improves system performance, but also allows for online selection of optimized solution based on environmental factors. . In the next chapters, these techniques are applied to the articulated vehicles for ATS controller design.

Chapter 5

Design Optimization of LQR-based ATS Controller Using GDE3

Chapter 3 and 4 serve as the foundation for the design optimization of the LQR-based ATS controller for car-trailer combination using GDE3. This chapter utilizes GDE3 to tune the LQR-based ATS controller a car-trailer combination represented by a 3-DOF yaw-plane model. The tuning is carried out at varying speeds, and for each speed, an optimal Pareto-front is generated. Each point in the Pareto-front cluster corresponds a control gain matrix K . This point cluster may serve as a look-up table for the gain scheduling controllers. The contents of this chapter have been published in SAE2019 World Congress [72].

5.1 Introduction

This chapter presents a robust ATS controller for car-trailer combinations. ATS systems have been proposed and explored for improving the lateral stability and enhancing the path-following performance of car-trailer combinations.

Most of the ATS controllers were designed using the LQR technique. In the design of the LQR-based ATS controllers, it was assumed that all vehicle and operating parameters were constant. In reality, vehicle and operating parameters may vary, which may have an impact on the stability of the combination. For example, varied vehicle forward speed and trailer payload may impose negative impacts on the directional performance of the car-trailer combination. Thus, the robustness of the conventional LQR-based ATS controllers is questionable. To address this problem, a LQR-based ATS controller with a look-up table gain-scheduling is proposed. In the design of the proposed ATS controller, at each operating point, the ATS controller is designed using the LQR control technique. At an operating point between two established adjacent operating points, the control gain matrix of the controller is determined using a linear interpolation method. To further improve the directional performance of the car-trailer combination, the weighting matrices of the LQR controller are determined using an optimization algorithm, namely, GDE3. The effectiveness of the proposed ATS controller is demonstrated using numerical simulation based on a car-trailer model.

5.2 Rearward Amplification (RWA)

Each AV may experience different operating conditions, e.g., traveling on a highway at a high speed or negotiating a curved path at a low speed. From the view of vehicle design, the criteria are different under the aforementioned operating conditions. When the AV is travelling at high speeds, it requires to be laterally stable. In the case of losing lateral stability, an AV may experience one of the following unstable motion modes: trailer sway, jackknifing, and roll-over. To evaluate the lateral stability of an AV at high speeds, rearward amplification (RWA) is a well-accepted performance measure [75].

The RWA measure may be determined under an evasive maneuver at a high speed. During an evasive maneuver, e.g., a single lane-change maneuver, the RWA may be defined as the ratio of the peak lateral acceleration observed at the center of gravity of the trailer and the leading vehicle unit [43]. An ideal RWA measure is 1.0, which is extremely difficult to achieve in real-world operating conditions. Physically, when the RWA takes the value of 1.0, the trailing vehicle unit will have similar dynamic behaviors as the leading vehicle unit. In other words, with the RWA measure of 1.0, the AV will have better path-following capability, and the driver may well control the vehicle to achieve high lateral stability [43]. Controlling lateral acceleration amplification and achieving the RWA measure of 1.0 is the primary objective. In addition, controlling the RWA ensures a higher lateral acceleration considering the rollover threshold value. Articulated vehicles tend to have a low rollover threshold, as low as $0.6g$ [37]. Since the RWA is evaluated in the linear dynamics of the vehicle, under testing evasive maneuvers, the peak lateral acceleration of the leading vehicle unit is set around $0.30g$. Equation (5.1) is used to calculate the RWA ratio:

$$RWA = \frac{|\,Peak\ lateral\ acceleration\ of\ the\ trailer\ at\ CG\,|}{|\,Peak\ lateral\ acceleration\ of\ the\ car\ at\ CG\,|} \tag{5.1}$$

5.3 Path-Following Off-Tracking

At low speeds, the trailer must follow the same trajectory as the leading vehicle unit. If the trailer is unable to follow the same path as the car, there is a great chance for the trailer to violate the boundary of the road. The measure to check the trailer ability to track the path of the car is called Path-Following Off-Tracking (PFOT) [76]. PFOT is defined as the maximum radial offset between the path of the center of the car front axle and that of the center of the trailer rear axle over an evasive maneuver. Accurate measurement of PFOT can be done by comparing the difference during a circular motion of an AV. If a controller is optimized for high-speed RWA, PFOT performance at low speeds may be degraded and vice-versa [3]. This makes this problem an optimization problem, where the parameters of the ATS controller are optimized.

5.4 Design ATS Controller

The design of the ATS controller involves modelling of the car-trailer combination and the implementation of the control system. To evaluate the resulting ATS system, a single lane-change maneuver is simulated. The proposed ATS system is designed based on a few performance measures, e.g., the lateral acceleration of the leading unit and the RWA of the combination.

A multibody dynamic software package, known as Equation of Motion (EOM) [77], is used to derive the linear 3-DOF yaw-plane car-trailer model. Once the derived model is verified, EOM is used to develop and simulate the ATS performance. The main purpose of using the ATS system is to control and reduce the lateral acceleration of the combination to ensure that the trailers lateral acceleration is not amplified compared to that of the car. In other words, a reduction in the RWA value will lead to a better dynamic performance of the combination [78].

The control method used in this study is known as LQR. The LQR-based ATS controller is optimized using GDE3, which is a MOEA. The LQR-based ATS controller and GDE3 will be discussed in the following sections.

5.4.1 3-DOF Yaw-plane Car-trailer Model

In this research, the mathematical model is derived and validated with the published models [5, 37]. In this model, each axle is represented by a single wheel, assuming that both tires on each axle have the same dynamic characteristic, i.e., the relationship between the tire slip angle and the cornering force. Fig. 5.1 shows the schematic representation of the car-trailer combination using the 3 DOF yaw-plane model.

For the yaw-plane model, the lateral and yaw motions of the car, as well as, the yaw motion of the trailer are considered. The governing equations of motion of the car are expressed as:

$$m_1(\dot{U} - Vr) = -X_1 cos\delta - X_2 + X \tag{5.2}$$

$$m1(\dot{V} + Ur) = f_1(\alpha_1) + f_2(\alpha_2) + X_1 sin\delta - Y \tag{5.3}$$

$$I_1\dot{r} = af_1(\alpha_1) - bf_2(\alpha_2) + \alpha X_1 sin\delta + dY \tag{5.4}$$

Eq.(5.2) to (5.4) govern the longitudinal, lateral, and yaw motions of the car,

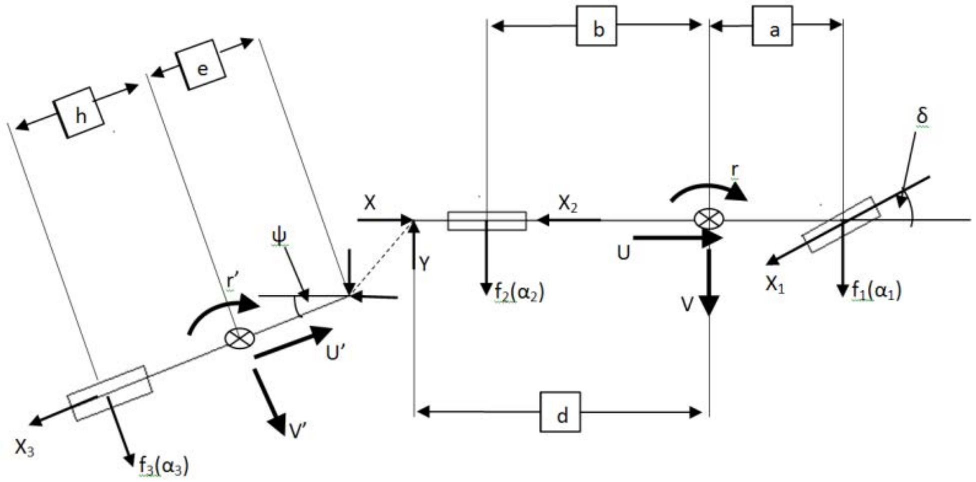

FIGURE 5.1: Schematic representation of the 3-DOF yaw-plane car-trailer model

respectively. Where m_1 is the mass of car, U is forward speed of car, V lateral speed of car and r yaw rate of car, δ is the front wheel steering angle, X_i are the longitudinal tire forces, α_i are the slip angles of the tires, f_i are the cornering stiffness of the tires, X is the longitudinal hitch force, Y is the lateral hitch force, I_1 is the moment of inertia of car and a, b, c and d are described in Table 5.1.

Similarly, the governing equations of motion for the trailer are shown as follows:

$$m_2(\dot{U}^{'} - V^{'}r^{'}) = -X_3 - Ysin\psi - Xcos\psi \tag{5.5}$$

$$m_2(\dot{V}^{'} + U^{'}r^{'}) = f_3(\alpha_3) + Ycos\psi - Xsin\psi \tag{5.6}$$

$$I_2\dot{r}^{'} = -hf_3(\alpha_3) - e(-Ycos\psi + Xsin\psi) \tag{5.7}$$

Eqs.(5.5) to (5.7) govern the longitudinal, lateral, and yaw motions of the trailer, respectively. Where m_2 is the mass of trailer, $U^{'}$ forward speed of trailer, $V^{'}$ lateral speed of trailer, $r^{'}$ is the yaw angle of trailer, I_2 is the moment of inertia of trailer and h and e are described in Table 5.1.

Once a mechanical hitch is introduced, the kinematic constraint is active. To simplify the model, the forward speed of the car is assumed constant. In addition, the articulation angle ψ assumed to be small, which leads to Eq.(5.8) to (5.10). Detail is available in [72].

$$cos(\psi) \approx 1 \tag{5.8}$$

$$sin(\psi) \approx \psi \tag{5.9}$$

Also, the following equation is determined at zero initial conditions.

$$\psi = r - r^{'} \tag{5.10}$$

Where ψ is the articulation angle, r is the yaw rate of car and $r^{'}$ is the yaw rate of the trailer. The notation is shown in Figure 5.1, and the values of the parameters used throughout this study are listed in Table 5.1.

Once the model is derived, EOM software package is used to validate the model by comparing the responses of both models under the same single lane-change maneuver. Both models generate the identical dynamic response, which proves that the EOM model matches the derived equations. Thus, the EOM software package is selected to design the ATS controller. However, in order to further enhance the model, more essential components are added to the system, e.g., an actuator is installed on the trailers axle, and an accelerometer is installed at

Table 5.1: Car-trailer combination parameters for ATS controller design

Name	Symbol	Value
Car Mass	m_1	1700kg
Car yaw inertia	I_1	2000kgm^2
Trailer mass	m_2	2000kg
Trailer yaw inertia	I_2	3000kgm^2
Distance between CG and front axle	a	1.5m
Distance between CG and rear axle	b	1.7m
Distance car CG to contact with trailer	d	2.9m
Distance trailer CG to contact with car	e	3.8m
Distance between the CG and the rear axle	h	0.2m
Height of vehicle CG	$H1_{CG}$	0.325m
Height of trailer CG	$H2_{CG}$	0.676m
Combined front tires cornering force	c_1	80000$\frac{Nm}{rad}$
Combined rear tires cornering force	c_2	80000$\frac{Nm}{rad}$
Combined trailers tires cornering force	c_3	80000$\frac{Nm}{rad}$

the Centers of Gravity (CG) of the leading and trailing units, respectively. The actuator is used to produce torque to steer the wheels on the trailer axle. In addition, the trailer forward speed should be the same as the car forward speed as the assumption made previously.

5.5 Single-lane Change Testing Maneuver

To conduct the designated RWA analysis, an open-loop testing procedure with a single lane change (SLC) maneuver is simulated. A single-cycle sine-wave steering input with adjustable frequency and amplitude represents the input for the SLC testing maneuver [79]. In this thesis, the frequency and the amplitude are maintained constant at 0.4hz and 2.6$degrees$, respectively. Fig. 5.2, shows the steering input used to test the performance measures of the combination.

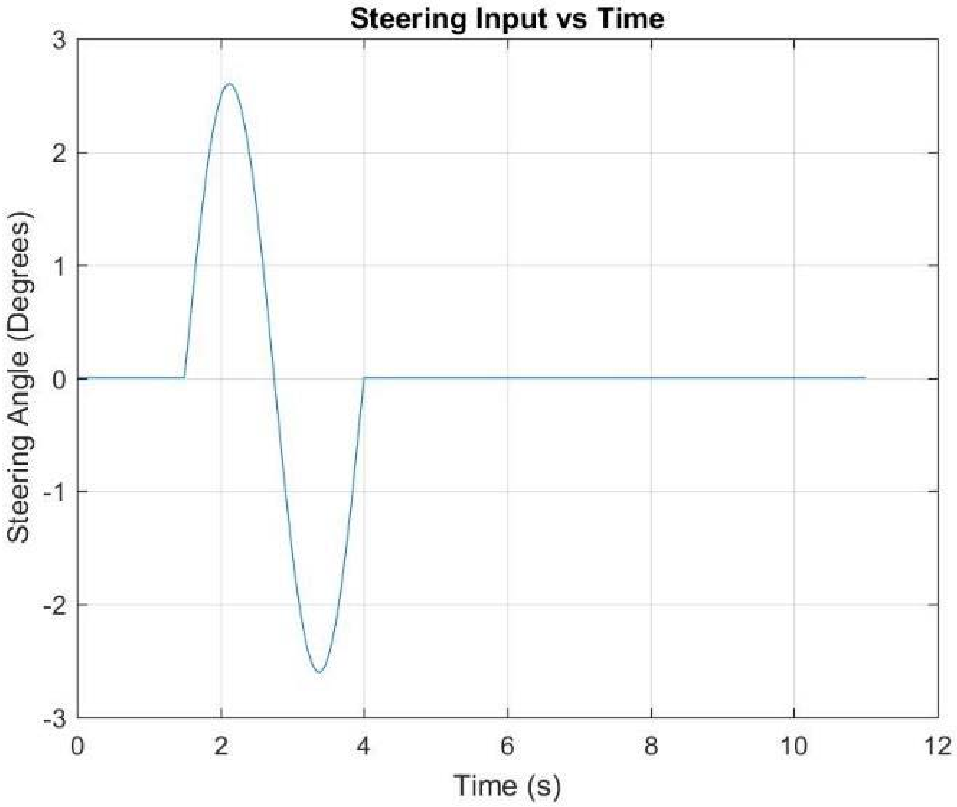

FIGURE 5.2: A representation of an SLC steering input

5.6 Testing Methodology

In previous chapters, details have been discussed, which have led to the use of GDE3 as the MOEA for tuning the LQR controller. Thus, the same algorithm is used to tune the LQR-based ATS controller for the car-trailer combination. A SLC maneuver is simulated to obtain lateral acceleration of the car and trailer at various speeds. This test is performed in three cases: baseline design (without control system), the LQR-based controller, and GDE3 optimized LQR controller (GDE3 optimized control system). During the SLC maneuver, the control system uses the actuator on the trailers axel to steer the wheels for generating the required cornering force to stabilize the trailer. The optimization algorithm reads the simulation results, for each speed, and changes the gain matrices of the LQR controller to improve performance. Thus, the gain-scheduling scheme considering the variation of the vehicle forward speed can be implemented. The fitness test performance measures for each solution are the RWA and peak-lateral acceleration of the car. The algorithm minimizes the value of peak-lateral acceleration of the car, while trying to maintain an RWA of 1.0.

5.7 Results and Discussions

This section presents the critical speed analysis and the simulation results of the car-trailer combination with and without the ATS system.

5.7.1 Critical Speed Analysis

As defined earlier, the critical speed is determined as the stability boundary of the car-trailer combination. Before testing the ATS system, the critical speed for the baseline model is investigated. For unbiased testing, the same steering input, as seen in Figure 5.2, is used to obtain simulation results at every speed. The forward speed of the combination is the variable in this procedure and is increased with an increment of $10km/h$. It is observed that the critical speed for the baseline system (without the ATS control) is $79.2km/h$. Figures 5.3 and 5.4, show the response of the leading and trailing units, respectively, at critical speed without ATS controller.

From Figures 5.3 and 5.4, the lateral acceleration of the car is around the specified boundaries, which means the combination is stable up to this speed. Furthermore, by looking at Figure 5.4, the lateral acceleration of the trailer is larger

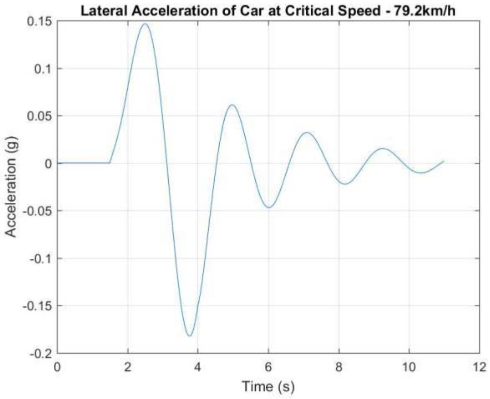

FIGURE 5.3: Lateral acceleration of the car at the critical speed 79.2km/h (without ATS controller)

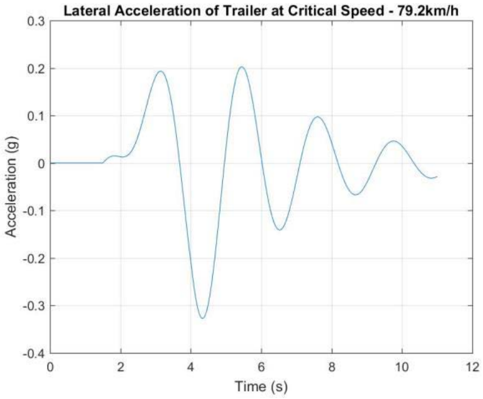

FIGURE 5.4: Lateral acceleration of the trailer at the critical speed 79.2km/h (without ATS controller)

in comparison to the lateral acceleration of the car. Thus, the trailer has an amplified lateral acceleration, which is also known as the RWA. The RWA of the combination is calculated to be about 1.8. In other words, the trailers lateral acceleration is 1.8 times greater than the cars lateral acceleration.

5.8 Forward Speed Variation

In this section, selected simulation results are compared, which are derived from the baseline design (passive system without control), the LQR-based controller, and the optimized LQR controller based on GDE3. Figs 5.5 to 5.8 show the optimal Pareto-front for speeds from 80km/h to 110km/h. The $x-axis$ shows the error from the ideal RWA of 1 and the $y-axis$ shows the peak lateral acceleration of the trailer.

Figs. 5.9 to 5.16 show the lateral acceleration of the car and trailer over the simulated SLC maneuver at the speed of 80km/h to 110km/h. The blue curve represents the passive system, the orange curve illustrates the ATS equipped with

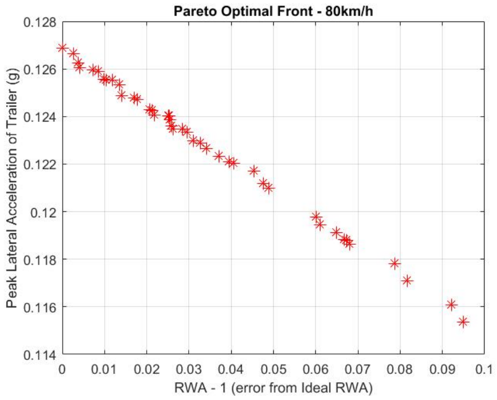

FIGURE 5.5: Optimal Pareto-front at 80km/h

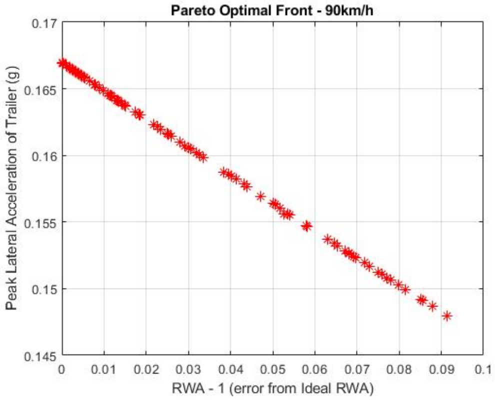

FIGURE 5.6: Optimal Pareto-front at 90km/h

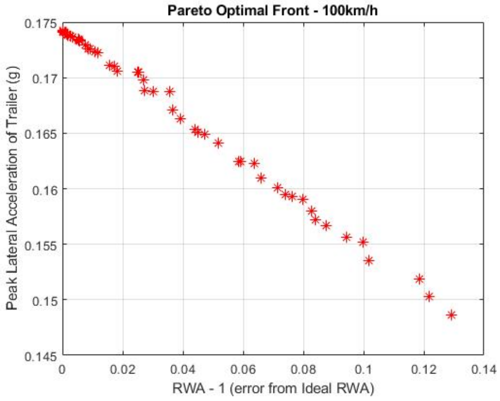

FIGURE 5.7: Optimal Pareto-front at 100km/h

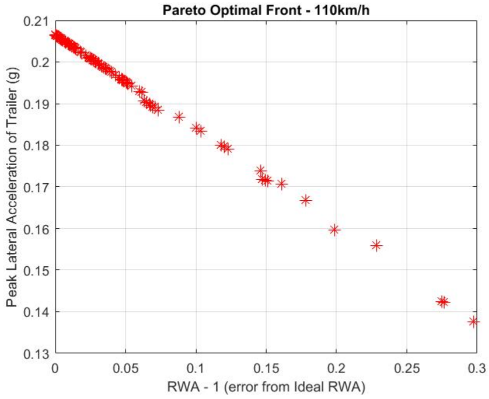

FIGURE 5.8: Optimal Pareto-front at 110km/h

the LQR-based controller, and the black curve shows the response of the optimized LQR controller. Compared with the passive system, the models with ATS systems obviously suppress the peak values of the lateral accelerations. It can be found that the lateral acceleration of the trailer is amplified in the passive system when compared to the cars lateral acceleration. However, by looking at the trailer equipped with the LQR and the optimized LQR controllers, it can be seen that the lateral acceleration is significantly suppressed as anticipated.

From Figs. 5.9 to 5.16 it is observed that the LQR-based controller over-suppresses the lateral acceleration of the trailer under the SLC maneuver. The over-suppressed lateral acceleration of the trailer will degrade the path-following performance

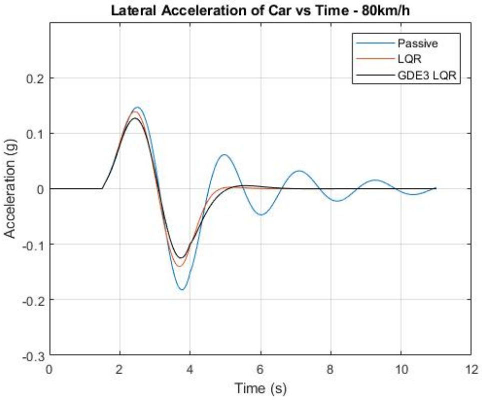

FIGURE 5.9: Lateral acceleration of the car at $80 km/h$

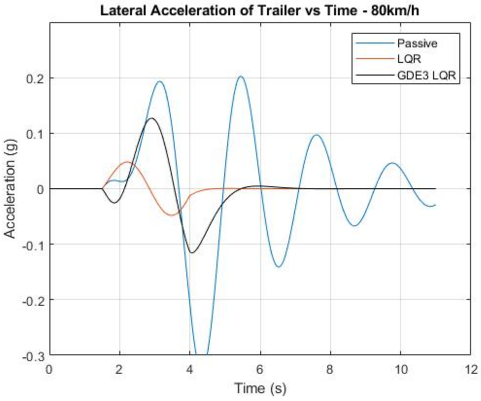

FIGURE 5.10: Lateral acceleration of the trailer at $80 km/h$

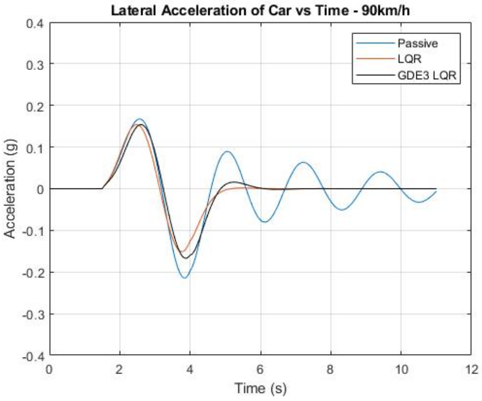

FIGURE 5.11: Lateral acceleration of the car at $90 km/h$

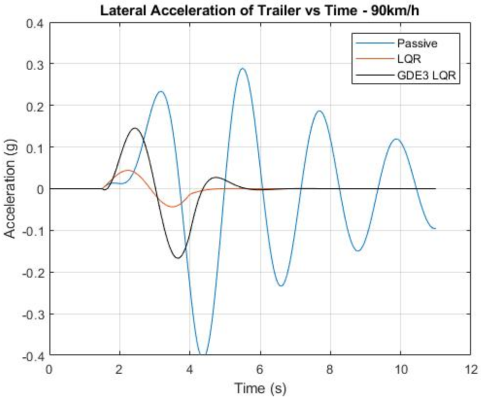

FIGURE 5.12: Lateral acceleration of the trailer at $90 km/h$

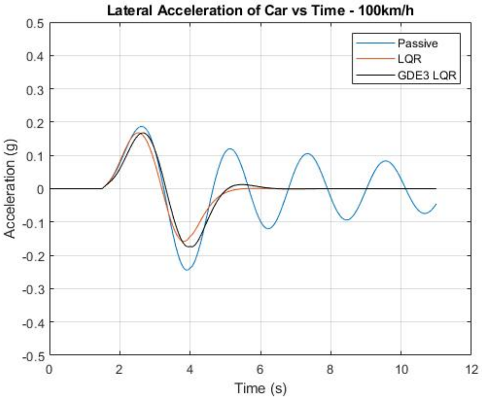

FIGURE 5.13: Lateral acceleration of the car at $100 km/h$

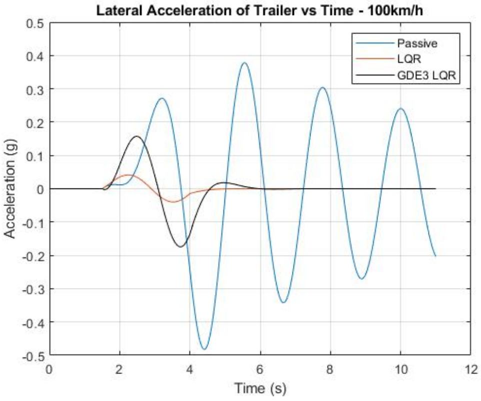

FIGURE 5.14: Lateral acceleration of the trailer at $100 km/h$

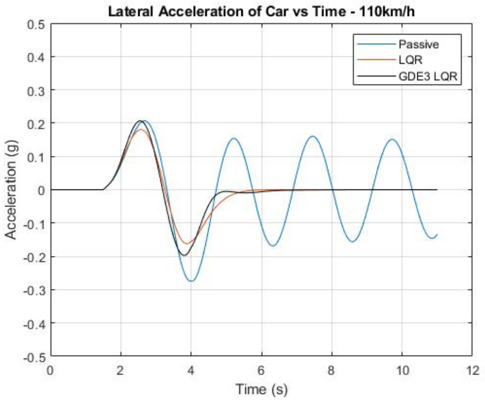

FIGURE 5.15: Lateral acceleration of the car at $110 km/h$

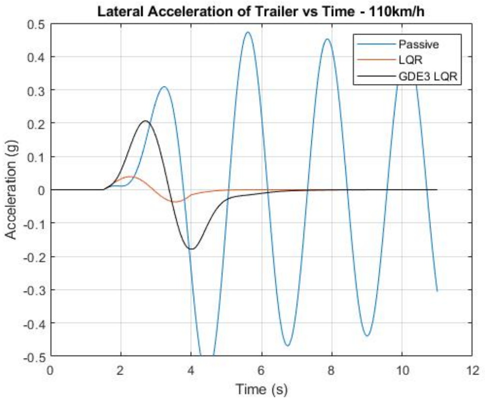

FIGURE 5.16: Lateral acceleration of the trailer at $110 km/h$

TABLE 5.2: RWA Result Comparison.[†]

Speed (km/h)	Passive	LQR	GDE3 LQR
80	1.7959	0.3435	1.0001
90	1.8988	0.2857	1.0000
100	1.9745	0.2462	1.0000
110	2.0337	0.2163	1.0000

[†] Ideal value of RWA is 1. RWA is calculated by applying Eq.(5.1) to Figs. 5.9 to 5.16.

TABLE 5.3: Peak Lateral Acceleration (g) of Trailer Result Comparison

Speed (km/h)	Passive (g)	LQR (g)	GDE3 LQR (g)	% Improvement[†]
80	0.3271	0.0481	0.1269	61.20
90	0.4068	0.0440	0.1670	58.95
100	0.4823	0.0412	0.1742	63.88
110	0.5575	0.0391	0.2065	62.96

[†] The % improvement is calculated w.r.t the passive system. Peak lateral acceleration should be below $0.3g$. $\%Improvement = |\frac{V_{new} - V_{old}}{V_{old}}| * 100$, where V is the value of lateral accelration.

of the combination, resulting in a less lateral displacement than the expected under the SLC maneuver. In contrast to the case of the LQR-based controller, the optimized LQR controller manipulates the lateral acceleration of the trailer, and the RWA is effectively controlled around the value of 1.0, at which the lateral stability and the path-following performance of the combination can be improved simultaneously.

Table 5.2 lists the RWA values of the system for speeds from $80km/h$ to $110km/h$ to illustrate a comparison and show the percentage improvement achieved after using a tuned controller. Table 5.3 shows the peak lateral acceleration of the trailer at different speeds. These two measures are the objectives, which the algorithm is tuning and are values of interest. From the aforementioned two tables, it is clear that GDE3 tuned ATS controller performs by the far the best. It not only avoids the pitfalls of over suppression, but ensures a more stable system as well. These results are consistent with the results published in [79] The above simulation results demonstrate that the optimized LQR controller can well control the lateral motions of the leading and trailing vehicle units to achieve desired and robust directional performance under the obstacle avoidance maneuver at varied forward speeds. The supervisor performance of the optimized LQR controller is attributed to the application of the gain scheduling scheme and the optimization algorithm, GDE3.

5.9 Summary

Comparing the numerical results shown in Tables 5.2 and 5.3 reveals the following facts: (1) for the baseline design, the RWA ratio decreases with the vehicle forward speed, which is consistent with the published results [79]; (2) the LQR-based controller over-suppresses the lateral accelerations of both the leading and trailing vehicle units, leading to an excessive small RAW ratio and poor path-following performance; (3) the optimized LQR controller can well control the lateral motions of both the leading and trailing vehicle units, achieve a robust RWA ratio of 1.0.

This chapter presents an innovative ATS controller, designed using the LQR technique, the GDE3 optimization algorithm, and the first step towards a gain scheduling scheme. Numerical simulations are conducted to test and validate the effectiveness of the proposed robust ATS controller in terms of the directional performance of a car-trailer combination under a single lane change maneuver at varied forward speeds. Simulation results demonstrate that the proposed ATS controller can well control the lateral motions of both the leading and trailing vehicle units, effectively manipulate the RWA ratio around the value of 1.0, and eventually achieve robust and improved lateral stability and path-following performance. In Chapter 6 this concept is taken further to develop a new gain scheduling controller.

Chapter 6

Design Optimization of Gain Scheduling Controllers Using GDE3 and Closed-loop Co-Simulation

For a dynamic system model, a state-space representation is required to design an LQR-based controller. In the design of the LQR-based controller, the weighting matrices, Q and R, should be determined in order to get the optimal control gain matrix, K. In Chapter 5, a LQR-based ATS controller is optimized using GDE3. In this chapter, a gain scheduling ATS controller is designed using GDE3 and closed-loop co-simulation. This method is proposed to generate a two-dimensional lookup table, using driver model reaction time and vehicle forward speed, for a gain-scheduled ATS system.

6.1 Introduction

Car-Trailer combinations exhibit highly non-linear dynamics. In Chapter 5, the car-trailer combination is linearized based on certain assumptions [72], which leads to a 3-DOF car-trailer model. This model is then utilized to design and evaluate an ATS controller. The open-loop dynamic simulation is carried out using MATLAB/Simulink without a driver model. Due to the absence of a driver model, a closed-loop SLC maneuver with a given trajectory cant be conducted. Secondly, the 3-DOF linear model neglects some important dynamics of the real-world car-trailer combinations, e.g., the lateral load transformation. Thus, the fidelity and accuracy of the 3-DOF yaw-plane model may not be desired. To address the above concerns, a new gain scheduling ATS controller is designed using GDE3 and closed-loop co-simulation.

This chapter is an important starting point towards optimization using co-simulation by combining the nonlinear car-trailer model developed in CarSim and the controller designed in Simulink/Matlab. There is one system limitations in using he CarSim model for the design optimization. The CarSim Model can at most cater to 1500 generations by the optimizer. The simulation, in contrast, may run for 100, 000 generations. This greatly reduces the time and opportunity for the algorithm to search the solution space. This limitation is most probably a system specific issue, and in no way is a general problem with CarSim software. Nevertheless, this limitation imposes the constraint for the maximum generations to be achieved.

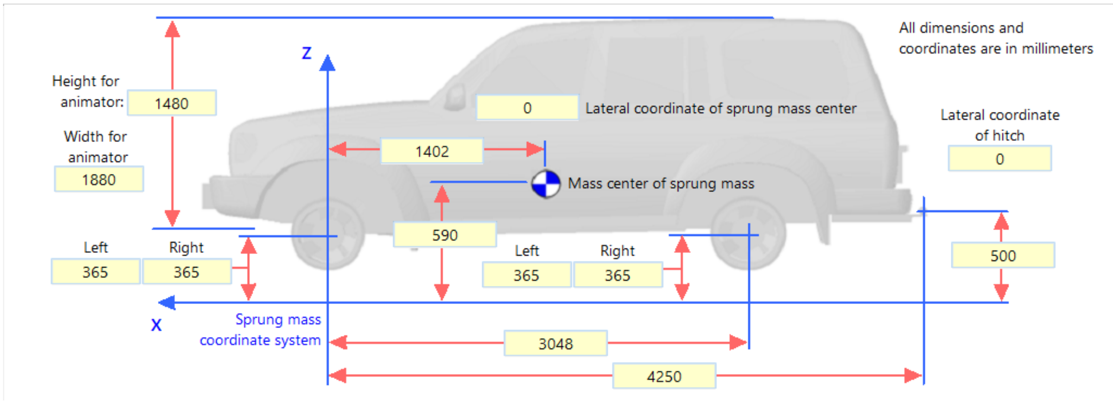
All dimensions and coordinates are in millimeters
Height for animator:
1480
Width for animator
1880
Z
0
Lateral coordinate of sprung mass center
Lateral coordinate of hitch
0
1402
Mass center of sprung mass
590
Left
Right
365
365
Left
Right
365
365
500
X
Sprung mass coordinate system
3048
4250

Figure 6.1: Car sub-model developed in CarSim

Table 6.1: Vehicle parameter values.

Parameters	Value
Sprung Mass Car	$1653kg$
Roll Inertia of Car	$2765kgm^2$
Sprung Mass of Trailer	$466kg$
Roll Inertia of Trailer	$1810kg$

6.2 CarSim Model and Methodology

CarSim is a mechanical simulation tool and is used to simulate automotive systems and analyze vehicle behavior. Based on analysis active controllers can be developed and their performance tested and validated. It is a great tool for active safety implementation and analysis. In this thesis, CarSim is used in conjunction with MATLAB/Simulink to develop an ATS system. CarSim acts as Software-in-Loop (SIL) and provides vehicle information required to tune the controller.

The car-trailer model for the co-simulation is directly developed and tested in CarSim software. Figs. 6.1 and 6.2 show the details of the car and trailer sub-models used in the experiment. All the values are in meters. Table 6.1 lists the car and trailer parameter values.

Fig. 6.3 shows the car-trailer model with the built-in driver model.

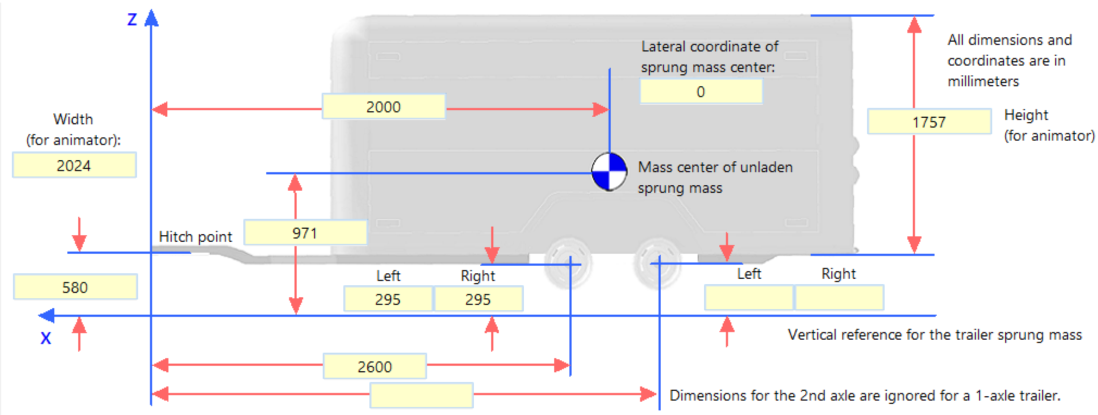

Figure 6.2: Trailer sub-model developed in CarSim

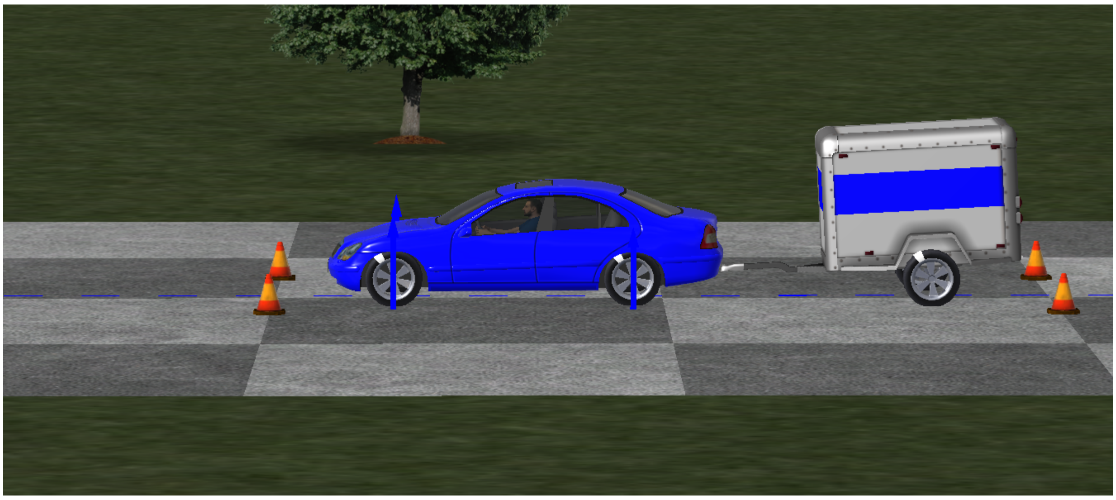

Figure 6.3: Car-trailer model with the built-in driver model.

The co-simulation with CarSim model only requires the K matrix. One important objective to bring this experiment to fruition is to greatly reduce the search space so that the optimizer is able to find an optimal Pareto-front within 1500 generations. In order to achieve this, the K is generated directly with multi-objective evolutionary algorithms.

6.2.1 Built-in Driver Model Offered from CarSim

In all previous tests of the car-trailer combination, open-loop simulations are conducted with a given steering input. A single cycle sine-wave input is used

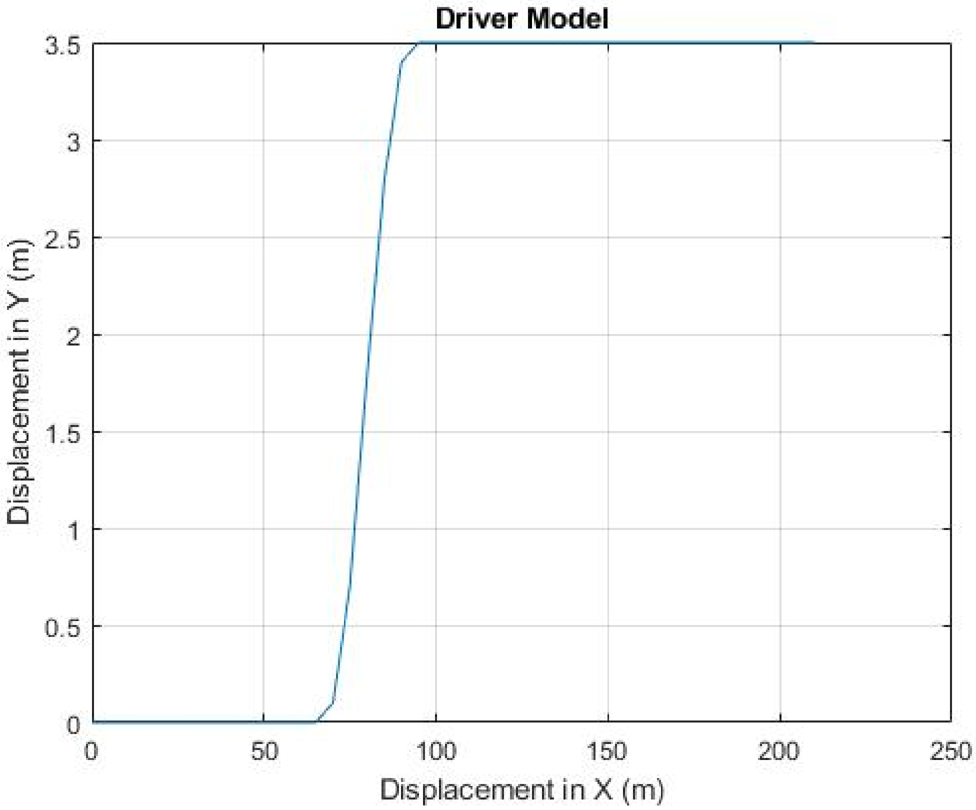

FIGURE 6.4: Predefined trajectory for the closed-loop single lane-change testing maneuver

to simulate a single lance-change maneuver. With a given steering input, depending on the control gain, the trajectory of the car-trailer changes. In order to ensure road safety, a proper maneuver (trajectory) must be ensured. In real life driving, regardless of the speed or driver reaction time, a defined single lane-change requires a car to move a fixed distance on the same road. To simulate a defined single lane-change maneuver, the built-in driver model offered in CarSim software is used. Fig. 6.4 shows the specified trajectory of the closed-loop single lane-change maneuver. Over the testing maneuver, the driver model will adaptively adjust its steering input in order to drive the vehicle to follow the predefined trajectory. Two separate parameters of the driver model are used to generate the look-up table for the GSC.

Two separate parameters of the driver model are used to generate the look-up table for the GSC. The two, driver model dependent, parameters are drivers reaction time and vehicle forward speed.

In order to test the hypothesis, the co-simulation is run and it is found that the design optimization using the MOEA is able to achieve an optimal Pareto-front. The focal point of this section is a look-up table for the gain scheduling controller (GSC) generated using GDE3.

6.2.2 Evaluation Criteria

Evaluation criteria used in this study is RWA and PFOT. Calculation of RWA is very straightforward, the graphs of lateral acceleration of both the car and trailer are acquired in Simulink. The peak lateral acceleration is calculated from these graphs and used to calculate the RWA value.

PFOT is discussed in Chapter 5.3. There are many ways to measure PFOT [23]. The method to measure the PFOT used in this study is the peak overshoot from the ideal trajectory of the car-trailer combination. The CarSim model generates x and y coordinates of trajectory over the simulation period as well as the ideal trajectory. A peak deviation from the trajectory (at any point) is calculated from this graph and used as fitness criteria. To sum it up, the two fitness values generated by the evaluation module are PFOT and RWA.

6.3 Gain Scheduling Controller

A gain scheduling controller (GSC) is used when a non-linear system can be broken down into various linear operating ranges [80]. Control gain values,

K are determined for each of these operating ranges to create a lookup table. Based on environmental factors or internal operating conditions, the GSC chooses a set of values from the look-up table. This allows for a more robust non-linear control using linear techniques and gains scheduling [80]. There are many variations of GSC based on how the parameters are varied [81], this thesis focuses on varying parameters based on vehicle forward speed and driver reaction time to generate a two-dimensional lookup table. Ref. [81] provides a survey into the success of gain scheduling controllers for non-linear control.

There are four steps to design and implement a GSC [81]. The first step is to breakdown the existing system into a number of linearized models, as needed. According to [81], a popular method to do this is the Jacobian linearization. However, the approach used in this thesis does not require mathematical linearization. By varying the speed and reaction time parameters in the CarSim model, the model is automatically updated, which is utilized to generate optimal control gain parameters K for that particular scenario. The second step is to design a linear controller [81], this is also covered by MOEA. The MOEA driven ATS controller is the linear control method. This controller follows design constraints, natural selection, and biological evolution to generate control gain K. The third step is to develop a look-up table and a look-up scheme [81]. This is the actual creation of the GSC. Each gain is scheduled based on vehicle forward speed and reaction time of the driver model. After the GDE3 optimizer terminates, an optimal Pareto-front is achieved. For each combination of vehicle forward speed and driver reaction time, there exists an optimal Pareto-front. To simplify the design, the best tradeoff value of each optimal Pareto-front is used to test the GSC. The last step is to evaluate the systems performance [81]

Table 6.2: Possibilities of GSC Look-up Table

Number	Forward Speed (km/h)	Reaction Time (s)
1	80	0
2	90	0
3	100	0
4	110	0
5	120	0
6	80	0.1
7	90	0.1
8	100	0.1
9	110	0.1
10	120	0.1

and ensure that it is working within design parameters. The GDE3 optimized controller and a gain scheduling controller are compared to confirm correct functionality and to analyze the advantages.

The GSC works based on a two-dimensional look-up table. The first independent parameter is the variation of vehicle forward speed, and the second independent parameter is the reaction time of the driver model. Table 6.2 shows all possible combinations. Each time a control gain value K is chosen for the system, the forward speed of the vehicle and driver reaction time will be the deciding parameters. Or in other words, based on the vehicle forward speed and driver model's reaction time the controller gain will be scheduled.

6.4 Modular Design Methodology

The term co-simulation, in this thesis, addresses the combined usage of CarSim with MATLAB/Simulink. CarSim software offers an integrated S-Function for the Simulink software package. The S-function data are sent and received by CarSim. The simulation results are directly from the CarSim model and the

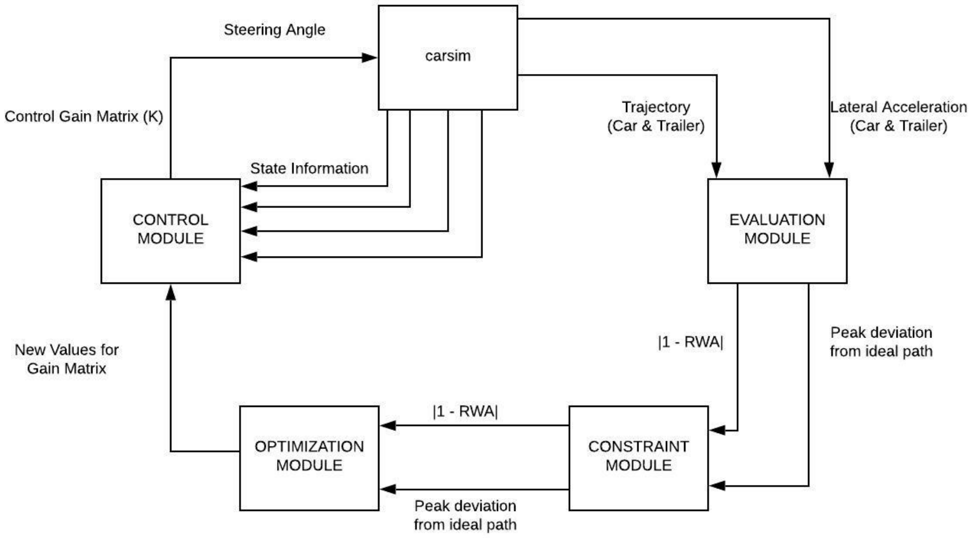

FIGURE 6.5: System model using modular blocks. State Information from top to bottom: Yaw-rate of car, Yaw-rate of trailer, lateral speed of car and lateral speed of trailer.

built-in testing maneuvers. This eliminates the chance of modeling errors and ensures the soundness of the results.

The system model shown in Fig. 6.5 is highly modular. An important part of this research is to ensure a method that is not system-specific. The system model is based on 5 main blocks. The CarSim block holds the car-trailer model, driver-model, and all built-in testing maneuvers. It also provides the simulation results, which are fed into the system. The evaluation block receives the simulation results from CarSim and assigns fitness values based on fitness parameters. If the fitness values exceed or violate the constraints, they are excluded from the simulation. The optimization module receives the fitness costs and uses them to assign fitness to population members for carrying out the evolutionary algorithm and generating new parameters. The control module uses the parameters from the optimization module to generate control gain matrix K which is then fed as the ATS system's steering angle to CarSim. Each of these blocks can be changed. The evaluation method, constraints, optimization techniques, control strategies, and car-trailer model can all be tailored to any system.

The control strategy, in the control and optimization modules, is evolutionary, e.g., GDE3, NSGA-II, etc. This thesis proves this approach with five varying evolutionary strategies, which will be discussed in section 6.6. As it works with five variants, it can be safely said this approach should be successful with other evolutionary algorithms.

The CarSim model changes parameter values every time a new speed or driver model reaction time is introduced. This validates the modularity of the CarSim block. Although not tested, it should work for software-in-loop simulation software, which can provide state information.

The constraints module ensures that, above all, the control gain value must be able to stabilize the system during the complete maneuver. The car-trailer combination must follow the predefined path, without violating road boundaries. The maximum allowed RWA of the car-trailer combination is 2.0. For the GSC, GDE3 is used in the optimization module whereas the MOEA driven ATS controller is hosted by the control module. The GDE3 will run for 1500 generations with a population size of 48.

6.5 Overcoming System Limitation with Innovization

An important part of what makes these experiments successful is the study of the solutions in the Pareto-optimal front. In single objective optimization, there is a lack of data to formulate an opinion about the pattern that the solutions follow. It is a common problem for optimization algorithms that they can get stuck on local optima rather than achieve the global optima. A solution to this problem is to use the results from MOEAs and reform the opinion about the design choices [82]. In this thesis, the parameters evaluated are the search space (SS) and the range of the design variables (DV). Due to the system limitation of just 1000 iterations, innovization is extremely important to reach an optimal Pareto-front with fewer iterations. If the SS is small and the variation of DV is condensed, the algorithm automatically converges faster to a better solution. Innovization is detailed in [82]. Instead of using an automated technique, this task is carried out manually as per the steps listed in [82], for each speed and reaction time pair.

There are two steps for using the innovization technique. First is the initial declaration of SS and range of DV. At the start of the experiment, there is no clear range of either SS or DV. The experiment is run and both parameters are defined based on the maximum and minimum values in the Pareto-optimal front. All algorithms listed in Fig. 6.6, apart from GDE3 Tuned, achieve their optimal Pareto-front following the first step in establishing a baseline SS and DV. For GDE3-Tuned, an additional innovization step was taken. This time, SS, DV, and the rejection range or the constraints are reconsidered. SS and DV are reinitialized based on new information and the constraints are changed from design parameters to the range seen in the optimal Pareto-front shown in Fig. 6.6. By doing this, a better optimal Pareto-front is achieved.

6.6 Algorithm Comparison

For the MOEA driven ATS controller, an algorithm must be decided. GDE3 is a great algorithm for ATS controller [72], but it is not compared against other MOEAs.

The five state-of-the-art MOEAs [83] are GDE3, Multi-objective Particle Swarm Optimization (MOPSO) [84], Non-dominated Sorting Genetic Algorithm II [9] and III [85] (NSGA II & III) and Strength Pareto Evolutionary Algorithm II (SPEA2) [86]. The details of these algorithms are discussed in the referenced articles. SPEA2, NSGA-II, and NSGA-III and GDE3 are rather similar, with the main difference being their method to generate the optimal Pareto-front. MOPSO, on the other hand, uses a completely different evolutionary technique of Particle Swarm Optimization (PSO). The last algorithm is innovized GDE3. By studying the optimal Pareto-front, the extreme points, and the knee-points, the algorithm can be improved or innovized to provide even better results [82]. This is extremely important in this case, as only 1500 generations could be used with a population size of 48.

After running the GDE3 algorithm and generating optimal Pareto-front at $120 km/h$, each point is carefully studied. Each point in the optimal Pareto-front corresponds to a control gain matrix, which has four terms in this case. By studying the four terms and their maximum and minimum values, the search space is reduced, and the GDE3 algorithm is run again. This is termed as tuned GDE3. This test is extremely expensive and required a long time, and was not conducted for all other algorithms nor all speeds. This is only displayed on the comparison curve in Fig. 6.7.

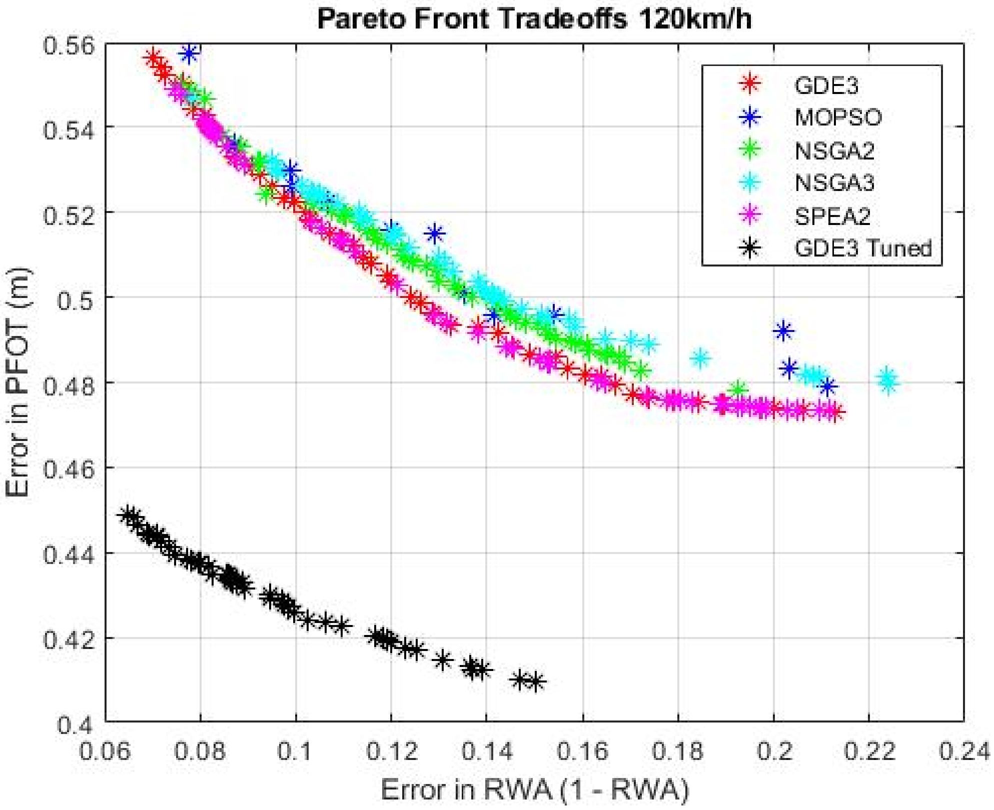

FIGURE 6.6: Algorithm comparison based on two design PFOT.

From Fig. 6.6 the points close to the origin are better as these are optimized points.

The best algorithm is the tuned GDE3. The other two algorithms competing for the first spot are GDE3 and SPEA2. GDE3 has more points and a better diversity than SPEA2. The best point of individual objectives generated by GDE3 is superior to SPEA2. These results along with the results from Chapter 5 prove that GDE3 is the better algorithm. The $x - axis$ of Fig. 6.6 illustrates the error from the ideal RWA of 1.0 and the $y - axis$ is the maximum deviation from the ideal value in meters. Due to the expensive nature of testing and time limitation, this same testing cannot be carried out on other speeds. The highest speed is chosen as it presents the greatest difficulty. Figs. 6.7 and 6.8 show the trajectory the vehicle followed in the driving simulation. These trajectories are

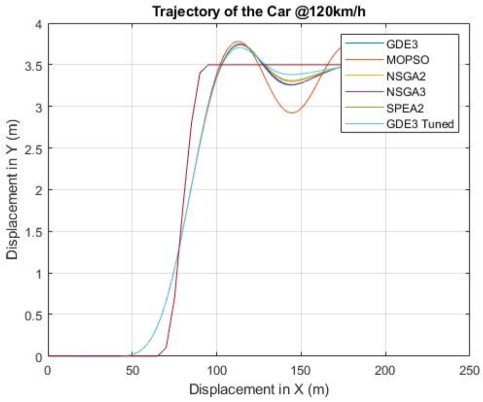

FIGURE 6.7: Trajectory of Car at $120km/h$ for Algorithm Comparison

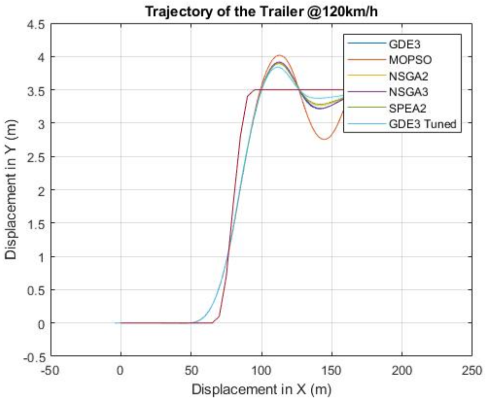

FIGURE 6.8: Trajectory of Trailer at $120km/h$ for Algorithm Comparison

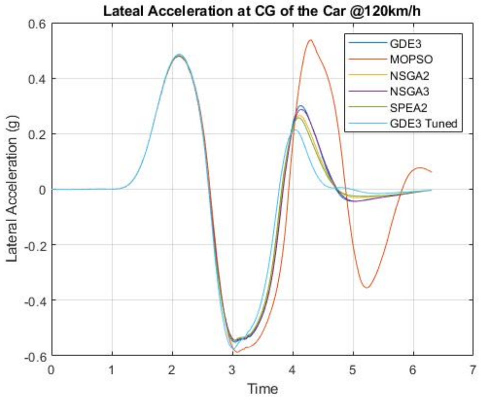

FIGURE 6.9: Lateral Acceleration of Car at $120km/h$ for Algorithm Comparison

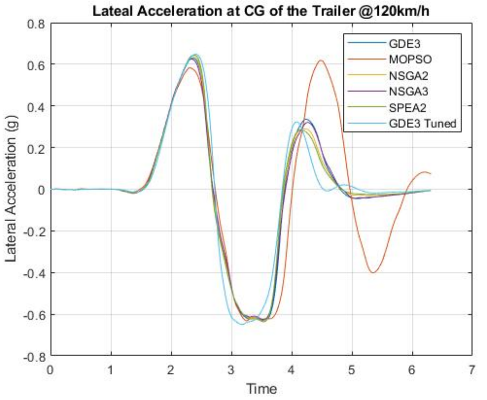

FIGURE 6.10: Lateral Acceleration of Trailer at $120km/h$ for Algorithm Comparison

calculated by using the trade-off point, the central point of the optimal Pareto-front, for each of the algorithms. Figs. 6.9 and 6.10 show the lateral acceleration of car and trailer respectively for the same simulation.

6.7 Results: GDE3 Generated GSC vs Passive

The optimal Pareto-front, for each speed and reaction time, has multiple values. With a population size of 48, there can be up to 48 distinct possibilities of control gain matrix K per schedule. It is not wasted labor to compare all points

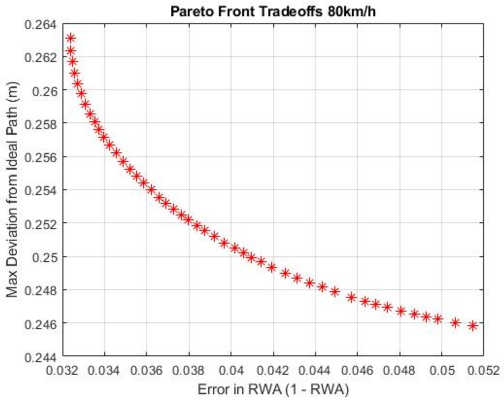

FIGURE 6.11: Optimal Pareto-front for 80 km/h and 0(s) driver model reaction time

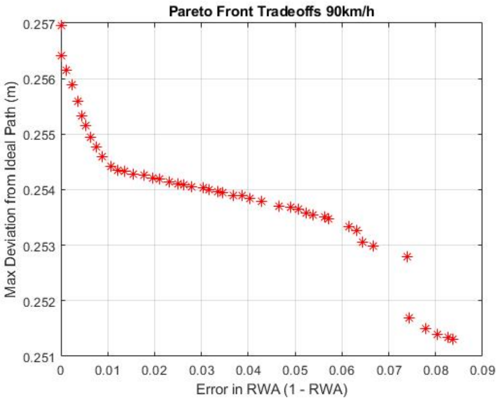

FIGURE 6.12: Optimal Pareto-front for 90 km/h and 0(s) driver model reaction time

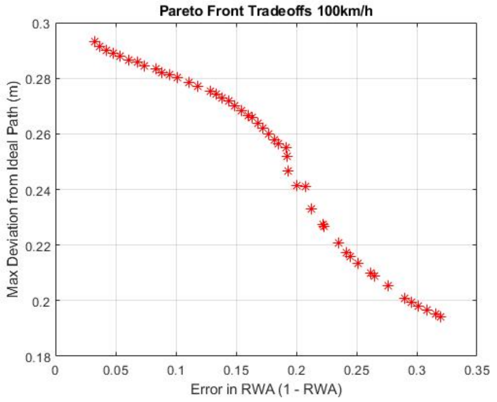

FIGURE 6.13: Optimal Pareto-front for 100 km/h and 0(s) driver model reaction time

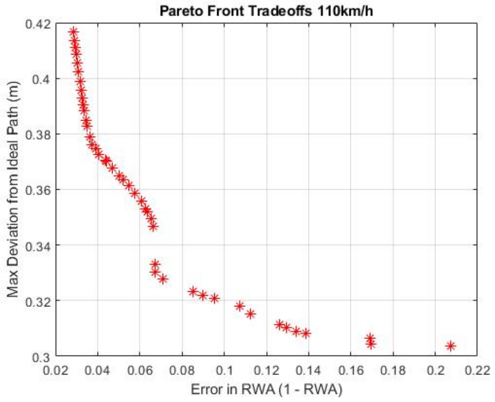

FIGURE 6.14: Optimal Pareto-front for 110 km/h and 0(s) driver model reaction time

with all other points. There are three main points of interest in every optimal Pareto-front: the two utopia points (extreme points) and the optimal tradeoff or the best compromise solution. The optimal tradeoff is a decision to be made by the system designer, but in this case, the point, which has the best compromise between the two objectives, is considered.

Figs. 6.11 to 6.20 show the optimal Pareto-fronts, generated by GDE3, after co-simulation using the system model and constraints specified in section 6.3. The optimal Pareto-fronts for all possible studied cases, according to Table 6.2, are listed. From each of these optimal Pareto-fronts, for each schedule, two utopia

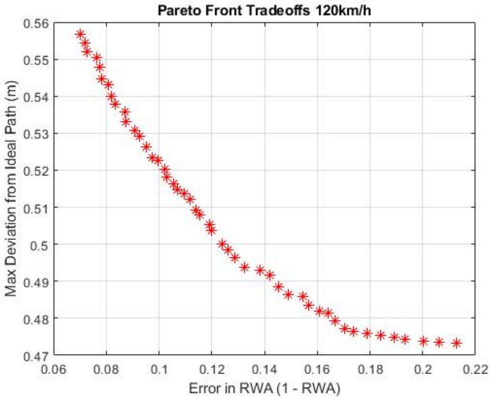

FIGURE 6.15: Optimal Pareto-front for 120 km/h and 0(s) driver model reaction time

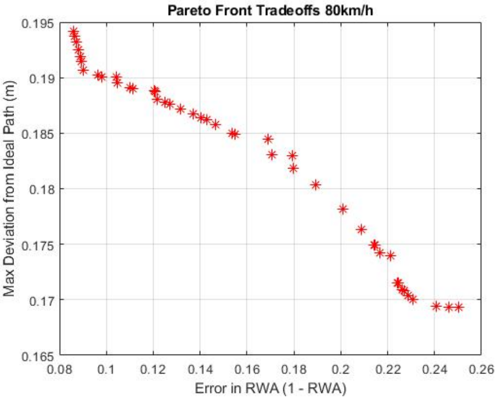

FIGURE 6.16: Optimal Pareto-front for 80 km/h and 0.1(s) driver model reaction time

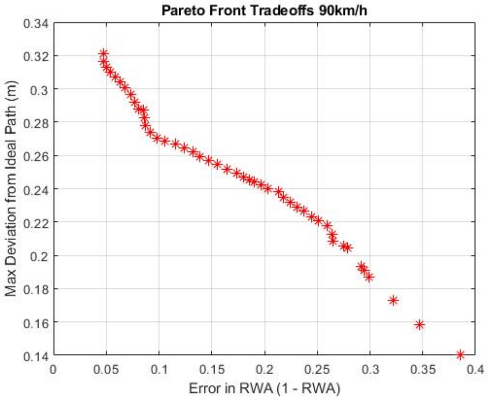

FIGURE 6.17: Optimal Pareto-front for 90 km/h and 0.1(s) driver model reaction time

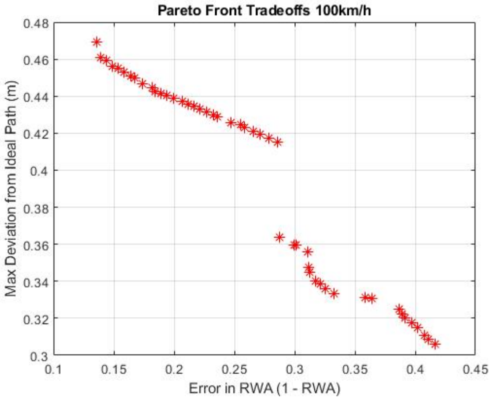

FIGURE 6.18: Optimal Pareto-front for 100 km/h and 0.1(s) driver model reaction time

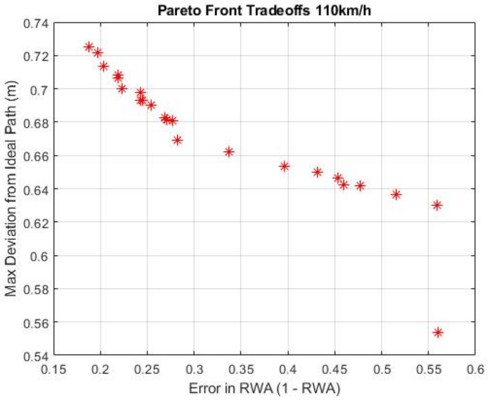

FIGURE 6.19: Optimal Pareto-front for 110 km/h and 0.1(s) driver model reaction time

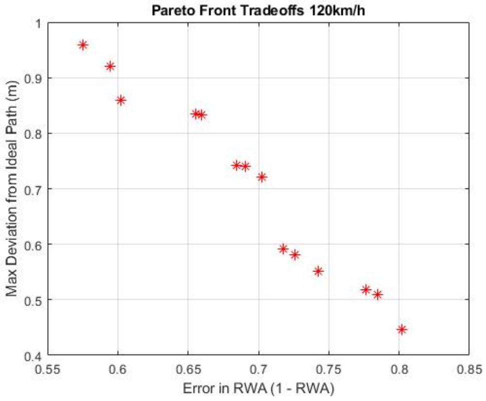

FIGURE 6.20: Optimal Pareto-front for 120 km/h and 0.1(s) driver model reaction time

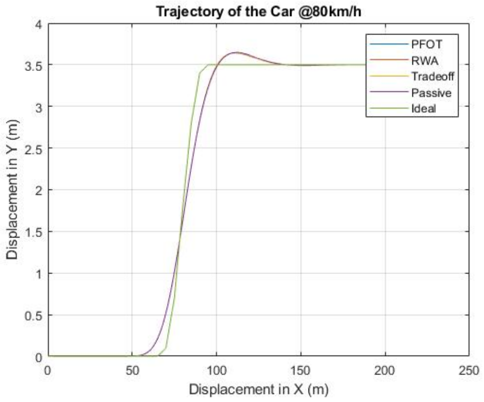

Figure 6.21: Trajectory of the car at 80 km/h and 0(s) driver model reaction time.

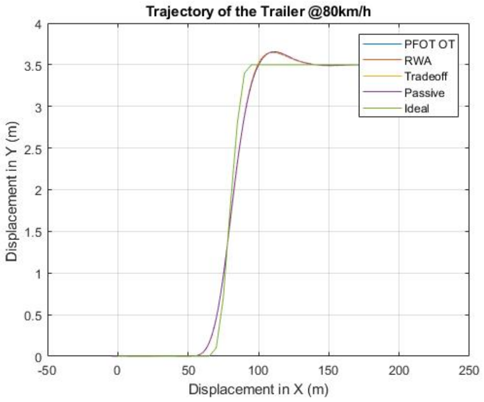

Figure 6.22: Trajectory of the trailer at 80 km/h and 0(s) driver model reaction time.

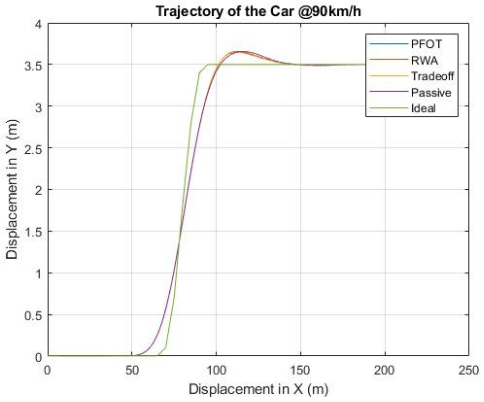

Figure 6.23: Trajectory of the car at 90 km/h and 0(s) driver model reaction time.

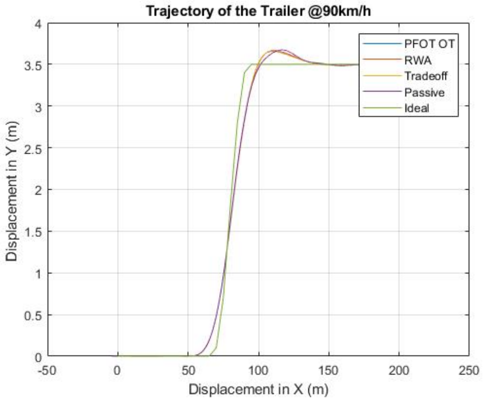

Figure 6.24: Trajectory of the trailer at 90 km/h and 0(s) driver model reaction time.

points and one optimal compromise are chosen, and the trajectories of car and trailer are compared with the passive system.

Figs. 6.21 to 6.40 show the trajectories of car and trailer for all ten possible schedules. As the speed increases, the value of the optimization algorithm and the MOEA driven ATS controller are more apparent. Above $100 km/h$, the passive system completely fails to complete the maneuver when the driver model reaction time is 0 seconds, and fails at $90 km/h$ when the driver model reaction time is set to 0.1 seconds. The GDE3 optimized ATS controller is able to stabilize the system and to complete the single lane change maneuver. From Fig

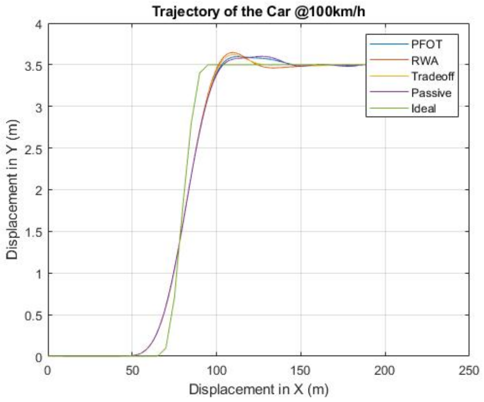

FIGURE 6.25: Trajectory of the car at 100 km/h and 0(s) driver model reaction time.

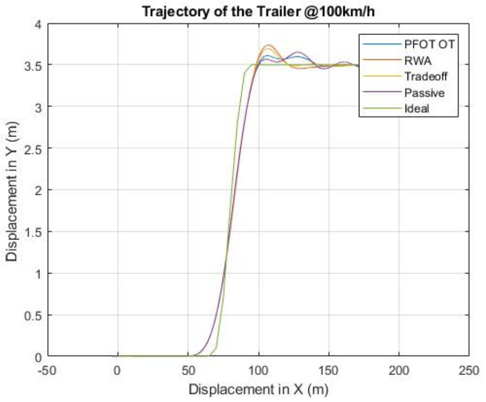

FIGURE 6.26: Trajectory of the trailer at 100 km/h and 0(s) driver model reaction time.

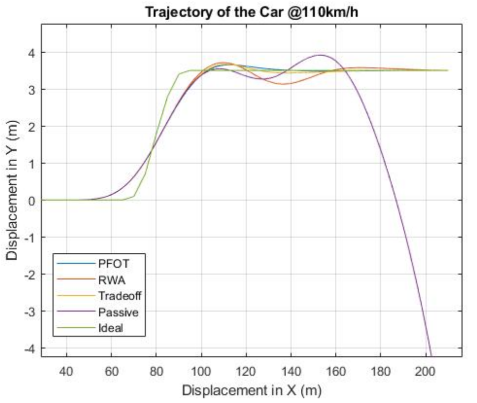

FIGURE 6.27: Trajectory of the car at 110 km/h and 0(s) driver model reaction time.

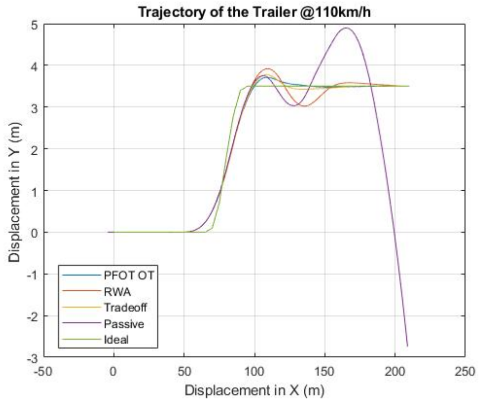

FIGURE 6.28: Trajectory of the trailer at 110 km/h and 0(s) driver model reaction time.

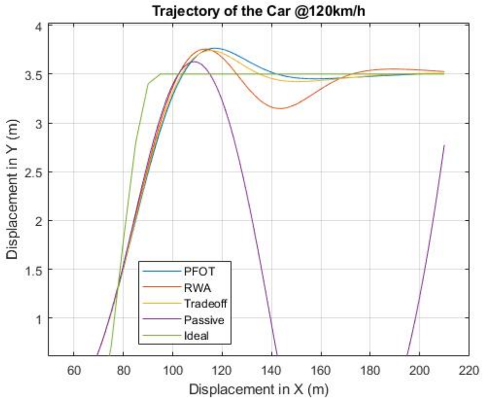

FIGURE 6.29: Trajectory of the car at 120 km/h and 0(s) driver model reaction time.

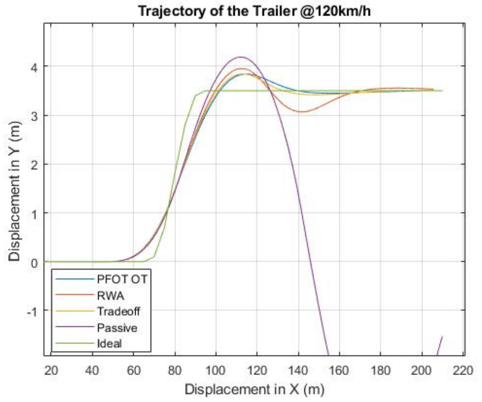

FIGURE 6.30: Trajectory of the trailer at 120 km/h and 0(s) driver model reaction time.

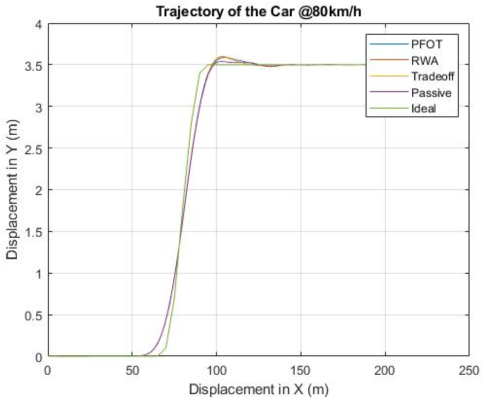

FIGURE 6.31: Trajectory of the car at 80 km/h and 0.1(s) driver model reaction time.

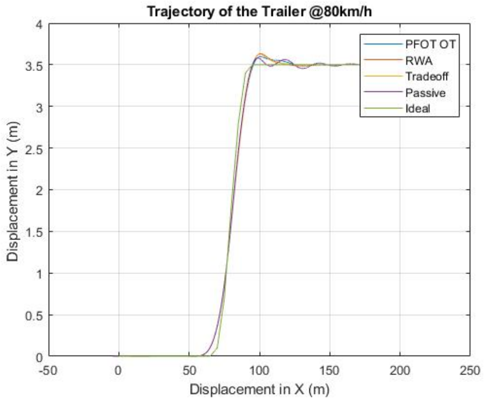

FIGURE 6.32: Trajectory of the trailer at 80 km/h and 0.1(s) driver model reaction time.

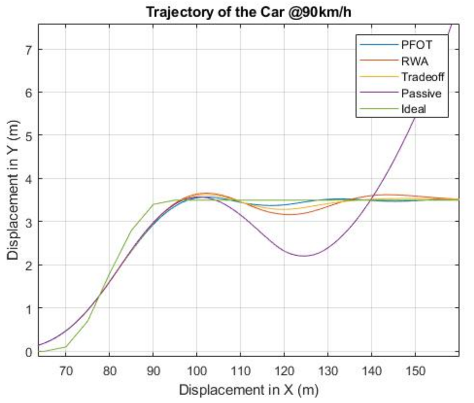

FIGURE 6.33: Trajectory of the car at 90 km/h and 0.1(s) driver model reaction time.

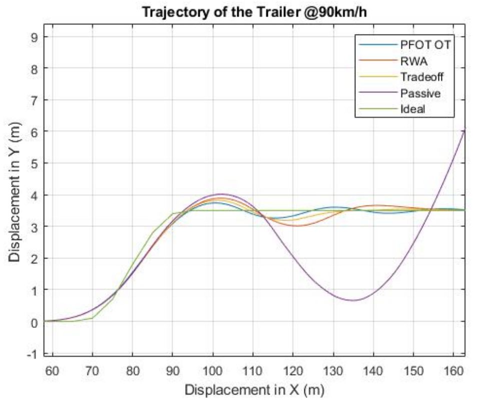

FIGURE 6.34: Trajectory of the trailer at 90 km/h and 0.1(s) driver model reaction time.

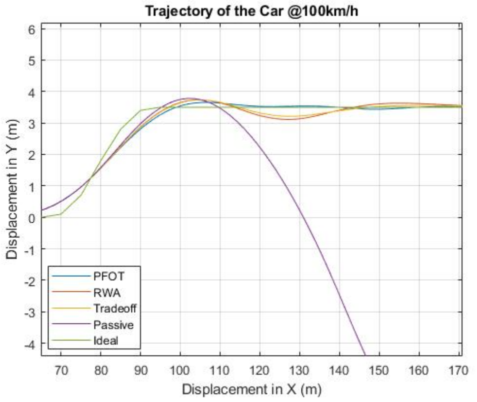

FIGURE 6.35: Trajectory of the car at 100 km/h and 0.1(s) driver model reaction time.

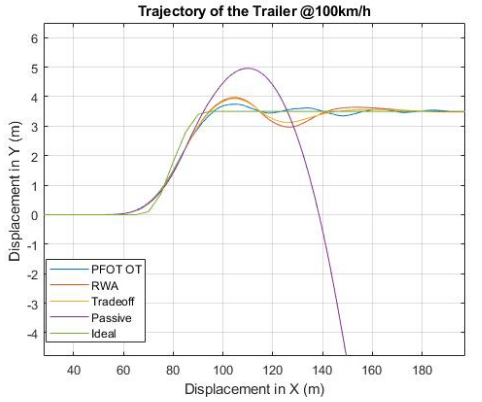

FIGURE 6.36: Trajectory of the trailer at 100 km/h and 0.1(s) driver model reaction time.

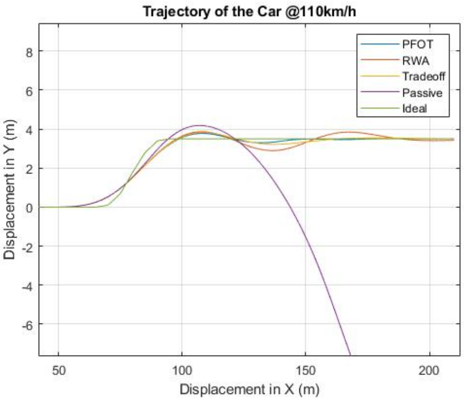

FIGURE 6.37: Trajectory of the car at 110 km/h and 0.1(s) driver model reaction time.

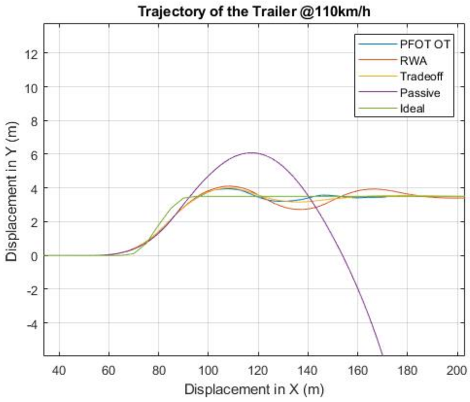

FIGURE 6.38: Trajectory of the trailer at 110 km/h and 0.1(s) driver model reaction time.

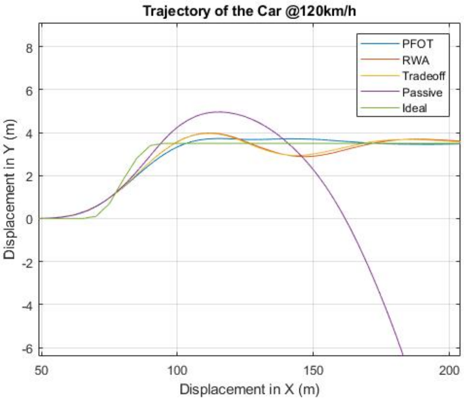

FIGURE 6.39: Trajectory of the car at 120 km/h and 0.1(s) driver model reaction time.

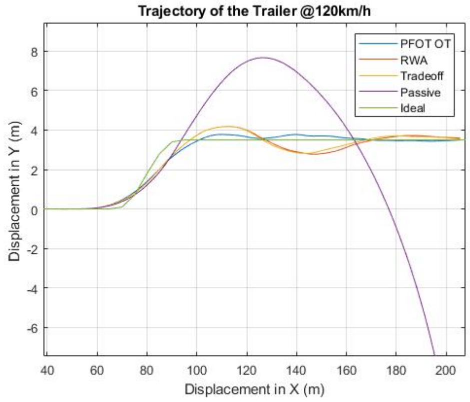

FIGURE 6.40: Trajectory of the trailer at 120 km/h and 0.1(s) driver model reaction time.

6.21, it can be observed that even though the maneuver is completed the RWA value is around 0.45 for the RWA optimized point. From the progression of the optimal Pareto-front for higher speeds and higher reaction times, the system limitation plays an integral part. With only 1500 generations, it cannot be said for certain that the best point has been found, for the hardest settings. The results show that the GDE3 optimized MOEA ATS controller is indeed able to stabilize the system to a great extent.

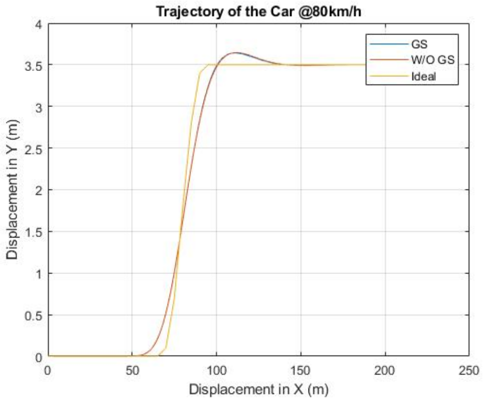

FIGURE 6.41: GSC Trajectory of the car at 80 km/h and 0.0(s) driver model reaction time.

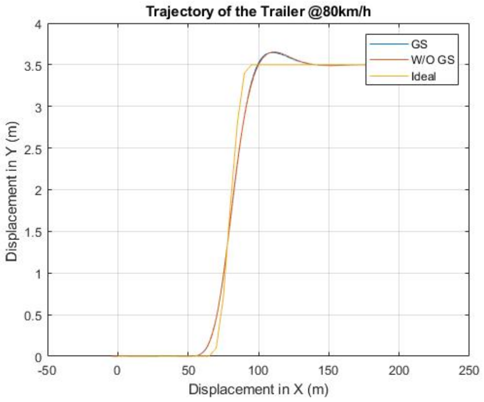

FIGURE 6.42: GSC Trajectory of the trailer at 80 km/h and 0.0(s) driver model reaction time.

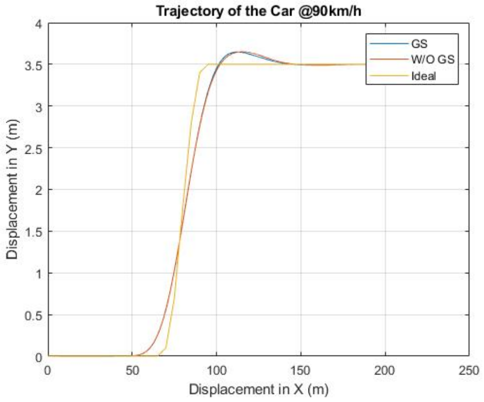

FIGURE 6.43: GSC Trajectory of the car at 90 km/h and 0.0(s) driver model reaction time.

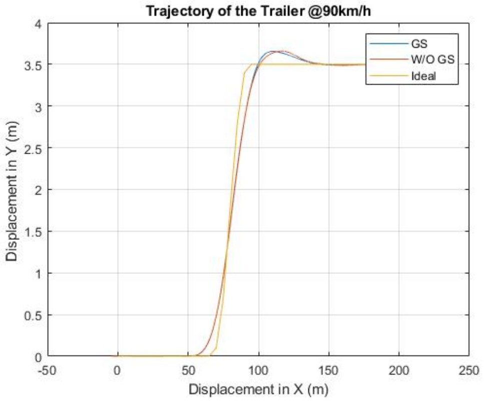

FIGURE 6.44: GSC Trajectory of the trailer at 90 km/h and 0.0(s) driver model reaction time.

6.8 Results: Gain Scheduling

The last part of this chapter is to demonstrate, using the aforementioned results, how the gain scheduling controller (GSC) further improves the ATS system. GSC controller is compared to a GDE3 optimized ATS tuned at $100km/h$ at 0 second of the driver model reaction time. The trajectories of car and trailer are compared. The control gains K is chosen such that the trajectory is optimized.

From Figs. 6.42 to 6.58 it is apparent that using a GSC has a great advantage

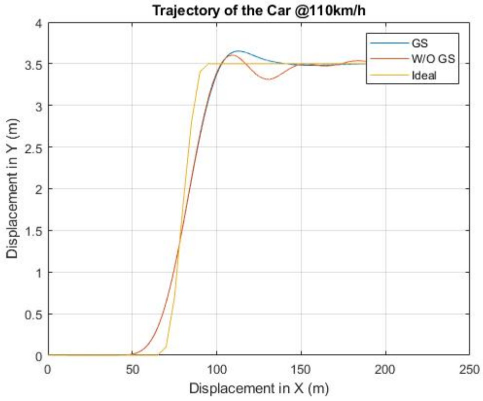

FIGURE 6.45: GSC Trajectory of the car at 110 km/h and 0.0(s) driver model reaction time.

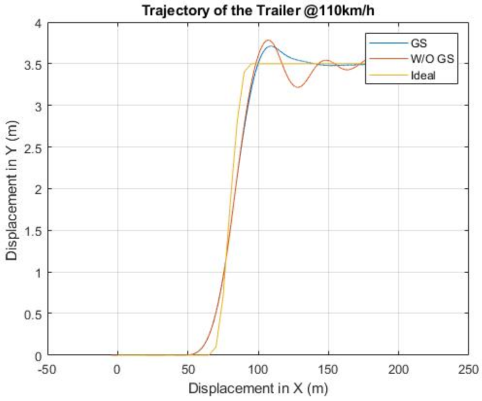

FIGURE 6.46: GSC Trajectory of the trailer at 110 km/h and 0.0(s) driver model reaction time.

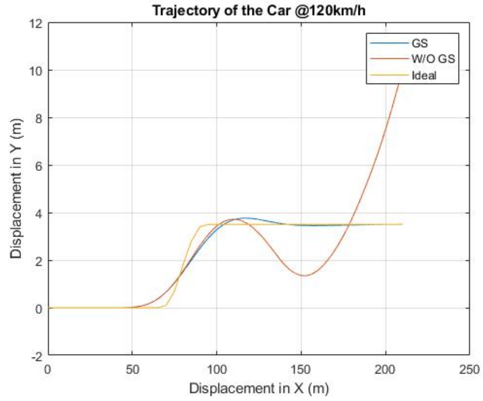

FIGURE 6.47: GSC Trajectory of the car at 120 km/h and 0.0(s) driver model reaction time.

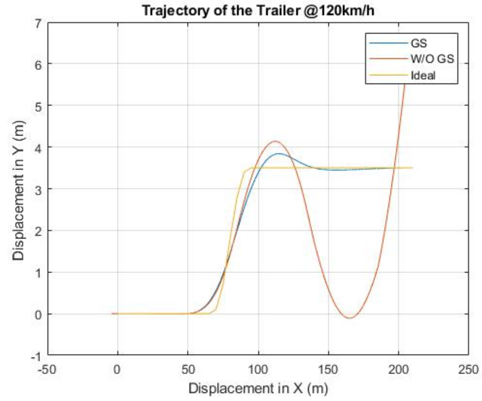

FIGURE 6.48: GSC Trajectory of the trailer at 120 km/h and 0.0(s) driver model reaction time.

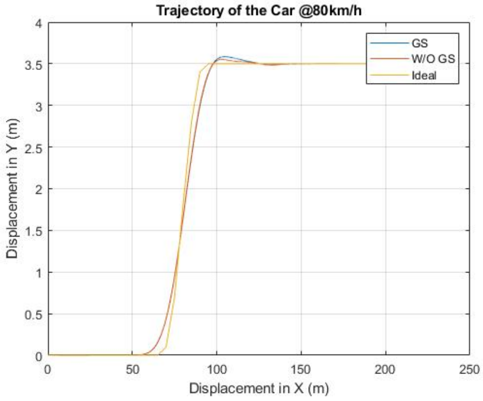

FIGURE 6.49: GSC Trajectory of the car at 80 km/h and 0.1(s) driver model reaction time.

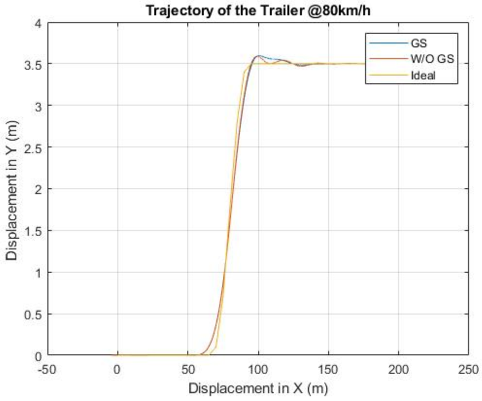

FIGURE 6.50: GSC Trajectory of the trailer at 80 km/h and 0.1(s) driver model reaction time.

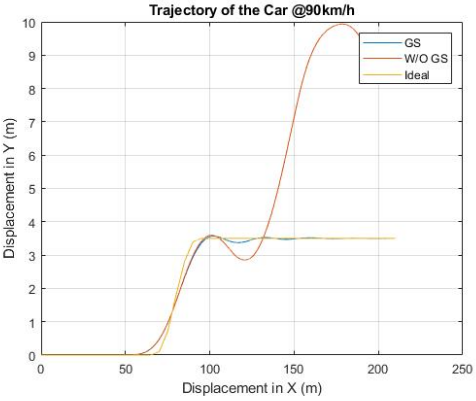

Figure 6.51: GSC Trajectory of the car at 90 km/h and 0.1(s) driver model reaction time.

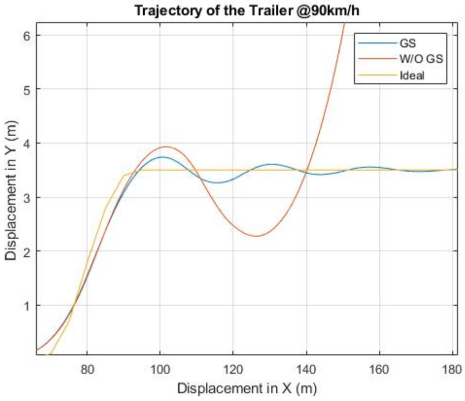

Figure 6.52: GSC Trajectory of the trailer at 90 km/h and 0.1(s) driver model reaction time.

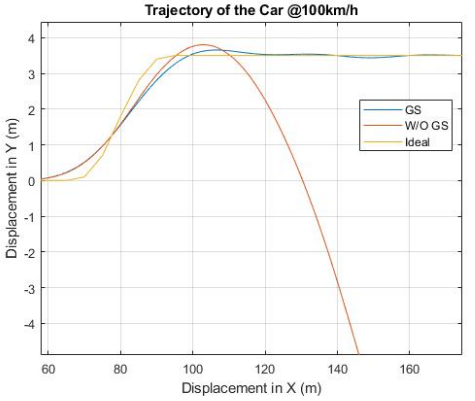

Figure 6.53: GSC Trajectory of the car at 100 km/h and 0.1(s) driver model reaction time.

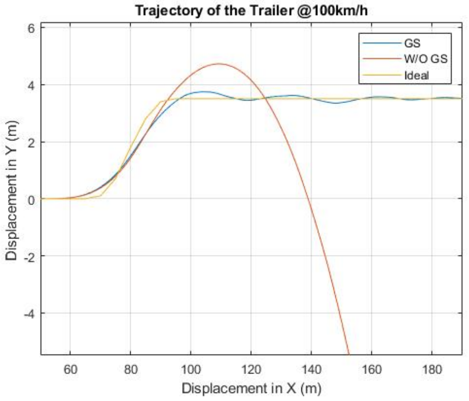

Figure 6.54: GSC Trajectory of the trailer at 100 km/h and 0.1(s) driver model reaction time.

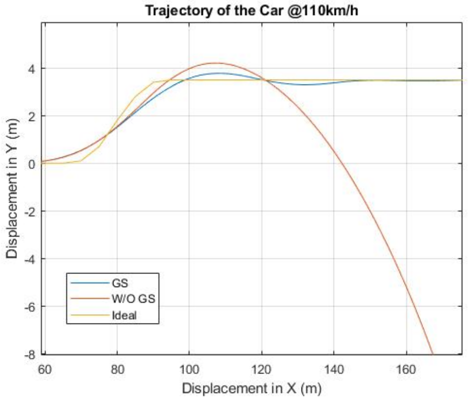

Figure 6.55: GSC Trajectory of the car at 110 km/h and 0.1(s) driver model reaction time.

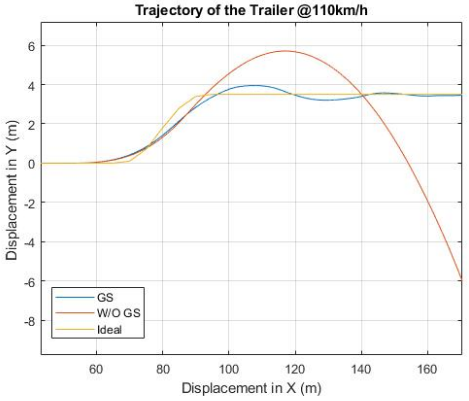

Figure 6.56: GSC Trajectory of the trailer at 110 km/h and 0.1(s) driver model reaction time.

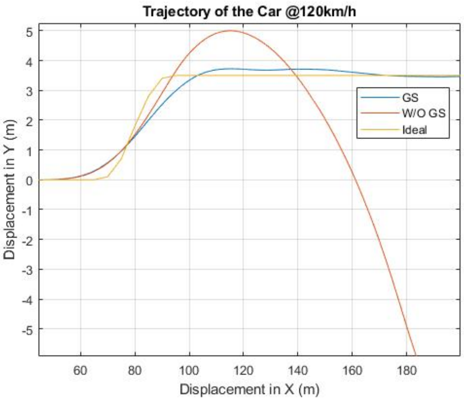

FIGURE 6.57: GSC Trajectory of the car at 120 km/h and 0.1(s) driver model reaction time.

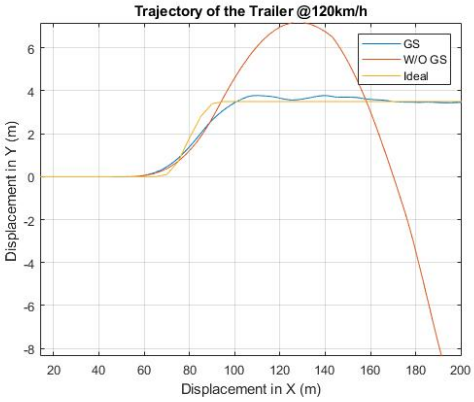

FIGURE 6.58: GSC Trajectory of the trailer at 120 km/h and 0.1(s) driver model reaction time.

TABLE 6.3: RWA Comparison GS vs W/O GS

Speed (km/h)	Reaction Time (s)	GS	W/O GS	Improvement (%)
80	0.0	1.0751	1.1749	8.49
90	0.0	1.0929	1.2662	13.69
110	0.0	1.2072	1.3120	7.99
120	0.0	1.3626	3.3314	59.03
80	0.1	1.2503	1.2770	2.09
90	0.1	1.3853	1.9563	29.19
100	0.1	1.4168	3.2507	56.42
110	0.1	1.5600	3.2304	51.71
120	0.1	1.8018	3.3320	45.92

over using a regular controller. For lower speeds, the improvement is not as apparent by looking at just the trajectory, but looking at Table 6.3 the improvement is more apparent. From this comprehensive study into gain scheduling and co-simulation, it is demonstrated that GDE3 optimized GSC greatly improves the performance of ATS for the non-linear car-trailer combination with varying speeds and driver reaction time. This study includes both forward speed variation and driver model reaction time variation. The simulation can be further tuned towards creating a GSC that caters to the real driver as well as the vehicle forward speed.

The GSC has a two-dimensional discrete scheme so an important note is when to switch between different modes of operation. In this thesis, two switching methods are tested. Firstly, controller gains are changed when the speed increases beyond the next discrete step, e.g. controller gains shifts from those associated with 80km/h to 90km/h if the speed goes to or beyond 90km/h. The second scheme tested is changing gains by rounding to the nearest value, e.g. mode shifts from 80km/h to 90km/h when the speed crosses 85km/h. The performance in both cases is seen to be similar but when using the first scheme the overall number of times the gains change is less than the second.

6.9 Important Observation

The modular method proposed in this chapter utilizes a control strategy not used before. Various MOEAs are used to generate control gain matrix K. It is found that the proposed control strategy operates for all five algorithms. Similarly, the modular approach works with 10 different combinations of vehicle forward speed and driver model reaction time, as shown in Table 6.3. During the process of innovization, the evaluation and constraint module is constantly changed. When a different MOEA is used, the optimization and control module also change. This is a comprehensive proof that this strategy is modular. It conducted co-simulation is based on CarSim software. CarSim provides various information to the system, e.g. lateral accelerations, trajectories, etc. If another software-in-loop (SIL) system or a hardware-in-loop (HIL) mechanical simulator can provide similar information, this approach should work for that particular tool too.

The modular method in section 6.4 utilizes a control strategy not used before.

Various MOEA are used to generate control gain matrix K. The big observation here is that it works, not just with one algorithm but 5. Similarly the modular approach works with 10 different CarSim models, anytime speed or driver model reaction time is changed the model changes. During the process of innovization the evaluation and constraint module is constantly changed. When a different MOEA is used the optimization and control module also change. This is a comprehensive proof that this strategy is modular at least for CarSim. CarSim provides various information back to the system, e.g. lateral accelerations, trajectories, etc. If another software-in-loop (SIL) system or a hardware-in-loop (HIL) mechanical simulator can provide the same information, this approach should work for that particular tool too.

Chapter 7

Conclusions

This chapter concludes the thesis and highlights the insightful findings achieved in the research. The motivation behind this thesis is to improve the lateral stability of car-trailer combinations by designing robust ATS controllers using multi-objective optimization. The knowledge retrieved from the results derived from Chapters 3 and 4, as well as the detailed literature review conducted in Chapter 2 is applied to the design optimization of the robust ATS controllers implemented in Chapters 5 and 6.

7.1 Contributions

By means of a case study, an effective differential evolution mutation strategy is tested and selected through a comprehensive performance and speed analysis. This strategy is used in the GDE3 variant for the design optimization of the robust ATS controllers. Results show that this variant outperforms its genetic algorithm counterpart, NSGA-II, and conventional tuning methods. Using GDE3, as the MOEA, an optimized ATS controller is obtained for car-trailer combinations. The system designer is able to choose a result from an optimal Pareto-front set of solutions.

A MOEA tuned gain scheduling controller (GSC) is designed for car-trailer combinations. The GSC is designed using the LQR control technique considering the variation of vehicle forward speed. A set of control gain matrices are used as the lookup tables for the GSC. The GSC outperforms non-GSCs in terms of improving the lateral stability and the PFOT performance of car-trailer combinations.

Considering the interactions of driver-vehicle/controller-road, a new GSC is designed using closed-loop co-simulation and multi-objective optimization method. Two independent parameters are selected to build the two dimensional lookup table for the gain scheduling scheme of the GSC. The two parameters are: 1) driver model reaction time, and 2) vehicle forward speed. The co-simulation is implemented by combining the controller designed in Simulink/Matlab with the virtual car-trailer developed in CarSim. Built upon the co-simulation, a multi-objective optimization method is applied to design the GSC. Numerical simulation demonstrates the robustness and effectiveness of the GSC.

7.2 Recommendations for Future Studies

Based on the investigations conducted in the research, the following recommendations are offered for future studies:

- The GSCs designed in this research are mainly focused on improving the lateral stability of car-trailer combinations under high-speed evasive maneuvers. In the future studies, attention may be paid to extend the functionality of the GSC for improving the low-speed PFOT in curved path negotiations.

- The GDE3 Tuned algorithm performs well thanks to innovization. A further study can be carried out on. Innovization is proven here to take the existing algorithm to a greater place. By tackling both system limitations and performing innovization, a more robust gain scheduling controller may be created.

- The maneuvers simulated in this thesis are closed-loop single lane-change maneuvers at a constant vehicle forward speed. A new testing maneuver may be designed, which incorporates variable vehicle forward speeds. This maneuver, with a greater speed sensitivity, may be used to further improve and test the GSC. The research into the maneuver, which may accurately depict and test a GSC, is a great step towards creating a robust control system.

Bibliography

[1] Peter J Fleming and Robin C Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control engineering practice*, 10(11): 1223–1241, 2002.

[2] Aidan O'Dwyer. *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press, 2009. ISBN 9781848162426. URL https://books.google.ca/books?id=zivrLUbIMgUC.

[3] Paul Fancher and Chris Winkler. Directional performance issues in evaluation and design of articulated heavy vehicles. *Vehicle System Dynamics*, 45, 07 2007.

[4] Der-Ho Wu. A theoretical study of the yaw/roll motions of a multiple steering articulated vehicle. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 215(12):1257–1265, 2001.

[5] Rafay Shamim, Md Manjurul Islam, and Yuping He. A comparative study of active control strategies for improving lateral stability of car-trailer systems. *SAE Technical Papers*, 04 2011.

[6] Gwen Moritz. The amazon effect. *Arkansas Business*, 34(26):18, Jun 2017. URL http://search.proquest.com.uproxy.library.dc-uoit.ca/docview/1915756926?accountid=14694. Name - Amazon.com Inc; Walmart Inc; Apple Inc; Copyright - Copyright Arkansas Business Jun 26-Jul 2, 2017; Last updated - 2018-11-01.

[7] CTV Windsor. Transport trucks involved in 1 in 5 crashes on ontario roads, 2017. URL https://windsor.ctvnews.ca/transport-trucks-involved-in-1-in-5-crashes-on-ontario-roads-1.3462207. [Online; accessed 2-January-2019].

[8] Hongliang Wang, Haobin Dong, Lianghua He, Yongle Shi, and Yuan Zhang. Design and simulation of LQR controller with the linear inverted pendulum. *Electrical and Control Engineering (ICECE), 2010 International Conference on*, pages 699–702, 2010.

[9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[10] Manjurul Md Islam, Steve Mikaric, Yuping He, and Thomas Hu. Parallel design optimization of articulated heavy vehicles with active safety systems. In *Proceedings of the FISITA 2012 world automotive congress*, pages 1563–1575. Springer, 2013.

[11] Yuping He, Md Manjurul Islam, and Timothy D Webster. An integrated design method for articulated heavy vehicles with active trailer steering systems. *SAE International Journal of Passenger Cars-Mechanical Systems*, 3 (2010-01-0092):158–174, 2010.

[12] Robert Dimeo and Kwang Y Lee. Boiler-turbine control system design using a genetic algorithm. *IEEE transactions on energy conversion*, 10(4): 752–759, 1995.

[13] Nabil Nassif, Stanislaw Kajl, and Robert Sabourin. Optimization of hvac control system strategy using two-objective genetic algorithm. *HVAC&R Research*, 11(3):459–486, 2005.

[14] Jonathan A Wright, Heather A Loosemore, and Raziyeh Farmani. Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and buildings*, 34(9):959–972, 2002.

[15] Kailash Krishnaswamy, George Papageorgiou, Sonja Glavaski, and Antonis Papachristodoulou. Analysis of aircraft pitch axis stability augmentation system using sum of squares optimization. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1497–1502. IEEE, 2005.

[16] Mahesh P Nagarkar and Gahininath Vikhe. Optimization of the linear quadratic regulator (LQR) control quarter car suspension system using genetic algorithm. *Ingeniería e Investigación*, 36(1):23–30, 2016.

[17] Amir Ghoreishi and Mohammad Nekoui. Optimal weighting matrices design for LQR controller based on genetic algorithm and pso. *Advanced Materials Research*, 433-440:7546–7553, 01 2012.

[18] Yuping He and Md Manjurul Islam. An automated design method for active trailer steering systems of articulated heavy vehicles. *Journal of Mechanical Design*, 134(4):041002, 2012.

[19] Eungkil Lee, Saurabh Kapoor, Tushita Sikder, and Yuping He. An optimal robust controller for active trailer differential braking systems of car-trailer combinations. *International Journal of Vehicle Systems Modelling and Testing*, 12(1-2):72–93, 2017.

[20] Rudolf Emil Kalman. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana*, 5(2):102–119, 1960.

[21] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[22] Amin Mohammadbagheri, Narges Zaeri, and Mahdi Yaghoobi. Comparison performance between PID and LQR controllers for 4-leg voltage-source inverters. In *International Conference Circuit, System and Simulation*, 2011.

[23] Md. Manjurul Islam. Design synthesis of articulated heavy vehicles with active trailer steering systems. Master's thesis, University of Ontario Institute of Technology, Oshawa, Ontario, Canada, 2010.

[24] Van Tan Vu, Olivier Sename, Luc Dugard, and Pter Gspr. Active anti-roll bar control using electronic servo valve hydraulic damper on single unit heavy vehicle. *IFAC-PapersOnLine*, 49:418–425, 12 2016. doi: 10.1016/j.ifacol.2016.08.062.

[25] Yuping He and Md Manjurul Islam. An automated design method for active trailer steering systems of articulated heavy vehicles. *Journal of Mechanical Design*, 134(4):041002, 2012.

[26] Kyong-il Kim, Hsin Guan, Bo Wang, Rui Guo, and Fan Liang. Active steering control strategy for articulated vehicles. *Frontiers of Information Technology & Electronic Engineering*, 17(6):576–586, Jun 2016.

[27] Guido Koch and Tobias Kloiber. Driving state adaptive control of an active vehicle suspension system. *IEEE transactions on Control Systems technology*, 22(1):44–57, 2013.

[28] Xuejun Ding, Yuping He, Jing Ren, and Tao Sun. A comparative study of control algorithms for active trailer steering systems of articulated heavy vehicles. In *2012 American Control Conference (ACC)*, pages 3617–3622. IEEE, 2012.

[29] Arjun C Unni, Anjali Junghare, Vivek Mohan, and Weerakorn Ongsakul. PID, fuzzy and LQR controllers for magnetic levitation system. In *2016 International Conference on Cogeneration, Small Power Plants and District Energy (ICUE)*, pages 1–5. IEEE, 2016.

[30] Mundher H.A. Yaseen. A comparative study of stabilizing control of a planer electromagnetic levitation using PID and LQR controllers. *Results in Physics*, 7:4379 – 4387, 2017. ISSN 2211-3797.

[31] Herman A Reise. Automatic trailer sway sensing and brake applying system, August 9 1977. US Patent 4,040,507.

[32] Mutaz Keldani and Yuping He. Design of electronic stability control (esc) systems for car-trailer combinations. EasyChair Preprint no. 313, EasyChair, 2018. doi: 10.29007/hrw2.

[33] Ning Zhang, Guo-dong Yin, Tian Mi, Xiao-gao Li, and Nan Chen. Analysis of dynamic stability of car-trailer combinations with nonlinear damper properties. *Procedia Iutam*, 22:251–258, 2017.

[34] John T Kasselmann, Henry Dorsett, and Harold J Burkett. Sway control means for a trailer, April 6 1976. US Patent 3,948,567.

[35] Paul Fancher and Chris Winkler. Directional performance issues in evaluation and design of articulated heavy vehicles. *Vehicle System Dynamics*, 45, 07 2007.

[36] Yuping He. *Design of Rail Vehicles with Passive and Active Suspensions Using Multidisciplinary Optimization, Multibody Dynamics, and Genetic Algorithms*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2003.

[37] Tao Sun, Yuping He, Ebrahim Esmailzadeh, and Jing Ren. Lateral stability improvement of car-trailer systems using active trailer braking control. *Journal of Mechanics Engineering and Automation*, 2(9):555–562, 2012.

[38] Ossama Mokhiamar. Stabilization of car-caravan combination using independent steer and drive/or brake forces distribution. *Alexandria Engineering Journal*, 54(3):315–324, 2015.

[39] Seshadri Sankar, Seshardri Sankar, Subhash Rakheja, and Alain Piché. Directional dynamics of a tractor-semitrailer with self-and forced-steering axles. *SAE transactions*, pages 528–541, 1991.

[40] Yuping He, Hoda Elmaraghy, and Waguih Elmaraghy. A design analysis approach for improving the stability of dynamic systems with application to the design of car-trailer systems. *Modal Analysis*, 11(12):1487–1509, 2005.

[41] Andrew MC Odhams, Richard L Roebuck, David Cebon, and Chris Winkler. Dynamic safety of active trailer steering systems. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of multi-body dynamics*, 222(4):367–380, 2008.

[42] Krishna Rangavajhula and HS Jacob Tsao. Command steering of trailers and command-steering-based optimal control of an articulated system for tractor-track following. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 222(6):935–954, 2008.

[43] Moustafa Ei-Gindy, Nezih Mrad, and Xiaoya Tong. Sensitivity of rearward amplication control of a truck/full trailer to tyre cornering stiffness variations. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 215(5):579–588, 2001.

[44] Paul Fancher, Chris Winkler, Robert Ervin, and Hongli Zhang. Using braking to control the lateral motions of full trailers. *Vehicle System Dynamics*, 29(S1):462–478, 1998.

[45] Darrell Etherington. The tesla semi tackles this classic truck safety problem using tech,, 2017. URL `https://techcrunch.com/2017/11/16/the-tesla-semi-tackles-this-classic-truck-safety-problem-using-tech/`.

[46] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[47] Gurusamy Jeyakumar and C. Shunmuga Velayutham. Distributed mixed variant differential evolution algorithms for unconstrained global optimization. *Memetic Computing*, 5, 12 2013.

[48] Muhammed Dangor, Olurotimi A Dahunsi, Jimoh O Pedro, and M Montaz Ali. Evolutionary algorithm-based PID controller tuning for nonlinear quarter-car electrohydraulic vehicle suspensions. *Nonlinear dynamics*, 78 (4):2795–2810, 2014.

[49] Edward M Kasprzak and Kemper Lewis. An approach to facilitate decision tradeoffs in pareto solution sets. *Journal of Engineering Valuation and Cost Analysis*, 3:173–187, 01 2000.

[50] Song Zhang, Hongfeng Wang, and Min Huang. Dominate gradient strategy based on pareto dominant and gradient method. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 4943–4948. IEEE, 2016.

[51] Kalyanmoy Deb and Shivam Gupta. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization*, 43(11):1175–1204, 2011.

[52] Joshua Knowles and David Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, 1, 01 1999.

[53] E. Zitzler. *Evolutionary algorithms for multiobjective optimization: Methods and applications*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.

[54] Varvara Mytilinou and Athanasios J. Kolios. A multi-objective optimisation approach applied to offshore wind farm location selection. *Journal of Ocean Engineering and Marine Energy*, 3(3):265–284, 2017.

[55] Franklin Mendoza, Jose L Bernal-Agustin, and José A Domínguez-Navarro. NSGA and SPEA applied to multiobjective design of power distribution systems. *IEEE Transactions on power systems*, 21(4):1938–1945, 2006.

[56] Arindam Deb, Jibendu Roy, and Bhaskar Gupta. Performance comparison of differential evolution, particle swarm optimization and genetic algorithm in the design of circularly polarized microstrip antennas. *Antennas and Propagation, IEEE Transactions on*, 62:3920–3928, 08 2014. doi: 10.1109/TAP.2014.2322880.

[57] Saku Kukkonen and Jouni Lampinen. GDE3: The third evolution step of generalized differential evolution. In *2005 IEEE congress on evolutionary computation*, volume 1, pages 443–450. IEEE, 2005.

[58] Thomas L. Magnanti. *Twenty Years of Mathematical Programming*. 1989. URL `https://books.google.ca/books?id=9xiaoAEACAAJ`.

[59] Anagha Philip Antony and Elizabeth Varghese. Comparison of performance indices of PID controller with different tuning methods. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pages 1–6. IEEE, 2016.

[60] Jussara MS Ribeiro, Murillo F Santos, MJ Carmo, and MF Silva. Comparison of PID controller tuning methods: analytical/classical techniques versus optimization algorithms. In *2017 18th international Carpathian control conference (ICCC)*, pages 533–538. IEEE, 2017.

[61] Jakob Vesterstrom and Rene Thomsen. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, volume 2, pages 1980–1987. IEEE, 2004.

[62] Nurhan Karaboga and Bahadir Cetinkaya. Performance comparison of genetic and differential evolution algorithms for digital fir filter design. In *International Conference on Advances in Information Systems*, pages 482–488. Springer, 2004.

[63] Malcolm JA Strens, Mark Bernhardt, and Nicholas Everett. Markov chain monte carlo sampling using direct search optimization. In *ICML*, pages 602–609. Citeseer, 2002.

[64] Manasa Madhavi Puralachetty and Vinay Kumar Pamula. Differential evolution and particle swarm optimization algorithms with two stage initialization for PID controller tuning in coupled tank liquid level system. In *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 507–511. IEEE, 2016.

[65] Andri Mirzal, Shinichiro Yoshii, and Masashi Furukawa. PID parameters optimization by using genetic algorithm. *arXiv preprint arXiv:1204.0885*, 2012.

[66] Michigan State University, Carnegie Mellon University, and University of Detroit Mercy. Control tutorials for matlab simulink. URL `http://ctms.engin.umich.edu/CTMS/index.php?aux=Home`.

[67] Fateme Azimlu, Shahryar Rahnamayan, Masoud Makrehchi, and Pedram Karimipour-Fard. Designing solar chimney power plant using metamodeling, multi-objective optimization, and innovization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 731–742. Springer, 2019.

[68] Zakiya Alfughi, Shahryar Rahnamayan, and Bekir Yilbas. Multi-objective solar farm design based on parabolic collectors. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 22(2):256–270, 2018.

[69] Brian Jujnovich and David Cebon. Comparative performance of semitrailer steering systems. In *7th International Symposium on Heavy Vehicle Weights & Dimensions, Delft, The Netherlands, Europe*, 2002.

[70] Ashish Tewari. *Advanced control of aircraft, spacecraft and rockets*, volume 37. John Wiley & Sons, 2011.

[71] Ruina Dang, Jianqiang Wang, Shengbo Eben Li, and Keqiang Li. Coordinated adaptive cruise control system with lane-change assistance. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2373–2383, 2015.

[72] Mutaz Keldani, Khizar Qureshi, Yuping He, and Ramiro Liscano. Design and optimization of a robust active trailer steering system for car-trailer

combinations. In *SAE Technical Paper*. SAE International, 04 2019. doi: 10.4271/2019-01-0433. URL https://doi.org/10.4271/2019-01-0433.

[73] Bang G. Zheng, Yi B. Huang, and Cong Y. Qiu. LQR+PID control and implementation of two-wheeled self-balancing robot. *Applied Mechanics and Materials*, 590:399–406, 06 2014. URL http://search.proquest.com.uproxy.library.dc-uoit.ca/docview/1542940398?accountid=14694. Copyright - Copyright Trans Tech Publications Ltd. Jun 2014; Last updated - 2014-12-09.

[74] Robert C. Nelson. *Flight stability and automatic control*. McGraw-Hill series in aeronautical and aerospace engineering. McGraw-Hill Ryerson, Limited, 1989. ISBN 9780070462182.

[75] Hans Prem, Austroads, and Australia. National Road Transport Commission. *Comparison of Modelling Systems for Performance-based Assessments of Heavy Vehicles: (performance Based Standards NRTC/Austroads Project A3 and A4) : Working Paper*. National Road Transport Commission, 2001. ISBN 9780642544896. URL https://books.google.ca/books?id=XrN-YgEACAAJ.

[76] Yuping He, Amir Khajepour, J McPhee, and Xiaohui Wang. Dynamic modelling and stability analysis of articulated frame steer vehicles. *International Journal of Heavy Vehicle Systems*, 12(1):28–59, 2004.

[77] B. P. Minaker and R. J. Rieveley. Automatic generation of the non-holonomic equations of motion for vehicle stability analysis. *Vehicle System Dynamics*, 48(9):1043–1063, 2010. doi: 10.1080/00423110903248702. URL https://doi.org/10.1080/00423110903248702.

[78] MFJ Luijten. Lateral dynamic behaviour of articulated commercial vehicles. *Eindhoven University of Technology*, 2010.

[79] Qiushi Wang and Yuping He. A study on single lane-change manoeuvres for determining rearward amplification of multi-trailer articulated heavy vehicles with active trailer steering systems. *Vehicle System Dynamics*, 54 (1):102–123, 2016.

[80] Douglas A Lawrence and Wilson J Rugh. Gain scheduling dynamic linear controllers for a nonlinear plant. *Automatica*, 31(3):381–390, 1995.

[81] Wilson J Rugh and Jeff S Shamma. Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.

[82] Kalyanmoy Deb and Aravind Srinivasan. Innovization: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1629–1636. ACM, 2006.

[83] Joshua B Kollat and Patrick M Reed. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Advances in Water Resources*, 29(6):792–807, 2006.

[84] CA Coello Coello and M Salazar Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pages 1051–1056. IEEE, 2002.

[85] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.

[86] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.