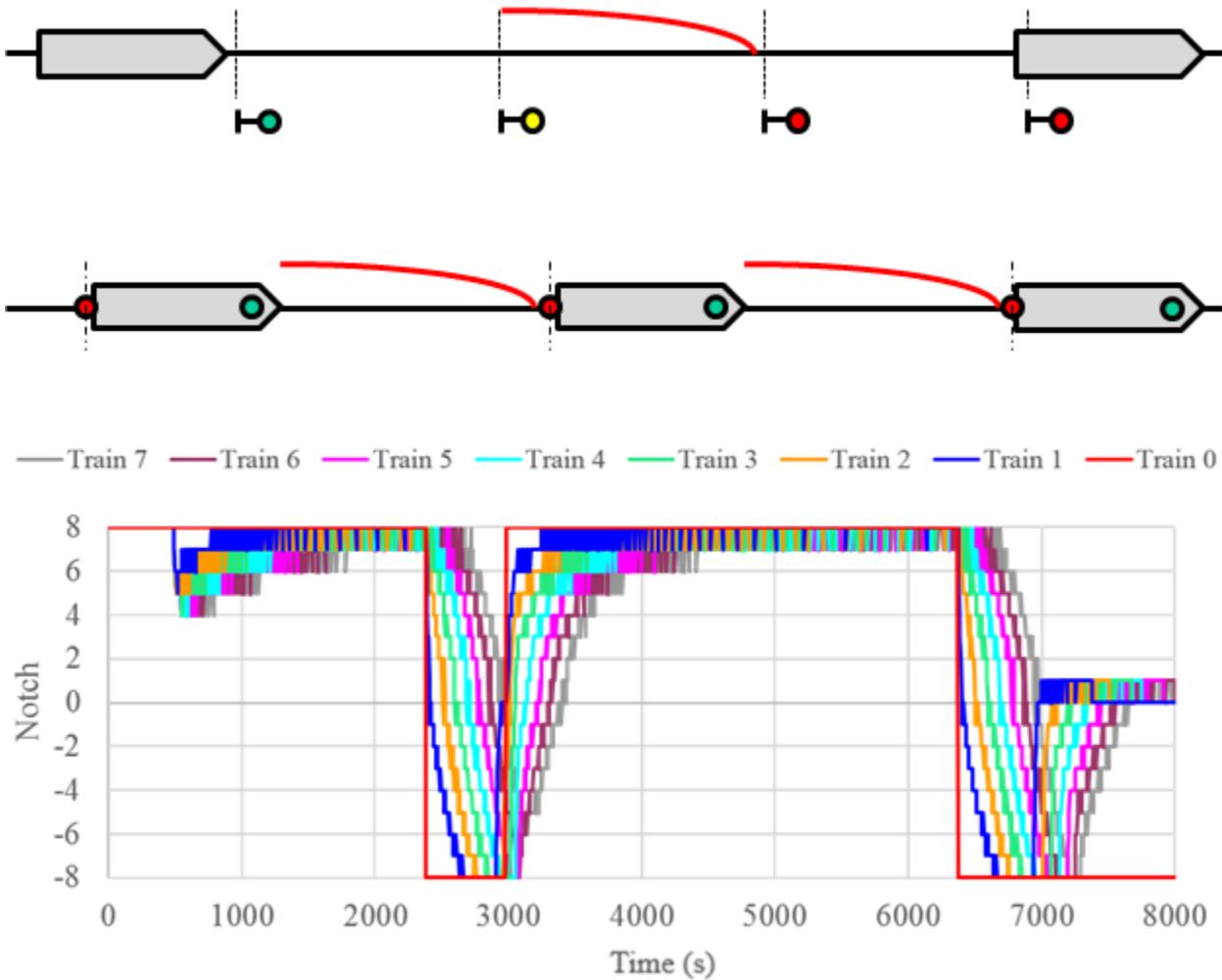




## Leveraging Connected Highway Vehicle Platooning Technology to Improve the Efficiency and Effectiveness of Train Fleeting under Moving Blocks



NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof. Any opinions, findings and conclusions, or recommendations expressed in this material do not necessarily reflect the views or policies of the United States Government, nor does mention of trade names, commercial products, or organizations imply endorsement by the United States Government. The United States Government assumes no liability for the content or use of the material contained in this document.

NOTICE

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report.

## REPORT DOCUMENTATION PAGE

*Form Approved*  
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE</b> (DD-MM-YYYY)		<b>2. REPORT TYPE</b> Technical Report		<b>3. DATES COVERED</b> (From - To) 01/2020-01/2021	
<b>4. TITLE AND SUBTITLE</b> Leveraging Connected Highway Vehicle Platooning Technology to Improve the Efficiency and Effectiveness of Train Fleeting under Moving Blocks				<b>5a. CONTRACT NUMBER</b> 693JJ6-19-C-000024	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> C. Tyler Dick <a href="https://orcid.org/0000-0002-2527-1320">https://orcid.org/0000-0002-2527-1320</a> , Daniel B. Work <a href="https://orcid.org/0000-0003-0565-2158">https://orcid.org/0000-0003-0565-2158</a> , Kuilin Zhang <a href="https://orcid.org/0000-0002-8180-5016">https://orcid.org/0000-0002-8180-5016</a> , Bo Zou <a href="https://orcid.org/0000-0003-4485-5548">https://orcid.org/0000-0003-4485-5548</a> , Geordie S. Roscoe <a href="https://orcid.org/0000-0002-3319-9293">https://orcid.org/0000-0002-3319-9293</a> , Pooria Choobchian <a href="https://orcid.org/0000-0002-6169-6625">https://orcid.org/0000-0002-6169-6625</a> , Yanbing Wang <a href="https://orcid.org/0000-0002-3988-8356">https://orcid.org/0000-0002-3988-8356</a> , Yun-Chu Hung <a href="https://orcid.org/0000-0003-2218-670X">https://orcid.org/0000-0003-2218-670X</a>				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> University of Illinois at Urbana-Champaign, University of Illinois at Chicago, Michigan Technological University, Vanderbilt University, New York Air Brake Corporation				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b> <b>DOT/FRA/ORD-22/29</b>	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> U.S. Department of Transportation Federal Railroad Administration Office of Railroad Policy and Development Office of Research, Development, and Technology Washington, DC 20590				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b>  This document is available to the public through the FRA <a href="#">website</a> .					
<b>13. SUPPLEMENTARY NOTES</b>  COR: Francesco Bedini Jacobini					
<b>14. ABSTRACT</b> Future advanced Positive Train Control systems may allow North American railroads to introduce moving blocks with shorter train headways. This research examines how closely following trains respond to different throttle and brake inputs. Using insights from connected automobile and truck platooning technology, six different following train control algorithms were developed, analyzed for stability, and evaluated with simulated fleets of freight trains. While moving blocks require additional train spacing beyond minimum safe braking distance to account for train control actions, certain following train algorithms can help minimize this distance and balance fuel efficiency and train headway by changing control parameters.					
<b>15. SUBJECT TERMS</b> Train control and communication, Positive Train Control, PTC, moving blocks, Intelligent Railroad Systems, connected vehicles, train fleeting, train platooning, adaptive cruise control, energy efficiency, line capacity					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b> 89	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>			<b>19b. TELEPHONE NUMBER</b> (Include area code)

# METRIC/ENGLISH CONVERSION FACTORS

## ENGLISH TO METRIC

### LENGTH (APPROXIMATE)

1 inch (in) = 2.5 centimeters (cm)  
 1 foot (ft) = 30 centimeters (cm)  
 1 yard (yd) = 0.9 meter (m)  
 1 mile (mi) = 1.6 kilometers (km)

### AREA (APPROXIMATE)

1 square inch (sq in, in<sup>2</sup>) = 6.5 square centimeters (cm<sup>2</sup>)  
 1 square foot (sq ft, ft<sup>2</sup>) = 0.09 square meter (m<sup>2</sup>)  
 1 square yard (sq yd, yd<sup>2</sup>) = 0.8 square meter (m<sup>2</sup>)  
 1 square mile (sq mi, mi<sup>2</sup>) = 2.6 square kilometers (km<sup>2</sup>)  
 1 acre = 0.4 hectare (ha) = 4,000 square meters (m<sup>2</sup>)

### MASS - WEIGHT (APPROXIMATE)

1 ounce (oz) = 28 grams (gm)  
 1 pound (lb) = 0.45 kilogram (kg)  
 1 short ton = 2,000 pounds (lb) = 0.9 tonne (t)

### VOLUME (APPROXIMATE)

1 teaspoon (tsp) = 5 milliliters (ml)  
 1 tablespoon (tbsp) = 15 milliliters (ml)  
 1 fluid ounce (fl oz) = 30 milliliters (ml)  
 1 cup (c) = 0.24 liter (l)  
 1 pint (pt) = 0.47 liter (l)  
 1 quart (qt) = 0.96 liter (l)  
 1 gallon (gal) = 3.8 liters (l)  
 1 cubic foot (cu ft, ft<sup>3</sup>) = 0.03 cubic meter (m<sup>3</sup>)  
 1 cubic yard (cu yd, yd<sup>3</sup>) = 0.76 cubic meter (m<sup>3</sup>)

### TEMPERATURE (EXACT)

$$[(x-32)(5/9)] \text{ } ^\circ\text{F} = y \text{ } ^\circ\text{C}$$

## METRIC TO ENGLISH

### LENGTH (APPROXIMATE)

1 millimeter (mm) = 0.04 inch (in)  
 1 centimeter (cm) = 0.4 inch (in)  
 1 meter (m) = 3.3 feet (ft)  
 1 meter (m) = 1.1 yards (yd)  
 1 kilometer (km) = 0.6 mile (mi)

### AREA (APPROXIMATE)

1 square centimeter (cm<sup>2</sup>) = 0.16 square inch (sq in, in<sup>2</sup>)  
 1 square meter (m<sup>2</sup>) = 1.2 square yards (sq yd, yd<sup>2</sup>)  
 1 square kilometer (km<sup>2</sup>) = 0.4 square mile (sq mi, mi<sup>2</sup>)  
 10,000 square meters (m<sup>2</sup>) = 1 hectare (ha) = 2.5 acres

### MASS - WEIGHT (APPROXIMATE)

1 gram (gm) = 0.036 ounce (oz)  
 1 kilogram (kg) = 2.2 pounds (lb)  
 1 tonne (t) = 1,000 kilograms (kg)  
 = 1.1 short tons

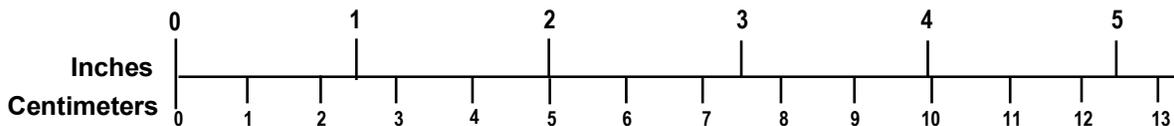
### VOLUME (APPROXIMATE)

1 milliliter (ml) = 0.03 fluid ounce (fl oz)  
 1 liter (l) = 2.1 pints (pt)  
 1 liter (l) = 1.06 quarts (qt)  
 1 liter (l) = 0.26 gallon (gal)  
 1 cubic meter (m<sup>3</sup>) = 36 cubic feet (cu ft, ft<sup>3</sup>)  
 1 cubic meter (m<sup>3</sup>) = 1.3 cubic yards (cu yd, yd<sup>3</sup>)

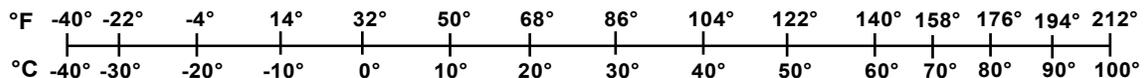
### TEMPERATURE (EXACT)

$$[(9/5) y + 32] \text{ } ^\circ\text{C} = x \text{ } ^\circ\text{F}$$

## QUICK INCH - CENTIMETER LENGTH CONVERSION



## QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSION



For more exact and or other conversion factors, see NIST Miscellaneous Publication 286, Units of Weights and Measures. Price \$2.50 SD Catalog No. C13 10286

Updated 6/17/98

## **Acknowledgements**

---

The academic team would like to acknowledge the important contributions made to this research by Chuck Coyle, Bill Kleftis, Rich Matusiak, and Keith Wait, all with New York Air Brake Corporation, in serving as industry partners for this project.

The project team also acknowledges the assistance of Matthew Parkes and Daniel Holmes, Undergraduate Research Assistants with the Rail Transportation and Engineering Center at the University of Illinois at Urbana-Champaign, in conducting simulations and collecting and processing the data required to complete this research.

# Contents

---

Executive Summary .....	1
1. Introduction .....	3
1.1 Background .....	3
1.2 Objectives .....	9
1.3 Overall Approach .....	10
1.4 Scope .....	11
1.5 Organization of the Report .....	11
2. Literature Review .....	12
2.1 Railway Train Control Algorithms.....	12
2.2 Car-Following Models without Connected Vehicles .....	17
2.3 Car-Following Models with Connected Vehicles .....	20
2.4 Truck Platooning Control Algorithms.....	21
2.5 Conclusions from the Literature Review.....	22
3. Multiple-Train Simulation Model and Baseline Simulation .....	24
3.1 Simulation Model Development .....	24
3.2 Baseline Simulation.....	29
4. Following Train Control Algorithm Development.....	35
4.1 Following Train Control Problem and Objectives .....	35
4.2 Proportional Derivative Controller.....	38
4.3 Modified Proportional Derivative Controller.....	40
4.4 Cooperative Adaptive Cruise Control Controller.....	41
4.5 Predictive Cooperative Adaptive Cruise Control Controller .....	43
4.6 Kinematic Adaptive Cruise Control (KACC) Controller.....	46
5. Simulated Performance of Following Train Control Algorithms.....	50
5.1 Experimental Design .....	50
5.2 Simulation Experiment Results .....	52
5.3 Results Summary.....	72
6. Conclusion.....	73
7. References .....	74

## Illustrations

---

Figure 1. Train Operations and Headway Under a) Fixed Wayside Signal Blocks and b) Moving Blocks .....	4
Figure 2. Stringline Diagram for Four-Train Baseline Simulation.....	30
Figure 3. Absolute and Relative Train Speed over Time for Baseline Simulation.....	31
Figure 4. Throttle Notch Position for First 1,000 Seconds and Full Baseline Simulation .....	32
Figure 5. Train Spacing and Distance to PTC Enforcement for Baseline Simulation.....	33
Figure 6. PD Controller Pseudocode .....	40
Figure 7. Block Diagram of CACC Controller.....	42
Figure 8. CACC Controller Pseudocode.....	44
Figure 9. KACC Controller Pseudocode .....	49
Figure 10. Naïve Controller versus PD Controller Notch Performance .....	53
Figure 11. Naïve Controller versus PD Controller Excess Spacing .....	54
Figure 12. Position-Only Test Plan, Notch Performance Comparison.....	60
Figure 13. Position-Only Test Plan, Spacing Excess Comparison .....	61
Figure 14. More Communication Test Plan, Notch Performance Comparison.....	67
Figure 15. More Communication Test Plan, Spacing Excess Comparison .....	68
Figure 16. Average Separation and Fuel Consumption Tradeoff for KACC Controller.....	71

## Tables

---

Table 1. Classification of Railway Train Control Algorithm Papers.....	13
Table 2. Summary of Train Plans .....	50
Table 3. Summary of Simulation Scenarios .....	51
Table 4. Position-Only Test Plan, PD Controller Parameter Variation .....	55
Table 5. Position-Only Test Plan, MPD Controller Parameter Variation.....	56
Table 6. Position-Only Test Plan, KACC Controller Parameter Variation .....	57
Table 7. Position-Only Test Plan, CACC Controller Parameter Variation .....	58
Table 8. Position-Only Test Plan, Best Input Values Summary.....	59
Table 9. Position-Only Test Plan, Results Summary.....	59
Table 10. More Communication Test Plan, MPD Controller Parameter Variation.....	63
Table 11. More Communication Test Plan, KACC Controller Parameter Variation .....	64
Table 12. More Communication Test Plan, PCACC Controller Parameter Variation .....	65
Table 13. More Communication Test Plan, Best Input Values Summary.....	66
Table 14. More Communication Test Plan, Results Summary.....	66
Table 15. Time Step and Interpolation Method, MPD Controller Summary .....	70
Table 16. Time Step and Interpolation Method, KACC Controller Summary .....	70

## Executive Summary

---

Emerging advanced Positive Train Control (PTC) systems may enable North American railroads to introduce moving blocks that allow for shorter train headways in the future. Although operating fleets of trains at shorter headways can conceptually increase mainline capacity, small fluctuations in the speed of the lead train may be amplified as each following train reacts and attempts to maintain safe braking headway. The resulting rapid changes in throttle and brake settings can reduce following train fuel efficiency, generate in-train forces, and, as headways fluctuate, diminish any capacity gains from shortened headways. To address this challenge, this research leverages emerging highway vehicle platooning technology to improve the efficiency and effectiveness of fleeting trains at minimum headways under moving blocks. The research aims to better understand how closely following trains respond to different throttle and brake control algorithms, and, using insights gained from automobile and truck platooning technology, develop improved train control algorithms balancing fuel efficiency and train headway. A project team led by the University of Illinois at Urbana-Champaign, and including the University of Illinois at Chicago, Michigan Technological University, Vanderbilt University and the New York Air Brake Corporation, conducted the research between January 2020 and January 2021.

From an extensive literature review, the project team found that improved following train control algorithms under moving blocks have not been previously researched within the North American heavy-haul freight and passenger railway context. Past research has been limited to heavy-rail transit (subway) and high-speed passenger rail applications, mainly in the international context. Because of the types of rail vehicles involved and their performance characteristics, researchers have adapted car following models to the problem of controlling transit and high-speed rail train headways in a string stable manner. A key difference between light-duty highway vehicle following models and the requirements of a heavy-haul freight rail application is the need to consider in-train forces, a finite number of throttle settings, more complicated braking systems, and additional lag time in response to control inputs.

To evaluate following train control algorithms, the project team developed a detailed multi-train performance simulator by adapting previous University of Illinois train performance simulation work and making improvements with the aid of insights and validation data from New York Air Brake. The multi-train model simulated the performance of individual trains within a fleet subject to different throttle and brake inputs as they attempted to follow a lead train with a specified throttle/brake plan and speed profile. A following train control algorithm determined the throttle and brake commands for following trains. The algorithm attempted to minimize train headway while still enforcing the moving-block safe braking distance, not exceeding the maximum authorized speed, and not generating excessive in-train forces. To compare different control algorithms, the model output performance metrics, including average train separation, number of following train incursions into the safe braking distance (PTC braking enforcement), fuel consumption, and standard deviation of throttle and dynamic brake notch settings.

As a baseline, the project team simulated a naïve algorithm designed to mimic the performance of a train crew reacting solely to the position of the train ahead. This approach required rapid oscillations between throttle and dynamic braking to manage train headways. This operating pattern was not fuel efficient, may place excessive strain on locomotive components, and may also create excessive in-train forces. Following trains amplified the behavior of preceding trains,

demonstrating string instability and increasingly aggressive control actions leading to PTC braking enforcements.

To improve on this behavior, the project team adapted highway vehicle platooning control methods to the heavy-haul freight rail domain. To investigate different families of control laws, the team formulated five following train control algorithms that more intelligently considered information on the status of the train ahead when specifying throttle or brake settings for each following train. A stability analysis determined the combined range of control algorithm parameters over which a fleet of trains was expected to exhibit string stability. With string stability, following trains attenuated the actions of preceding trains, and each successive train required less aggressive acceleration and braking rates to maintain headways.

To evaluate and compare all control algorithms, the multi-train model simulated each algorithm over a series of 18 different train fleeting scenarios. Each scenario involved a different factorial combination of freight train type, maximum and minimum throttle/brake notch, and number of acceleration and braking cycles over a 200-mile tangent route with zero gradient. The team simulated two different communication topologies, one where following trains only had information on the position of the train ahead, and a second where additional speed, acceleration, and throttle/brake status could be communicated between trains.

The simulation results suggest that certain families of control laws were better than others at managing train separation and fuel consumption within train fleets. Certain controllers were fast-acting but demonstrated notch instability when attempting to minimize headways. Other controllers were slow-acting and required a large baseline train spacing to avoid an incursion into the safe braking distance and a corresponding PTC braking enforcement. While all cases required additional train spacing beyond the minimum safe braking distance to account for train control actions, certain following train control algorithms helped minimize this distance. The control laws developed through this research exhibited an efficiency and headway tradeoff that may allow railway operators to optimize performance according to their specific business objectives by changing algorithm parameters. Relative to the scenario where only information on the position of the train ahead is known, the headway and fuel efficiency performance of the train control algorithms could be improved by communicating additional information on the speed and acceleration of the train ahead. These benefits of additional communication were enhanced when the frequency of train position reports and controller updates were increased. This result suggests that enhanced communication may be essential to effectively managing train fleets and achieving the full capacity benefits of moving blocks.

Additional research and simulation experiments should be conducted to evaluate the most promising control algorithms on actual rail corridor topography and train fleet operating conditions. The results of this research would allow industry practitioners to develop improved locomotive driver advisory and semi-autonomous adaptive train cruise control systems for the operation of fleets of trains under moving blocks, and railroad operators to make more informed decisions regarding the potential fuel efficiency and capacity benefits of these systems.

# 1. Introduction

---

This Technical Report summarizes the research, development, and simulation of following train control algorithms for moving block operations. The research was conducted to improve the efficiency and effectiveness of fleeting trains at minimum headways under future moving-block control systems. In developing and evaluating potential control algorithms to adapt to heavy-haul freight rail applications, the research leveraged existing and developing connected highway vehicle and truck platooning technology. Research was conducted between January 2020 and January 2021 by a consortium led by the University of Illinois at Urbana-Champaign and including the University of Illinois at Chicago, Michigan Technological University, and Vanderbilt University as academic partners, and the New York Air Brake Corporation as an industry partner. The research was sponsored by the Federal Railroad Administration (FRA) through the 2018 Broad Agency Announcement (BAA) on Intelligent Railroad System Research.

## 1.1 Background

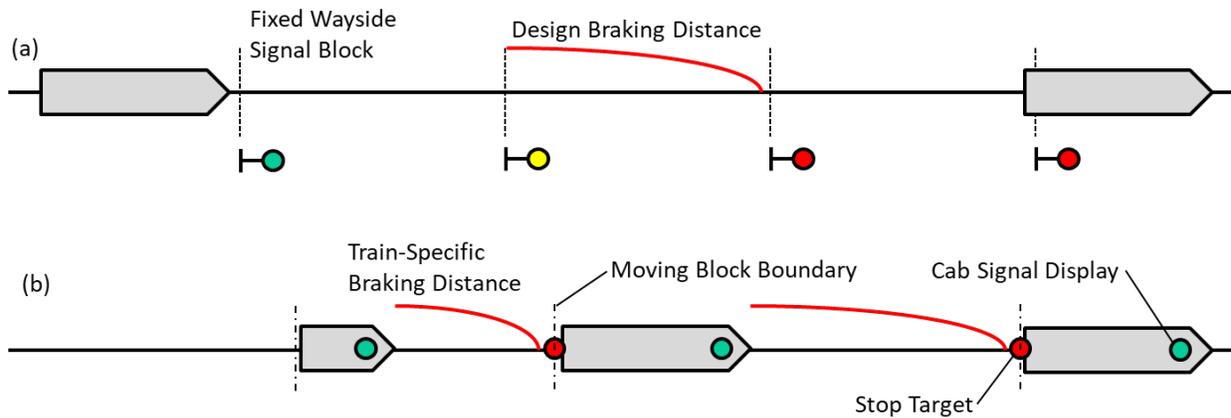
### 1.1.1 Research Motivation

North American railroads face increasing demand for safe, efficient, and reliable freight and passenger transportation. Rising energy costs and competition from other transportation modes give railroads an economic incentive to optimize train handling for maximum fuel efficiency. Similarly, the high cost of constructing additional track infrastructure to increase capacity and improve reliability provides railroads with a strong financial motivation to increase the productivity of their existing mainlines by reducing the headway between trains.

Facing similar efficiency and capacity demands, highway transportation has turned to intelligent transportation systems technology. Adaptive cruise control systems developed for highway vehicles are being combined with other technologies to support autonomous highway vehicle operations. These systems react to the surrounding terrain and traffic stream to control the vehicle throttle and brake in a manner that promotes fuel efficiency. More advanced versions of these systems use connected vehicle technology to form platoons of closely spaced highway vehicles that travel together in a coordinated manner. The reduced spacing between highway vehicles in a platoon decreases aerodynamic drag, increasing fuel efficiency, and creates capacity for additional vehicles. When a small number of autonomous vehicles operating in an ideal fuel-efficient manner are introduced into a traffic stream, they help to regulate the actions of all drivers, improving the overall flow and fuel efficiency of the traffic stream.

To improve the fuel efficiency of rail transportation, railways have implemented similar driver advisory systems to advise locomotive operators on optimal throttle and brake commands. Systems such as LEADER from New York Air Brake have become progressively more sophisticated in their ability to support semi-autonomous adaptive train cruise control and even full autonomous “auto pilot” capability under test conditions in Western Australia. While systems such as LEADER can improve fuel efficiency, their ability to reduce train headway and increase capacity is constrained by the existing wayside block signal system used to control railway traffic and maintain safe separation between trains. Advanced Positive Train Control (PTC) systems currently under development may replace this wayside system with virtual or moving blocks that allow for shorter train headways. Conceptually, eliminating the headway inefficiency of fixed signal blocks with lengths established by the braking distance of the

poorest-performing train allows trains to follow each other at minimum safe braking distances (Figure 1). The reduced headway between trains can increase capacity on double-track main lines, such as those found on key freight corridors with high traffic density or lines in urban areas with combinations of freight and commuter rail service (Dick, 2000; Dingler et al., 2009; Dingler et al., 2010; Dick et al., 2019; Diaz de Rivera and Dick, 2021).



**Figure 1. Train Operations and Headway Under a) Fixed Wayside Signal Blocks and b) Moving Blocks**

Although operating highway and rail vehicles at shorter headways can conceptually increase capacity, it is difficult to implement in practice. In actual operating settings, the lead vehicle rarely maintains a perfectly constant speed due to changes in grade, curvature, and wind speed and direction. Small fluctuations in the speed of the lead vehicle are amplified by each trailing vehicle as they react and attempt to maintain a minimum required headway. Tests of highway vehicles with adaptive cruise control have shown that platoons of vehicles exhibit a “rubber band” effect where the headway between subsequent vehicles rapidly expands and shrinks as following vehicles attempt to match the speed of the lead vehicle. The rubber band effect becomes more pronounced for each subsequent vehicle in the platoon, with the final car or truck requiring frequent throttle and brake adjustments to rapidly change speed and maintain the desired headway. Under these unstable conditions, the fuel efficiency of the vehicles at the end of the platoon can deteriorate and, as headways begin to fluctuate, any capacity gains from shortened headways are diminished. Recent research has attempted to minimize these effects and improve the efficiency and effectiveness of platooning through vehicle-to-vehicle communication. Connecting vehicles to share throttle and brake commands can allow for new control algorithms that improve fuel efficiency and the ability of the vehicles to maintain a constant minimum headway. This research aims to learn more about these vehicle control algorithms and how they can be used to solve similar problems when fleets of closely following trains are operated under advanced PTC systems.

A naïve implementation of PTC with moving blocks is likely to encounter similar problems with varying headway and decreased fuel efficiency. These effects are likely to be more pronounced in a railway environment compared to a highway application. Due to their lower power-to-weight ratio, trains are more sensitive to changes in grade and curvature than highway vehicles, and locomotives have a limited number of throttle settings, making it more difficult for the lead train to maintain a constant speed. While highway vehicles respond rapidly to changes in throttle and brake settings, the size and inertia of diesel-electric locomotive prime movers makes them

slow to respond to changes in throttle settings, and it may take many seconds for all of the brakes on a train to apply or release. With longer reaction times to control inputs, the following trains are likely to experience difficulty in maintaining the minimum headway distance. Additional complications arise from the sensitivity of train braking distance to small changes in gradient and train speed. To avoid an enforcement brake application from the PTC system, a train-following at the minimum headway when cresting a vertical curve will need to reduce speed and increase headway as its safe braking distance increases. Subsequent following trains will need to match this new lower speed but may also be able to follow at closer headways as their safe braking distance becomes shorter at lower speeds.

### **1.1.2 Technological Challenge**

Technology to control fleets of closely spaced trains with moving blocks has been advanced in urban transit operations. Moving-block operations and coordinated train control has been demonstrated and deployed on several subway lines worldwide, including the Docklands Light Railway in London, the L Line in New York, and Subway Line 2 in Beijing (Gao et. al., 2015). Although these transit installations involve many of the same component technologies as a moving-block system on a conventional mainline railway (such as moving-block train control architecture, train-to-train communication, and driver advisory or automated control systems with underlying control algorithms), they are applied to a more highly constrained operating environment than a conventional railway. Since transit systems typically feature vehicles of similar length, acceleration and braking properties, consistent speeds, highly structured schedules, and dedicated guideways, there are fewer technical challenges to address when developing control algorithms for fleeting trains under moving blocks.

In Europe, provision for operations with moving blocks is made via the European Train Control System (ETCS) Level 3. ETCS is characterized by bidirectional information transmission between the train and the radio block center (RBC), train positioning by EUROBALISES (beacons) installed every 1,000 meters between the rails, continuous and safe speed control, onboard train integrity check, and virtual block control by RBCs. Currently, there are no plans to introduce ETCS Level 3 on any European rail lines in the short term. However, a number of lines are expected to test ETCS levels with moving blocks in the coming years. Like previous transit applications, the ETCS experience is not directly applicable to mainline heavy-haul freight railways. European railway operations typically use electric traction; have a focus on short, fast, and light passenger trains; and impose limits on the length and weight of freight trains that constrain train sizes to be far less than those operated on North American railways. Greater homogeneity in train size and weight normalizes train performance. More consistent train braking and throttle response removes one additional source of complexity for the train-following control algorithm.

In the context of mainline line-haul freight and commuter, regional, and intercity passenger rail operations in North America, technology to control fleets of closely spaced trains with moving blocks is still under development. Locomotive driver advisory systems have been in operation for the past 10 to 15 years on several of the major Class I railroads to improve fuel efficiency. However, because of the constraints of the wayside block signal control system, these systems have not been optimized to the specific problem of managing headways between following trains under moving-block PTC.

Outside North America, tests of locomotive auto-pilot systems on iron ore railway lines in Western Australia have been conducted under moving-block PTC. Although these lines are heavy-haul freight operations, the trains are of relatively consistent length and weight and use electronically controlled pneumatic (ECP) brakes. In contrast, North American line-haul freight and passenger operations involve train fleets with a wider range of performance characteristics and standard air brakes that have a less predictable reaction time and performance compared to ECP brakes. Thus, continued refinement and development of following train control algorithms is still required to effectively and efficiently operate closely spaced trains under moving blocks.

Through transit operations and demonstration applications in Europe and Australia, the relevant component technologies for operating closely spaced trains under moving blocks have been demonstrated in a relevant operating environment, but there is room for additional development and improvement of train-following control algorithms to improve overall performance.

Within the context of highway intelligent transportation systems, various tests and demonstrations of passenger vehicle platooning technology have been conducted over the past 20 years. Platoons of heavy commercial trucks are a more recent phenomenon, with several research and demonstration projects over the past 5 years, and efforts by private firms to commercialize truck platooning technology. Developing control algorithms that account for the mass, acceleration, braking, and control response characteristics of heavy trucks has proven to be a challenge compared to those for passenger vehicles. The extreme physical and performance characteristics of heavy-haul freight trains compared to trucks and passenger vehicles reinforce the need and challenge of developing efficient following train control algorithms that are the subject of this research.

### ***1.1.3 Development of Highway Vehicle Platooning and Control***

The idea of highway vehicle platooning with coordinated control can be traced to a demonstration in New York City in 1925, when a driverless automobile was remotely controlled by an operator in a trailing vehicle using radio waves. The system, developed by the Houdina Radio Control Company, was mainly used for advertising and publicity stunts, but the U.S. military explored various practical applications of the technology. The idea of autonomous vehicles was popularized during the 1939 World's Fair, when the American industrial designer Norman Bel Geddes proposed automatic highway vehicles guided by magnetic rails embedded in the road. In the 1960s, researchers at Ohio State University realized this idea when they developed an automated automobile with computer-controlled electro-hydraulic steering, acceleration, and braking. The automatic steering system used coils that sensed magnetic fields generated by cables embedded in the pavement. To maintain separation between vehicles when traveling in a platoon, a physical reel-type measuring device stretched between the lead human-controlled vehicle and the trailing automated test vehicle monitored headway distance and relative speed. An onboard computer running a basic control algorithm used the measured distance and speed (and their respective instantaneous rates of change) to control the vehicle's accelerator and brake automatically. Although the physical measuring system was impractical, this research forms the foundation of subsequent research on algorithms to control closely spaced platoons of highway or railway vehicles. The researchers themselves acknowledged that more sophisticated control algorithms would be required to achieve the roadway capacity and efficiency benefits of automated vehicles (Fenton and Olson, 1969).

In 1972–73, the European ARAMIS project platooned 25 small transit vehicles on a French test track. The vehicles used ultrasonic and optical range sensors to operate at 50 mph with a separation distance of 1 foot between each vehicle. The European Prometheus Project (1980–1995) advanced communications, vehicle control, and artificial intelligence to create intelligent vehicles designed to run on an advanced road system. The vehicles had fully automated steering and longitudinal control, and Volkswagen conducted test-track trials at highway speed.

Experiments to develop better platooning technologies have continued in the U.S. Most notably, the Partners for Advanced Transportation Technology (PATH) project at UC Berkeley started in 1986 as a collaboration between State and local governments and pioneered several intelligent systems. The collaboration developed the Automated Highway System that was used to demonstrate four-car platoons in 1994 and eight-car platoons in 1997. The project was recently used to demonstrate three-truck platoons operating at 14-foot intervals.

Ongoing research at Michigan Technological University involves developing model predictive control models, implementing real-time solution algorithms, and conducting real-world testing and simulation analysis for connected and automated vehicles. This research includes developing algorithms to support platoon driving or cooperative and adaptive cruise control (CACC) applications. An optimal control model based on model predictive control (MPC) has been developed and implemented for the CACC application using vehicle-to-vehicle communication technology. The MPC-based CACC control model aims to improve stability, robustness, driving safety, and energy efficiency of a platoon of CAVs. The control model is formulated with the objective of maintaining a constant time headway and is solved in real-time with a rolling horizon framework. A fleet of eight fully connected Chevrolet Volt vehicles is employed in real-world road testing of the control algorithms. Based on this testing, Zhao and Zhang (2018; 2020) have reported on the performance of the MPC-based CACC model and real-time solution algorithms under uncertain traffic conditions. Although it may be more challenging to maintain a constant time (or distance) headway for fleets of trains compared to highway vehicle platoons due to the low power-to-weight ratio of trains, this framework provides key insights for developing train fleet control algorithms.

Practical implications and performance of highway vehicle platoons and their overall effect on the traffic stream are the subject of past and ongoing investigations at Vanderbilt University. The research has investigated the instability observed in vehicle platoons and the impact of traffic waves resulting from this “rubber band” effect on vehicle braking events and fuel economy (Stern et al., 2018). The research has progressed to examine how a number of autonomous and connected vehicles operating with appropriate control algorithms can minimize the occurrence and impact of these traffic waves, even if human drivers still control several of the vehicles in the platoon. Through field experiments, the team has developed control strategies to dampen the effect of traffic waves and produce more stable vehicle platoons. Since the overall stability of a train fleet and the ability of trains to maintain minimum headway is key to achieving the potential capacity benefits of PTC with moving blocks, these developments also influence the design of the train fleet control algorithms.

#### **1.1.4 Development of Train Fleets with Moving Blocks**

Within the rail mode, the ability of North American railway operators to form platoons of trains traveling together at minimum headways has been constrained by railway traffic control systems that operate on fixed or discrete control blocks that can be occupied by only one train at a time.

Although fleets of trains can be operated together in close succession, the wayside signal block control system and track circuits found on most high-density rail lines in North America have limited ability to determine the exact location of each train. To ensure safety and adequate separation between trains, the block signal control system creates additional spacing between trains beyond the minimum safe braking distance. Thus, true “train platoons” at minimum headways analogous to those achieved with highway vehicles have not been possible in the North American freight and passenger railway operating environment.

PTC, or related advanced communications-based train control systems that use GPS and other positioning technologies to continuously monitor the precise location of each train, have the potential to unlock the benefits of train platoons. Although the current implementation of PTC systems on U.S. railways are primarily designed to function as a safety overlay on existing block signal control systems, advanced versions of PTC may eliminate the fixed control blocks and adopt a moving-block control architecture. The moving-block concept allows a train to receive a movement authority between any two locations, rather than being constrained to the fixed-block boundaries of conventional signaling. In addition, movement authority limits update automatically and more frequently than in conventional wayside signal systems. By dynamically creating control blocks based on the exact location and braking performance of each train, and continuously moving these control blocks along with each train, advanced PTC systems are being developed with moving blocks that can facilitate train fleets that function as platoons at minimum headways.

In the early 2000s, the North American Joint Positive Train Control (NAJPTC) project was conducted to develop, test, and demonstrate PTC capabilities – including flexible block operations – in a corridor with both freight and passenger service (Polivka et al., 2009). In 2001, the NAJPTC system was developed and tested on a 120-mile corridor of the Union Pacific Railroad in Illinois that also hosted regional intercity passenger service. Development work moved to the Transportation Technology Center in 2006. Although current PTC installations incorporate many elements developed under the NAJPTC project, the moving-block architecture was not among them. The NAJPTC project highlighted many important technical challenges associated with moving blocks, including the bandwidth of the radio links, data latency, and the need for more adaptive and robust braking and control algorithms.

Maximizing capacity benefits while minimizing train headways under moving blocks requires trains to operate right at the edge of the safe braking distance enforced by the PTC system. To avoid repeated enforcement brake applications while maintaining minimum headways, it is envisioned that train crews will be aided by driver advisory systems or semi-autonomous, adaptive train cruise control systems. The development of these types of systems began with the first implementation of automated driverless train operations on transit systems during the 1960s. In addition to the transit systems mentioned previously in Section 1.1.2, the SkyTrain in Vancouver, Canada, opened in 1986, is notable for its use of both autonomous trains and moving blocks managed from a wayside computer control system. North American freight applications of automated trains have been limited to industrial railway operations with dedicated trains operating on closed-loop systems using electric traction. Examples of these systems include the Carol Lake Railway in Canada and the Muskingum Electric Railroad in Ohio. Although closed in 2002, the Muskingum Electric was the first automated freight railway in the U.S. when it opened in 1968.

While autonomous applications have been limited, the North American railway industry has focused on developing locomotive driver advisory systems and technology-assisted train operations to improve train handling and energy efficiency. In the 1970s, the railway industry began to develop computer programs to simulate the physics of train movements in detail. These programs allowed railroads to refine train makeup and locomotive assignments, minimize in-train forces, and determine the set of throttle and brake commands to optimize train speed, schedule, and fuel efficiency. As computing power increased and hardware became smaller and more durable, computers onboard locomotives could be used for optimal driving strategies in real-time and provide control instructions to the train crew.

New York Air Brake began the development of its Locomotive Engineer Assist/Display & Event Recorder (LEADER) driver advisory system in the late 1990s, based on earlier train dynamics software first developed in the 1970s. LEADER uses integrated GPS train location information and a detailed database of track geometry and maximum authorized speeds to assist locomotive engineers in reducing fuel consumption while effectively managing trip time and minimizing in-train forces. Over the past 20 years, LEADER has evolved from an information-only display to a proactive system that assists train crews in selecting throttle and brake settings. Through continuing research and development on the iron ore railways in Australia, LEADER is currently evolving into the prototype for a full autopilot system capable of autonomous train operations. Under development for over a decade, testing of the New York Air Brake autonomous train system began in 2014. In 2017, an Australian iron ore train successfully operated in autonomous mode for 60 miles with no crew members on board the locomotive. In July 2018, trains began to make the entire 175-mile trip from Rio Tinto's Tom Price mine to the port of Cape Lambert in autonomous mode without a train crew on board. The train was monitored remotely by the operations center in Perth, more than 900 miles away. The LEADER system currently uses a simulation approach to make throttle and brake decisions; at a given instant, computers on board the locomotive simulate the results of possible throttle and brake adjustments and then select the alternative that produces the best outcome with respect to the track topography, maximum authorized speed, movement authority, and train operating plan.

One justification for advanced PTC with moving blocks is the capacity benefit of operating fleets of closely spaced trains (Diaz de Rivera et al., 2020a; 2020b). To realize this benefit, the next step for systems such as LEADER and Trip Optimizer from GE Transportation is to integrate their locomotive control optimization framework with moving-block PTC to more effectively and efficiently manage the separation distance between trains in a fleet. Currently, driver advisory systems independently optimize the performance of a single train relative to the topography, maximum authorized speed, and limits of movement authority. With the exception of a pace setting to avoid excess idling when meeting opposing trains on a single track, the systems have limited ability to coordinate operations with other trains for greater overall efficiency. The next step in advancing these systems is to leverage train-to-train communication and connected vehicle technologies to develop control algorithms for more effective and efficient train fleetings with moving blocks.

## **1.2 Objectives**

The objective of this research was to leverage existing and developing connected highway vehicle platooning technology to improve the efficiency and effectiveness of fleetings trains at minimum headways under moving blocks. The research aimed to better understand how closely

following trains respond to different throttle and brake control algorithms, and, using insights gained from connected automobile and truck platooning technology, develop improved train control algorithms that allow railway operators to optimally balance fuel efficiency and train headway. The results of this research will allow industry practitioners to develop improved locomotive driver advisory and semi-autonomous adaptive train cruise control systems for the operation of fleets of trains under moving blocks, and railroad operators to make more informed decisions regarding the potential fuel efficiency and capacity benefits of these systems.

### **1.3 Overall Approach**

To improve the efficiency and effectiveness of operating fleets of closely spaced trains with moving blocks, this research used simulation to better understand the baseline headway and fuel efficiency implications of this type of operation. The project team then developed and simulated improved “train-following” control algorithms to better manage train fleeting operations, adapting published approaches to highway vehicle platooning where applicable.

In implementing this approach, the project was divided into three major tasks. In the first task, the project team reviewed published research literature to better understand the algorithms and quantitative approaches used by state-of-the-art car following and truck platooning models. The literature review also examined published research on train-following models and algorithms to optimize train energy efficiency and locomotive driver advisory systems.

In the second task, the project team adapted an existing train performance calculator into a multiple-train simulation model to specifically model the headway and energy consumption of a train fleet as it traverses a representative mainline study route. Modifications included developing a low-level controller to interpret the desired acceleration and braking rates requested by the following train control algorithm into specific locomotive throttle notch and brake settings for simulated freight trains. The model initially used naïve following train control logic to demonstrate the effect of aggressive train operator behavior on overall fuel consumption and time spent at the minimum safe braking distance headway as the train fleet traversed a study route under advanced PTC with moving blocks.

In the third task, the project team developed a series of new train-following algorithms designed to balance fuel consumption and time at the minimum headway for heavy-haul freight trains on mainline corridors. In adjusting the traction and braking force of each train along the study route, the algorithms must account for the instantaneous location and speed of all trains and preserve the minimum instantaneous safe stopping distance between any two successive trains. Based on the literature review and past project team experience with control algorithms for platoons of connected highway vehicles, the team explored various algorithmic approaches. Control algorithms were first formulated analytically and then, where possible, analyzed for the stability of following vehicle performance. During this formulation process, the control algorithms were coded into the multiple-train simulation model to observe the efficiency and headway performance of simulated train fleets. The observed performance was used to refine and improve the algorithm formulation and adjust control parameters in an iterative manner. A total of six control algorithms were fully developed for two different scenarios: the reactive case, where following trains responded to observed changes in the speed of the train ahead; and the connected case, where train-to-train communication was used to inform following trains of changes in the throttle and brake settings of the train ahead. Once fully developed, each following train control algorithm was implemented with the multiple-train-following model, and

18 different train-following scenarios were simulated to evaluate and compare the fuel efficiency and headway performance of each algorithm. Comparison of simulation results between the controllers suggests that certain families of control algorithms offer the most promising performance, and that control parameters can be adjusted to favor headway or fuel efficiency performance based on the business objectives of a particular railroad or train operator.

## **1.4 Scope**

Since this is an initial investigation of train-following control algorithms for heavy-haul freight applications, the focus was on screening and identifying families of control algorithms that show potential for good train-following performance. To reduce the complexity of the train-following scenarios considered by the control algorithms, various simplifying assumptions were made to limit the scope of the project. The study corridor consists of level, tangent tracks, so that grade and curve resistance could be neglected. Each simulated train fleet consists of identical trains, eliminating heterogeneity in train performance within a fleet. Lead train behavior was limited to scenarios using throttle and dynamic brakes. Algorithms that perform poorly under these ideal conditions could be screened out – as they are unlikely to perform well under more complex and realistic scenarios involving actual route topography, heterogeneous trains, and lead trains making air brake applications. Although the scope of simulation scenarios used to evaluate and compare the train following algorithms only involved three types of heavy-haul freight trains (i.e., intermodal, bulk unit, and manifest trains), the team anticipated that similar control laws could be applied to operations involving passenger or commuter rail operations.

The project scope was limited to two different communication topology scenarios: a basic scenario where following trains only had information about the position of the train ahead at discrete intervals, and a scenario with train-to-train communication of throttle and brake actions to following trains. The latter scenario may allow the following trains to anticipate changes in the speed of the lead train and make more efficient and timely control decisions that preserve both headway and fuel efficiency. While such train-to-train communication is difficult today, the communication network provided by future advanced PTC systems with moving blocks may facilitate this type of coordination in the future.

## **1.5 Organization of the Report**

The remainder of the technical report is organized into sections corresponding to the main project tasks. Section 2 reviews published literature on railway train control algorithms, car-following models with and without connected vehicle technology, and truck platooning control. Section 3 describes the multi-train-following simulation model developed to evaluate the train-following control algorithms. Section 4 details the development and formulation of the six control algorithms posited through this research. Section 5 outlines the train-following simulation experiments and summarizes and compares the performance of the six control algorithms. Section 6 summarizes the conclusions of this research.

## 2. Literature Review

---

The project team conducted an extensive literature review to determine the current state of knowledge regarding control algorithms for closely following trains, light-duty highway vehicles, connected light-duty highway vehicles, and heavy trucks. Specifically, the literature review was targeted at four distinct topic areas covering previous research on optimizing the energy efficiency and headway of closely following groups of different vehicles:

- Railway train control algorithms
- Car-following models
- Car-following models that leverage connected vehicle technology.
- Truck platooning control algorithms

The following sections summarize the key findings of the literature review in each of these four topic areas.

### 2.1 Railway Train Control Algorithms

The papers collected on the topic of railway train control algorithms ([Table 1](#)) can be classified by the scale of the operation they consider (single train, train-following, or network) and on the approach used (control algorithm or optimization). While this project focuses on the train-following problem, and thus papers describing train-following algorithms are most relevant, select references on single-train models and overall train network models can provide important insights on approaches and key assumptions germane to all train performance simulation and train control algorithm development efforts. Papers on single trains and train networks will be summarized first before the more detailed discussion of the train-following research is presented.

#### 2.1.1 *Single Trains*

As described by Albrecht et al. (2016), the classic single-train control problem is to minimize the energy required to drive a train from one station to the next within a given time. The authors summarized previous research to support the optimal heavy-haul freight train control strategy of applying maximum power to reach maximum authorized track speed, then reducing power to hold speed constant, then coasting followed by applying maximum braking force. Speed-holding must be interrupted by pulses of power to overcome steep grades and periods of coasting to negotiate steep downhill sections. In this manner, the optimal throttle and braking strategy for a given train over a given route becomes a problem of determining the optimal time and location to switch between maximum power, maximum brake, speed holding, and coasting. In introducing a general solution to this optimization problem, the authors discuss the relative differences between modeling a heavy-haul freight train as a distributed mass compared to a point mass. The authors also indicate that there are still unsolved problems relating to optimal driving strategies for fleets of trains: “The most pressing research challenges for the future in this area are to develop optimal control policies for trains traveling in the same direction on the same line in such a way that safe separation is maintained between trains.”

**Table 1. Classification of Railway Train Control Algorithm Papers**

Scope	Control	Optimization
Single Train	Barney et al., 2001 Jaekel & Albrecht, 2014 Panou et al., 2013 Song et al., 2011 Xia & Zhang, 2011 Zhuan & Xia, 2006	Albrecht et al., 2016
Train-Following	Alikoc et al., 2013 Durmus et al., 2013 Gao et al., 2015 Ho, 1998 Karredla & Srinivas, 2014 Li & Guan, 2009 Li & Gao, 2007 Li & Gao, 2013 Li et al., 2005 Li et al., 2011 Li et al., 2015 Liu, 2016 Ning, 1998 Pan & Zheng, 2014 Takagi, 2012 Tang & Li, 2007 Wang et al., 2012 Xu et al., 2014 Xun et al., 2013 Yang et al., 2010 Ye & Li, 2013 Zhao et al., 2016 Zhou & Mi, 2012	Kraay et al., 1991 Tang et al., 2015 Wang & Goverde, 2016
Train Network	Gordon & Lehrer, 1998 Polivka et al., 2009 Su et al., 2015 Takeuchi et al., 2003	Wang, 2014 Wang et al., 2013a Wang et al., 2013b Wang et al., 2014 Ye et al., 2013

Jaekel and Albrecht (2014) also discussed the relative merits of modeling a freight train as a distributed “mass strap” compared to a point mass, documenting specific deviations between the two approaches over a study route. This comparison was made in the context of evaluating three different approaches to calculating the movement of a train over a route: infinitesimal calculus, the explicit time-step Euler method (based on piece-wise constant acceleration), and Gauss-Legendre-quadrature. Barney et al. (2001) described how simplifying assumptions must be made

when implementing these approaches to calculate train braking distances and perform other train performance calculations.

A shortcoming of modeling a freight train as a point mass is that in-train forces cannot be considered. Zhuan and Xia (2006) concluded that optimizing the speed profile of a single heavy-haul freight train requires managing in-train forces in addition to minimizing energy consumption. The authors noted that in-train forces are rarely considered in non-heavy-haul scenarios, limiting the utility of the work in the passenger and transit domain. Trains can be modeled as a set of point masses connected by non-linear springs. In a subsequent paper, Xia and Zhang (2011) developed a linear-quadratic regulator (LQR) controller to optimize a compound objective function that weights the relative importance of minimizing energy and managing in-train forces for a single train. However, Song et al. (2011) argued that it is infeasible to fully model and control a train as a set of spring-connected masses because the control laws become complex, given the precision of braking and acceleration controls available. This paper highlights the need to consider the actual control “notches” in train acceleration and braking when developing a law to control a real train. The authors use complex mathematical reduction to develop a relatively simple single-train control law and validate its effectiveness through simulation.

Finally, Panou et al. (2013) investigated some practicalities of implementing an optimal single-train control algorithm that is relevant to this research effort. The authors considered the pros and cons of computing optimal speed profiles on board the train, off board at a central location, or off board at one of many distributed locations across the network. There were advantages and disadvantages to each approach, mainly due to the tradeoff between limited computation power on board the train and higher data transmission requirements for off board solutions. Polivka et al. (2009) described how radio frequency bandwidth is limited and the radios and protocol that can transmit and receive adequate amounts of data over the extremely varied and relatively long distances required are very hard to develop. Polivka et al. further discussed how these and other technical challenges must be overcome before a standalone vital moving-block control system can be developed and implemented.

### **2.1.2 Train Networks**

The train network control and optimization problem attempts to simultaneously determine a speed profile for each train operating on a given segment with the objective of minimizing the global energy consumption without sacrificing running time performance. This is most critical for railway operations that use electric traction power systems and have the capability to capture and reuse regenerated braking energy (Su et al., 2015). Gordon and Lehrer (1998) used a neural network to increase headway between trains and reduce wasted energy consumed by following trains speeding up and slowing down too often. Takeuchi et al. (2003) examined this problem in the context of moving- and fixed-block control systems, noting that multiple trains accelerating at close headways in moving blocks can draw high peak power loads. The authors noted that moving block offers the highest capacity, but train acceleration should be limited to avoid propagating delays between trains when short headways are implemented. Ye et al. (2013) proposed optimizing the system by adding a term to the standard car following model to account for regenerative braking and the ability of accelerating trains within the same substation region to use this energy.

Wang et al. (2013a) developed a mixed-integer linear program (MILP) to simultaneously optimize the trajectories of a leading and following train to lower overall energy consumption with regenerative braking under moving block signals within the context of the Beijing subway system. The authors noted that as the number of trains in this fleet grows, the size of the problem grows very quickly and computation time will be substantially larger. Subsequent research described a suboptimal control scheme with mode vector constraints to decrease MILP solution times for controlling multiple trains under moving blocks, again in the transit context (Wang et al., 2013b). The authors then implemented a greedy approach that first solves the optimal trajectory of the lead train. Then, based on the optimal control inputs of the leading train, the trajectory planning problem for the following train is solved (Wang et al., 2014). The research also compared pseudospectral approaches to the MILP formulation, noting that although they both have similar control performance, the MILP requires less computation time. Wang (2014) further extended this research to determine the optimal train schedule for a single, electrified subway line under moving blocks. The resulting problem is nonlinear and nonconvex, requiring a new iterative convex programming (ICP) approach to solve.

### **2.1.3 Train-Following**

The train-following problem attempts to determine an optimal set of throttle and brake settings for a following train given a fixed lead train trajectory. In one of the earliest descriptions of the train-following problem, Kraay et al. (1991) noted that pacing trains can result in energy savings. The authors proposed set generation with heuristic filtering to effectively solve the problem of optimal train trajectories under pacing. Wang and Goverde (2016) developed a multiple-phase optimal control model to enforce a “green wave” policy of clear signals for a following train under fixed blocks. The authors used the pseudospectral method to convert the complex problem into a non-linear programming problem to be solved more quickly and efficiently.

Published research on these types of train control algorithms has mainly focused on rail transit applications. Since transit systems operate on fixed schedules with a limited variety of small, responsive, lightweight vehicles using electric traction and regenerative braking, there are fewer variables to consider compared to typical line-haul railway applications. Thus, published research on transit control algorithms has focused on pre-calculation of an optimal driving strategy with dynamic programming or genetic algorithms as opposed to algorithms that function in real time and are “in the loop,” providing recommended control inputs based on current operating conditions.

As an example, Tang et al. (2015) optimized the energy efficiency of a subway system operating with regenerative braking on close headways by developing an optimal throttle and brake control algorithm. Several versions of the algorithm were developed. The first version independently optimized the energy efficiency of the following train with no information on the throttle and brake status of the lead train. The second version optimized the energy efficiency of the following train with information on the throttle and brake status of the lead train transmitted via train-to-train communications. The final version coordinated control of both the lead and following trains via train-to-train communications to optimize the overall energy efficiency of the system. In all cases, the algorithm enforced minimum safe headway distances between trains and the inter-station travel times dictated by the subway schedule.

While the optimization approach to train-following has merits due to its high solution quality, locomotives have very limited processing ability on board. Additionally, quicker updates on

board the locomotive to control the train as well as more rapid occupancy updates could allow trains to run closer together. Thus, a control-based approach rather than an optimization approach has the potential to perform better while also avoiding costly investments in locomotive computing power or radio frequency bandwidth and the back office.

Research into this area commenced around the time moving-block systems were first being developed and deployed because rapid and continual state updates are a prerequisite of most control algorithms. Ning (1998) compared two possible types of moving block system – absolute or relative braking distance – and links the train-following control problem to the most basic car-following model, where acceleration is related to following speed difference with a reaction time delay included. Ho (1998) showed that moving-block systems can substantially increase track capacity over fixed-block systems through a multi-train movement simulator, a technique that many researchers will continue to use. Simulation is required for evaluating these sorts of problems because they are simply too complex for an analysis-based approach to yield results.

Seven years after these two papers, Ning and others at Beijing Jiaotong University adapted the NaSch probabilistic cellular automaton model, which is normally used to describe single-lane highway traffic, to the railway domain to describe one single-direction track of a moving-block signaled railway (Li et al., 2005). While this first paper allowed overlapping braking curves like in highway traffic, a series of papers over the following years continued to adapt and extend the model. Li and Gao (2007) added a term to better capture the safety distance requirements of moving block, allowing them to better model a railway station. Tang and Li (2007) allowed an instantaneous braking rate, but calculated proper safety distances under this assumption. Li and Guan (2009) adapted the optimal velocity control form of the car-following equation to the cellular automaton model and allowed using the true safety stopping distance. Li et al. (2011) modified the optimal velocity control form of the car following equation in a different manner to obtain slightly better performance. Li and Gao (2013) extended these models to simulate a 3-aspect, fixed-block railway by adding different speed targets for each of the aspects that are all based on the optimal velocity car following equation. Ye and Li (2013) proposed a different form of the modified optimal velocity, car-following equation based on a planned travel time.

Other researchers also picked up the cellular automaton framework for moving-block railway simulation. Zhou and Mi (2012) utilized tempo-spatial constraints to represent movement authority and track speed restrictions, which allows for limiting train braking rates to reasonable values. Wang et al. (2012) used the existing cellular automaton models to analyze a mixture of trains with different maximal speeds and evaluate the effect of changing station density. Train overtaking was allowed at stations. They found that increasing station density results in a lower average speed and that decreasing the ratio of slow trains to fast trains results in lower average speeds for the slow trains and little change to the average speed of the fast trains. They demonstrated that existing cellular automaton models were good enough for analysis of a railway network. Zhao et al. (2016) utilized these models to compare the capacity of fixed-block, moving-block, and train convoy signaling. Train convoy signaling is also commonly referred to as virtual coupling. They found that train convoy signaling had 1.25 times the capacity of moving block, 3.75 times the capacity of fixed block, and the highest robustness to delay disturbances.

In addition to the cellular automaton simulation technique, more traditional simulation techniques have been used to analyze this problem. Yang et al. (2010) implemented a discrete, time-based simulation and found that the backwards propagation of train delay resulted in

suboptimal accelerations and decelerations with simple control laws. Takagi (2012) showed that simple kinematic equation-based control laws were enough assuming perfectly synchronous control of multiple trains. They also proposed a group local service that could effectively take advantage of this control methodology while still serving all stations. This service is also referred to as a skip-stop service. Pan and Zheng (2014) used an empirical safe following distance formula to derive simple control laws which were enough for control, assuming these trains could respond instantly and effectively to the requested accelerations. However, this specific safe following distance formula was limited to flat ground and identical trains. Karredla and Srinivas (2014) proposed a quasi-moving-block system to increase track utilization over fixed block, calculated an instantaneous adequate distance formula, and computed the steady-state capacity of the system. Liu (2016) proposed and evaluated a possible algorithm for ERTMS Level 3 (moving block), finding that the range of stable parameters was relatively limited and varied with desired headway. Thus, they concluded that proper train-following control parameters would be critical for the success of ERTMS Level 3.

While simulations are very useful for validating control laws, robust stability analysis based on mathematical control theory can be performed to prove the effectiveness of a set of control laws under all possible conditions. These formal proofs are most informative when supported by simulations demonstrating good performance. Alikoc et al. (2013) used the cluster treatment of characteristic roots (CTCR) to analytically determine the region of stability for moving-block trains following in a circle and in a straight line and validated using simulation. Durmus et al. (2013) used an adaptive PD controller to control train separation in moving block and found that it effectively mitigated measurement noise and step and sinusoidal disturbances. Xun et al. (2013) proposed a train headway adjustment model based on linear quadratic optimal control. Using a cellular automaton simulation model to test the headway adjustment approach, they found that delay time could be reduced by 11.8 percent when the dispatching headway was 75 seconds, and that computation time was not excessive even for large numbers of trains in the fleet. Xu et al. (2014) proposed using single-train, optimal-speed profiles to advance each train, with recalculations occurring at discrete events. They found that a discrete event-based train movement simulator was less computationally expensive than a discrete time-based simulator, and that simple control strategies still resulted in excessive accelerations and decelerations. Additionally, this paper shows that a hybrid approach between optimization and control may be possible given a good train simulator. Li et al. (2015) used an artificial potential function that used the locations of the two neighboring trains in addition to minimum and maximum headway constraints to coordinate cruise control over a fleet of trains. However, they found that the algorithm required changing the target speed if large disturbances were introduced into the system. Gao et al. (2015) proved that string stability was possible to achieve in moving-block railway operations with only the location information of the preceding and succeeding train using the Lyapunov stability theorem. They also found that using information from both the preceding and the succeeding trains was generally more stable than just using information from the preceding train, and that the addition of speed and acceleration information only marginally improved performance. Lastly, they used simulation to demonstrate the effectiveness of the proposed control laws.

## **2.2 Car-Following Models without Connected Vehicles**

This section briefly summarizes some of the relevant developments in the fields of car-following modeling of vehicles on uninterrupted roadways and an automated vehicle technology known as

*adaptive cruise control* (ACC). The two fields are closely related. Car-following models are used to describe the following behavior of a vehicle, which often results in a model that describes the vehicle's motion (e.g., velocity or acceleration) as a function of the car and the vehicle ahead. In contrast, ACC is a control strategy that describes how the acceleration of the vehicle should change to achieve a desired car-following behavior. The main distinction is that the car-following models are descriptive methods that describe how vehicles behave under human control, whereas adaptive cruise control algorithms are prescriptive techniques that describe how the vehicle should accelerate and decelerate to achieve a desired behavior. Below is a description of the developments of car-following models and then a summary of key adaptive cruise control algorithms.

### **2.2.1 Car-Following Models**

The modeling of the longitudinal behavior of human drivers in vehicles is referred to as car-following. These models are typically written as ordinary differential equations or discrete difference equations that describe the trajectory of a vehicle as a function of the properties of the vehicle and the car ahead. Common forms of the models describe how the velocity or acceleration of a vehicle changes as a function of the current vehicle velocity, the distance to the vehicle ahead, and/or the relative velocity to the vehicle ahead.

A historical overview of the development of car-following models is provided in Brackstone and McDonald (1999), with an emphasis on the experiments and data used to design the models. While a large number of car-following models exist, this section points out some of the most influential models. In 1958, Chandler, Herman, and Montroll introduced a seminal car-following theory known as a *follow-the-leader* (FTL) model, where vehicles are assumed to accelerate or decelerate to match the speed and/or a desired spacing to the vehicle ahead. The article is one of the foundations of transportation science because it 1) introduced a model for car-following; 2) illustrated that some car-following models (e.g., those that keep a constant spacing or those that keep a sufficiently small time headway) will give rise to string instability commonly recognized today as a phantom traffic jam; 3) used data collected from human driving studies to fit the model; and 4) determined that human drivers operate near the boundary of a regime that is string unstable. Later, Gazis et al. (1959) used the FTL model to show that the car-following behavior at the level of the vehicle is related to the macroscopic traffic flow properties of the link, described by the fundamental diagram.

An alternative to the FTL car-following model is the family of models known as the *optimal velocity* (OV) model (Bando et al., 1995), which assumes that a vehicle accelerates or decelerates to match the current speed to a desired or optimal velocity given the distance to the vehicle ahead. The models are simple but powerful; they also exhibit string instabilities depending on the parameters of the model. In other words, small disturbances in the traffic stream (when all vehicles operate according to the OV) can result in large speed variations that lead to phantom traffic jams. To experimentally demonstrate the string instability phenomenon, Sugiyama et al. (2008) placed 22 vehicles on a short, single-lane, circular track. Initially, drivers could maintain a uniform speed, but quickly the traffic on the ring degraded into a phantom traffic jam. A general approach to determine if a car-following model is string stable is provided in Wilson and Ward (2011). In Jiang et al. (2001), the OV model was extended to include a term that accounts for the relative velocity to the car ahead in addition to the space gap. The extended model,

known as a *full velocity difference model* (FVDM) is important from a control point of view because some ACC control policies can be viewed as variations of the FVDM.

Two popular models in the traffic simulation community are worth mentioning. First, the Gipps model (Gipps, 1981) was designed specifically for increased speed when simulating a large number of vehicles on a roadway, and modified versions of the model now form the starting point of models used in commercial traffic simulation products such as TSS Aimsun. Second, the intelligent driver model (IDM) car-following model was proposed in Treiber et al. (2000). Unlike other models, the IDM was derived from the standpoint of reasoning about observed traffic behavior rather than trying to reason about human driving behavior directly. It has since become a widely used model in traffic simulation and analysis.

### **2.2.2 Adaptive Cruise Control**

Many important steps have been taken in developing vehicle automation systems that longitudinally control the vehicle, known as *adaptive cruise control* (ACC) (Rajamani, 2011). Initially proposed in the research community, these systems are now widely available on many of the best-selling vehicles in the U.S. market either as a standard or optional feature. The commercially implemented systems typically rely on a sensor (such as a radar) to measure the space gap and relative velocity to the vehicle ahead, and a control algorithm to accelerate or decelerate the vehicle to maintain a desired following behavior. The systems do not use any communication between the vehicles, and thus are easy to deploy gradually (i.e., all vehicles do not need to be equipped for the system to work).

One of the core design principles of ACC is that the control design should be string stable. If the controller is string stable, then collections of ACC vehicles will not create phantom traffic jams when following one another. It is widely known that control policies that try to maintain constant spacing between the vehicles do not lead to string stable behavior. In contrast, constant time headway policies can be made to be string stable provided that the time headway between vehicles is sufficiently large. These concepts are the cornerstone of the *autonomous intelligent cruise control* (AICC) system developed in Ioannou and Chien (1993). Later, Swaroop et al. (1994) showed that the desired control torques were inversely proportional to the headway time.

Liang and Peng (1999) introduced an ACC system that included two different forms of system delay. One, called system delay, is used to model delays in sensor measurements of the vehicle's state. The other, called parasitic delay, is used to describe delays between a commanded acceleration action and the actual acceleration the vehicle experiences. An analysis was provided to show how string stability could be achieved by selection of the control parameters to account for the delays. The results of the controller were validated in simulation. Zhou and Peng (2005) developed a string stable adaptive cruise control system that offers string stability at higher flow rates than traditional, constant-time, headway-based controllers. It achieves better performance by considering that the actuation delay (servo-loop time constant) is not perfectly known, but the acceleration of the controlled vehicle is known. This information is then exploited in the design of the ACC system.

An interesting extension of ACC is the bilateral controller (Wang and Horn, 2019). Unlike standard ACC systems which only use information on the vehicle ahead, the bilateral controller also uses sensors to measure information to the vehicle behind. The authors illustrated how to design a string stable adaptive cruise control system using this additional information.

Because ACC systems will be implemented on vehicles with different performance characteristics, extensions of ACC have been proposed in Xiao and Gao (2011) that account for vehicle heterogeneity. The vehicles in the traffic stream are provided with different control parameters and are subject to different parasitic time delays and lags with different values of the following time headway. The article provides an approach to verify the string stability of a heterogeneous platoon of vehicles.

Given that commercial ACC systems are now being deployed on the roadways, there is a growing interest to use vehicle control systems to improve overall traffic flow. It was shown in theory (Wu et al., 2018) and in practice (Stern et al., 2018) that the introduction of a small fraction (as low as 5 percent) of longitudinally controlled vehicles can stabilize traffic flow and reduce the occurrence of phantom jams caused by non-controlled vehicles. In the experimental tests reported in Stern et al. (2018), the controlled vehicle could reduce fuel consumption of the total traffic stream by approximately 40 percent by eliminating the stop-and-go driving present before the control vehicle was activated. While the above control algorithms relied on classical model-based control approaches, there is now a growing interest in designing longitudinal control systems powered by artificial intelligence methods such as deep reinforcement learning (Wu et al., 2017).

## **2.3 Car-Following Models with Connected Vehicles**

This section of the review focuses primarily on coordinated vehicle control algorithms with communication. Pervasive wireless communication technologies, including *vehicle-to-vehicle* (V2V) and *vehicle-to-infrastructure* (V2I), allow individual vehicles to exchange information among other vehicles and infrastructures for cooperative automated driving decisions to improve safety, mobility, and energy efficiency. Shladover et al. (2012) provided definitions and operating concepts of cooperative adaptive cruise control (CACC) using communication-enabled vehicle-following and speed control in a communication environment. The literature on designing coordinated vehicle control algorithms considering wireless communication mainly addresses three issues: 1) the latency and quality of communication; 2) the string stability of control models; and 3) real-world road testing. Each topic is briefly summarized in the following sections.

### **2.3.1 Latency and Quality of Communication**

The issue of latency and quality of vehicular communication in coordinated vehicle control systems has been extensively studied. To reduce the negative effects of attacked or failed communication links in the vehicular communication networks, multiple papers discuss alternative designs of CACC explicitly considering these effects (Guo and Yue, 2014; van der Heijden et al., 2017; Gong et al., 2019). To significantly reduce inter-vehicle gaps, Milanés et al. (2014) designed a CACC controller introducing feedforward terms with wireless communication and confirmed improvements through road testing. To improve existing traffic flow modeling and communication, Jia and Ngoduy (2016a, 2016b) evaluated an enhanced cooperative driving system using V2X communication under various traffic scenarios. Qin et al. (2017) focused on the stability and frequency response of CACC under stochastic communication delays and then investigated a CACC design incorporating stochastic delay variations.

### **2.3.2 String Stability**

String stability is a key property of longitudinal controllers for CAV platoons. Ge and Orosz (2014) considered an acceleration-based connected cruise control (CCC) design and compare the string stability of platoons with different structures of connectivity. Considering local and multi-criteria string stability, Zhou et al. (2019) presented a model predictive control approach and provide mathematical proofs to verify its effectiveness. From previous studies, significant increases in highway capacity under different conditions by employing CACC systems were demonstrated through several experiments (Ploeg et al., 2011; Shladover et al., 2015). Talebpour and Mahmassani (2016) investigated the string stability of mixed traffic streams with varying percentages of different vehicle types and found the throughput increased as the market penetration rate of CAVs increased. Assuming detected information is shared with others via V2V and V2I communication, Zhou et al. (2017) proposed a rolling horizon stochastic optimal control strategy for both ACC and CACC systems under uncertainty and then proved they had better performance than deterministic controllers through simulations. Zhao and Zhang (2020) formulated a distributionally robust optimization-based model predictive control model to address CACC under traffic uncertainty and conduct an empirical analysis of string stability under traffic oscillations with multiple traffic shocks.

### **2.3.3 Real-World Testing**

Several CACC algorithms have been implemented in real-world road testing. Based on implementing existing technologies, Chang et al. (1991) used two Ford cars to show that the automated following vehicle successfully followed the manually driven lead vehicle through several different kinds of maneuvers. Öncü et al. (2014) used two CACC-equipped prototype vehicles to demonstrate the validity of their presented networked control system framework. Naus et al. (2010) tested two CACC-equipped vehicles to validate the proposed decentralized CACC design. Milanés and Shladover (2014) deployed four vehicles equipped with a commercial ACC system and a newly developed CACC controller to measure the actual responses of vehicles.

## **2.4 Truck Platooning Control Algorithms**

Empowered by recent advances in V2V communication and control technologies, interest in platooning possibilities within the trucking sector has grown. This part of the review mainly focuses on truck platooning control algorithms. Feritz (1999) presented longitudinal and lateral control of heavy-duty trucks for vehicle following, followed by Gehring and Fritz (1997), where practical results of a longitudinal control concept for truck platooning were obtained. The authors used distance measurement between trucks and vehicle-to-vehicle communication but did not consider road infrastructure. A two-layer control structure was proposed. The inner control loop includes a nonlinear acceleration controller linearizing a large part of the nonlinearities. Due to the different actuator systems, the dynamic behaviour of a truck was different during acceleration and braking. Furthermore, each truck may have had a different power train and load. Additional disturbances may also occur. A robust platoon controller was then introduced for the outer control loop by using sliding mode control design. Practical results of a platoon consisting of seven trucks showed that by using the proposed control concept, string stability could be achieved. Along this line of thought, subsequent papers studied further control algorithm designs for heavy duty vehicles. Liang et al. (2016) discussed a fuel-efficient control of heavy-duty

vehicle platoons. The authors tested some of the presented methods on real vehicles in traffic with experimental results on automatic control of heavy-duty vehicle platoons on a Swedish highway.

Cooperative driving, aiming at the compatibility of safety and efficiency of road traffic, means that automated vehicles drive by forming a flexible platoon over a couple of lanes with a short inter-vehicle distance while performing lane changing, merging, and leaving the platoon. The vehicles for a demonstration of this concept were equipped with automated lateral and longitudinal control functions with localization data by the differential global positioning system (DGPS) and the inter-vehicle communication function with 5.8-GHz dedicated short-range communication (DSRC) designed for the dedicated use in the demonstration. Kato et al. (2002) described the technologies of cooperative driving with automated vehicles and inter-vehicle communications for cooperative driving with automated vehicles and inter-vehicle communications. Xu et al. (2018) presented a robust control method for heterogeneous vehicle platoon subject to varying road slopes, aerodynamic drag, and wireless communication delay. Focusing on safety, comfort and with an overall aim of the comprehensive improvement of a vision-based intelligent vehicle, Zhang et al. (2017) proposed a novel advanced emergency braking system based on a nonlinear model predictive algorithm and established a vision-based longitudinal vehicle dynamics model considering the nonlinearities of vehicle dynamics. Lima (2018) designed a smooth and accurate model predictive controller tailored for industrial vehicles, where the main goal was to reduce the vehicle “wear and tear” during operation. The author showed controller effectiveness both in simulation and experimentally in a Scania construction truck. This doctoral dissertation showed that the proposed controller had promising performance in real experiments.

From previous studies, significant increases were demonstrated in highway capacity under different conditions by employing different control algorithms. Johansen et al. (1998) performed a linearization of a time-invariant nonlinear system and local control design in conventional gain-scheduled control design for the resulting set of linear time-invariant systems at a set of equilibrium points. Ying et al. (2014) applied a robust control technique to solve the problem of platoon system instability caused by velocity changes. The robust control strategy adopted in this paper was sliding mode control, which had the robustness with respect to its system parameters change and external disturbances. Wang and Nijmeijer (2015) studied a heterogeneous vehicle platoon equipped with CACC systems. Deng (2016) used a simulation framework to study HDV platooning and established the corresponding concept and operations. Finally, Ramezani et al. (2018) developed a micro-simulation model for truck CACC and platooning and studied their traffic impacts through a case study. It incorporated truck-following models that have been recently developed for the separate automated modes of CACC, ACC, and CC.

## **2.5 Conclusions from the Literature Review**

Improved train control algorithms to optimally balance train headway and fuel efficiency under moving blocks have not been previously researched within the North American heavy-haul freight and passenger railway context. Train-following under wayside block signal systems has been considered in optimizing individual freight train trajectories for energy efficiency. Research on train-following algorithms under moving blocks has been limited to heavy-rail transit (subway) and high-speed passenger rail applications. Because of the types of rail vehicles involved and their performance characteristics, transit and high-speed rail applications have

certain commonalities with highway applications of car-following models. Therefore, researchers have adapted car-following models to the problem of controlling transit and high-speed rail train headways. A common theme among many of the documented rail, highway, connected vehicle, and truck algorithms is the concept of string stability. To fill the identified knowledge gap and meet the objectives of this project, this research must consider how these models translate to the unique characteristics of the North American operating environment and the extreme performance characteristics of heavy-haul freight trains. A key difference between light-duty highway vehicle-following models and the requirements of a heavy-haul freight application is the need to consider in-train forces, a finite number of throttle settings, more complicated braking systems, and additional lag time in locomotive and train response to control inputs. While highway models do not model the vehicle in detail and typically assume it is a point mass, heavy-haul freight algorithms will need to consider the composition of the train and internal forces in greater detail. Heavy truck platooning models, which are more sensitive to vehicle performance characteristics, are a key bridge between current car-following models and future heavy-haul freight train-following algorithms.

### **3. Multiple-Train Simulation Model and Baseline Simulation**

---

Following the completion of the literature review, the second project task included three main components:

- Further develop and improve the capabilities of an existing single-train performance calculator previously developed by the University of Illinois.
- Adapt this calculator into a multi-train-following simulation model.
- Use the new model to simulate the baseline behavior of following trains under moving blocks without any intelligent control algorithms.

The following sections summarize the development of the multi-train-following simulation model used to evaluate potential train control algorithms for fleets of trains under moving blocks. The final sub-section presents the results of an example baseline simulation exhibiting poor train-following behavior. This baseline performance is presented to serve as a motivation for developing improved train control algorithms, as outlined in the following section.

#### **3.1 Simulation Model Development**

To evaluate the effectiveness of various train control algorithms, a detailed, discrete, time train performance simulator was developed in C++. A discrete time simulation approach was selected because train performance depends on numerous factors with non-linear and piecewise relationships, making an analytical solution to train performance intractable. Time was selected as the discretization variable, as opposed to distance or speed, because multiple subsystems, including the locomotive diesel prime movers and the train brake pipe, have performance that depends solely on time. The C++ language and programming environment was used for implementation to ensure that the simulation would run quickly while also allowing for a high level of detail in the model.

The train-following simulator was composed of two main components: the trains and a track network. Each train contains all parameters needed to calculate its performance, including aerodynamic, bearing, and rolling resistance coefficients, locomotive tractive effort curves, brake valve, brake pipe, and brake shoe characteristics, and the upcoming path through the network. The track network contained all turnouts, track segments, and all currently granted movement authorities. The approaches used to incorporate each of these subcomponents in the train-following simulator are explained in detail in the remainder of this section.

##### **3.1.1 Trains**

The overall train performance framework was adapted from previous train performance simulation work conducted by the University of Illinois and improved with the aid of insights and validation data provided by industry partner New York Air Brake.

##### **Resistance to Motion**

The four main components of train resistance are bearing and rolling resistance, aerodynamic resistance, grade resistance, and curve resistance. To simplify the scope of this project, the train control algorithms were tested on straight and level track, meaning that grade and curve resistance had no impact on performance.

Bearing resistance was computed for each axle on each railcar in the train as a constant 18 lbf. Rolling resistance was computed as  $1.5 \text{ lbf/ton} \times W$ , where  $W$  is the weight supported by the axle (including its own weight) in tons. Aerodynamic resistance was computed for each railcar in the train and considered both the perpendicular surface area not occluded by the preceding railcar and the surface area parallel to the direction of travel (skin friction).

### **Locomotive Tractive Effort**

Tractive effort curves are used to determine the force output of each locomotive given its throttle notch or dynamic brake setting and speed. To determine the force output when changing notches, limits were set on the rate of increase and decrease of tractive effort, engine power, and dynamic brake force. During the simulation, the value of each of these variables at the previous time step were stored to enforce these limits. Engine power is related to the tractive effort curve using the relationship  $\text{power} = (\text{tractive effort}) \times \text{speed}/\text{efficiency}$ , where  $\text{efficiency} = 0.9$ . Additionally, the model enforced timeouts (on the order of seconds) when switching from power throttle to dynamic braking.

### **Air Brakes**

For modeling purposes, train air brakes were divided into four main components:

- Main reservoir and equalizing reservoir on each locomotive
- Brake pipe
- Control valves and downstream empty/load sensors and pistons
- Brake rigging and brake shoes

The locomotive independent brake components are currently not simulated as a separate system by the model. To account for these braking forces in the model, the locomotive brakes were connected to the train brake pipe and act as if they were just a railcar at the start of the train.

To reduce the complexity of air brake calculations in the model, the main reservoirs and equalizing reservoirs were modeled in a simplified manner. The main reservoirs were not simulated, the equalizing reservoirs instantly responded to control changes, and the equalizing reservoirs had an infinite supply of air at the specified pressure. These assumptions were reasonable, as most of the simulated train-following scenarios only require the use of dynamic brakes to maintain train speed control and ensure a safe following distance between trains.

### **Brake Pipe**

The brake pipe was simulated as the combination of a simple reservoir that was always in equilibrium, and a set of waves traveling at a set speed marking increases or decreases in pressure. There was a “tap” from the pipe at each connected component (control valve and equalizing reservoir) that stored the current pressure in the brake pipe at that position.

When air is taken from (or added to) the brake pipe, the model considers the tap pressure and total brake pipe volume in determining the resulting brake pipe pressure. Using an equation for air power transmitted given upstream pressure, downstream pressure, and the minimum cross-section connecting them, the maximum energy exchanged in one time step is calculated. This

energy is then limited by the total energy required to reach equilibrium between the connected reservoirs, which is why total volume must be known. Once this energy change is calculated for each component connected to the brake pipe, a full energy balance is performed on the brake pipe to ensure the total energy taken out of the brake pipe still leaves the average pressure above atmospheric pressure. The new average brake pipe pressure is then set using these values.

With the energy taken from (and added to) the brake pipe at each tap all properly limited, the appropriate pressure waves (one in each direction along the brake pipe) can be added at each tap. The pressure change that this wave propagates is calculated as the quotient (**energy change**)/(**total brake pipe volume**). This pressure change is used because it is equal to the change in average brake pipe pressure due to this energy change. Thus, when all waves have reached the ends of the brake pipe, each tap pressure will exactly match the average brake pipe pressure.

### Control Valves

Each control valve has three states: release, apply, and emergency. State changes occur in response to pressure differences between various components at the control valve. The connected auxiliary and emergency reservoirs are included in the simulation of the control valves. In each of the three states, different connections are open between various components. The energy transferred between each of these components is calculated using the air power through a cross-section equation and is limited according to equilibrium. To simplify calculations, all components connected to the brake cylinder outlet of the control valve have a constant combined volume of 40 percent the size of the auxiliary reservoir (i.e., the standard design ratio). Since these downstream components are also only a short distance from the control valve, air pressure values can be sent directly from the control valve with no need for the pressure wave modeling used in the train brake pipe.

Depending on the characteristics of each railcar, some number of empty-load sensors and brake pistons are connected to the brake cylinder outlet of the control valve. Each empty-load sensor has a specified proportioning ratio, empty-load changeover mass evaluated against specific axles on the railcar, and minimum activation pressure. Each brake piston has a bore, a constant pressure loss, and an efficiency. These are used to convert the applied air pressure to a force applied to the brake rigging.

### Overall Train Force

The train was modeled as a series of rigidly-connected masses. Each locomotive and railcar had its own mass and was subject to its own resisting and braking forces. The couplers were modeled as rigid links between each railcar mass, so that all of the locomotives and railcars behaved as one rigid body and had to move together at the same speed and acceleration. Thus, to obtain the net acceleration on the train, the net force was computed as the sum of all forces listed above, and then Newton's second law was applied. Importantly, to reduce errors near zero speed, the forces that always resist motion were summed as resistances and were separate from pure forces. The following equations summarize these terms:

$$\Sigma \text{ force} = \Sigma_{\text{railcars}}(\Sigma_{\text{axles}}(\text{tractive effort})) \quad (1)$$

$$\sum \text{resistance} = \sum_{\text{railcars}} (\sum_{\text{axles}} (\text{bearing resistance} + \text{rolling resistance} + \text{dynamic braking} + \text{air braking}) + \text{aerodynamic resistance} + \text{grade resistance} + \text{curve resistance}) \quad (2)$$

To properly apply Newton's second law, an adjustment must be made to the total mass of the system to capture the rotational inertia of the axles. Once the base acceleration was computed using  $a = \frac{\sum \text{force} + \sum \text{resistance}}{m}$ , the velocity after one time step was checked to see if it switched sign. If so, an adjusted acceleration was computed that ensured the resistances always opposed the direction of motion. This adjusted acceleration was finally used to compute the new velocity and position for each railcar by using the standard kinematic equations and assuming acceleration was constant for the time step.

### **In-Train Forces**

Even though the train was modeled as a rigid body, the forces on each coupler could be computed by sequentially applying Newton's second law to each railcar, as was done for the entire train. These computations were necessary to ensure the train would not split in two from excessive force.

In the future, the model can be further improved by modeling in-train forces in more detail, including simulation of the draft gear and coupling systems.

### **3.1.2 Network**

The track network contained all turnouts, track segments, and track blocks. Turnouts form the nodes in the network while track segments formed the links and were directional. Turnouts, while included in the network, are currently not being used as, for the purposes of this research, all following trains were assumed to traverse the same path through the network. Track segments stored information including elevations, headings, and track block entry points. Track blocks were non-overlapping portions of track segments that stored authorities and referenced all other track blocks that lockout each other's usage. These blocks served as reference points for authorities under moving-block operations and should not be confused with traditional fixed, wayside signal blocks.

### **Paths**

To simplify calculations, each train had a planned set of track segments it traversed as it moved through the network. This set plan was transformed into the path for a train by condensing all characteristics defined by the track segments into what was effectively one large segment combining all segments that the train would traverse. The path also stored a copy of all authorities granted to the train at the entry point to each block with authority granted. This path structure was useful for performing calculations because multiple track segments do not have to be considered when finding a specific offset and because all properties are sorted and have offsets relative to the train origin rather than multiple points (i.e., the start of each successive track block).

## **Authorities**

Each track block contained the set of all authorities currently granted within it. Each authority contained the start offset and end offset relative to the start of the track block, the train index, the direction, and the sequence index. The direction was used to define how the authority could grow and shrink. The sequence index differentiates between multiple authorities for one train that are necessary if, for example, the train is moving around a loop track. Because these authorities are used directly to prevent collisions, all trains must always have an authority for all track that they are occupying.

### **3.1.3 Simulation Model Loop**

The outer network loop, the middle controller loop, and the inner train loop together form the overall simulation model loop. All three loops were synchronized in time but operated with different time steps such that the inner (train) loop ran an integer number of times per middle (controller) loop and the middle (controller) loop ran an integer number of times per outer (network) loop.

#### **Outer Network Loop**

The outer network loop synchronized all trains within the network. First, the “used” authority for each train as it moves forward was removed up to the current position of the back of that train. Next, the authority for each train was extended as far as possible along the path that this train would travel (i.e., end of train ahead). Lastly, the middle controller loop was called to run the controller and update the state of each train to the next network time step.

#### **Middle Controller Loop**

The middle controller loop is responsible for running the controller, sending commands to the train, and calling the inner train loop until reaching the next controller time step. The following five steps were performed in sequence to accomplish this.

1. Run high-level control law to determine the desired acceleration.
2. Run low-level controller to convert desired acceleration into notch setting.
3. Run speed controller to reduce notch setting if near speed limit.
4. Run coupler force controller to shift notch toward zero to limit coupler force.
5. Send final notch setting to all control groups as the current command.

#### **Inner Train Loop**

The inner train loop is responsible for the discrete time simulation of each train. When the inner loop is called, the first step is to run the train controller and send updated commands to the train if applicable. Afterwards, the following eight steps were executed in a loop until the total time elapsed reaches the next network time step.

1. Command sent from future commands to appropriate locomotives.
2. Locomotive power and dynamic braking state update
3. Train brake state update

4. Locomotive independent brake state update
5. Axle force calculation
6. Railcar force calculation
7. Railcar acceleration, position, speed, and coupler forces update
8. Update current time and future command sequence.

These eight steps can also be viewed as a summary of Section 3.1.1.

## **3.2 Baseline Simulation**

To test the multi-train-following simulation and illustrate the behavior of a naïve train control algorithm, baseline simulations were conducted. The detailed results of one particular baseline simulation scenario is presented here as an illustrative example.

### **3.2.1 Baseline Scenario**

The baseline network had two track segments, one in each direction, to represent a 60-mile-long, straight and level hypothetical rail corridor. The baseline network used one track block for the entire corridor. The maximum authorized speed was set to 60 mph so as to be above the balancing speed for the selected train consist, avoiding the need for the speed controller.

Four heavy-haul freight trains at 10,000-foot spacing were started from rest at one end of the 60-mile segment. Each train consisted of two ES44AC locomotives followed by 100 loaded, covered, 286,000-pound gross rail load grain hoppers. This consist corresponded to an overall train length of 6,151 feet and an overall gross weight of 14,732 tons. Each train had 8,800 of available horsepower, yielding a power-to-weight ratio of 0.6 horsepower per ton (hp/ton).

In the simulation, the network time step was set to 4 seconds, the controller time step was also set to 4 seconds, and the train time step was set to 0.1 second.

The first train in the four-train fleet was pre-programmed with a fixed control plan. This plan specified that the lead train use throttle notch 6 between mileposts 0 and 20, notch 4 between mileposts 20 and 40, and notch 6 between mileposts 40 and 60. A PTC enforcement, consisting of a full-service brake application plus the current level of dynamic braking, was used to stop the lead train prior to the end of the 60-mile segment.

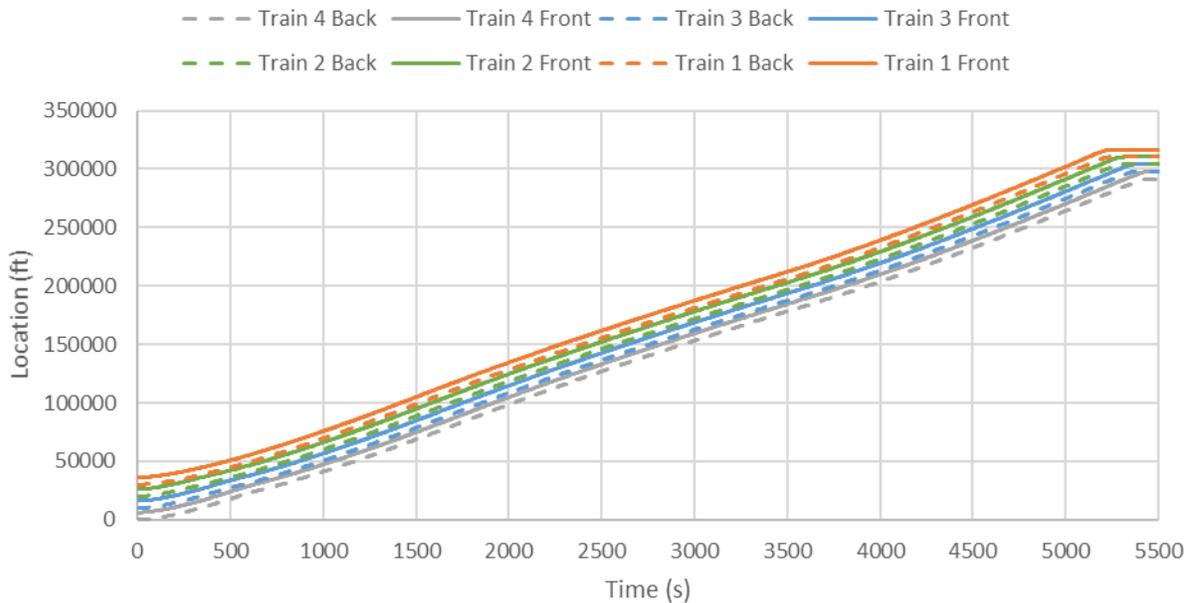
The three following trains were all controlled by the same naïve algorithm; the train was set to full power throttle (notch 8) unless it was within 1,000 feet of the start of a PTC enforced brake application, in which case it was put into full dynamic braking (notch -8). In the case of this train-following simulation, the trains continued at full throttle until they were only separated from the preceding train by the safe braking distance (calculated at each time step based on individual train speed and characteristics) plus a 1,000-foot buffer. Once the train encroached on the train ahead to be separated by less than the combined braking distance and 1,000-foot buffer, the dynamic brake was applied to slow the train. After applying the dynamic brake, the combination of decreased train braking distance as decreased speed relative to the preceding train would eventually increase the train separation to the point where the following train could resume power throttle.

This naïve control algorithm was designed to emulate the potential behavior of a train crew that has little information about the action of the preceding train other than the position of its last car (corresponding to the end of movement authority for the following train) and the corresponding PTC enforcement distance. As the following train closes on the enforcement distance to the train ahead, the crew will likely brake to avoid an enforcement. When the crew observes that the distance to the preceding train is increasing, they will likely increase throttle to bring the train back up to its balancing or maximum authorized speed.

### 3.2.2 Baseline Simulation Results

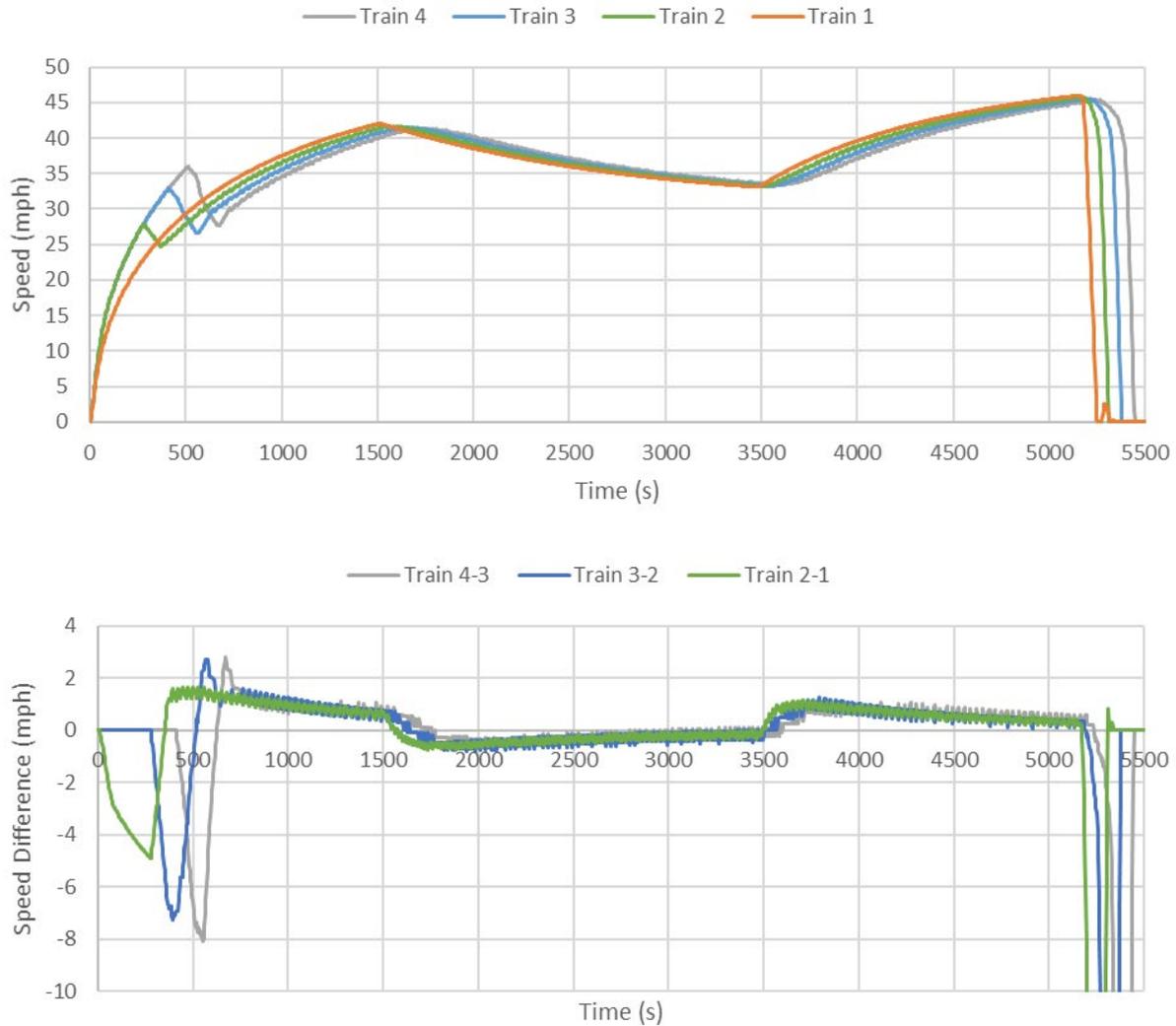
The baseline simulation run required approximately 2 minutes of computer time to complete, with most of the running time spent on computing PTC braking distances. This processing time is required because the PTC braking distance is currently computed by copying the train state, making a full-service brake application, and simulating that condition until the train stops. A simpler and faster PTC braking distance calculator was developed later in the process to alleviate this computation bottleneck.

To evaluate the effectiveness of the control algorithm, several properties of each train were output at each network time step. Time and distance data facilitated creation of a string-line diagram showing the spatial progression of all four trains in time (Figure 2).



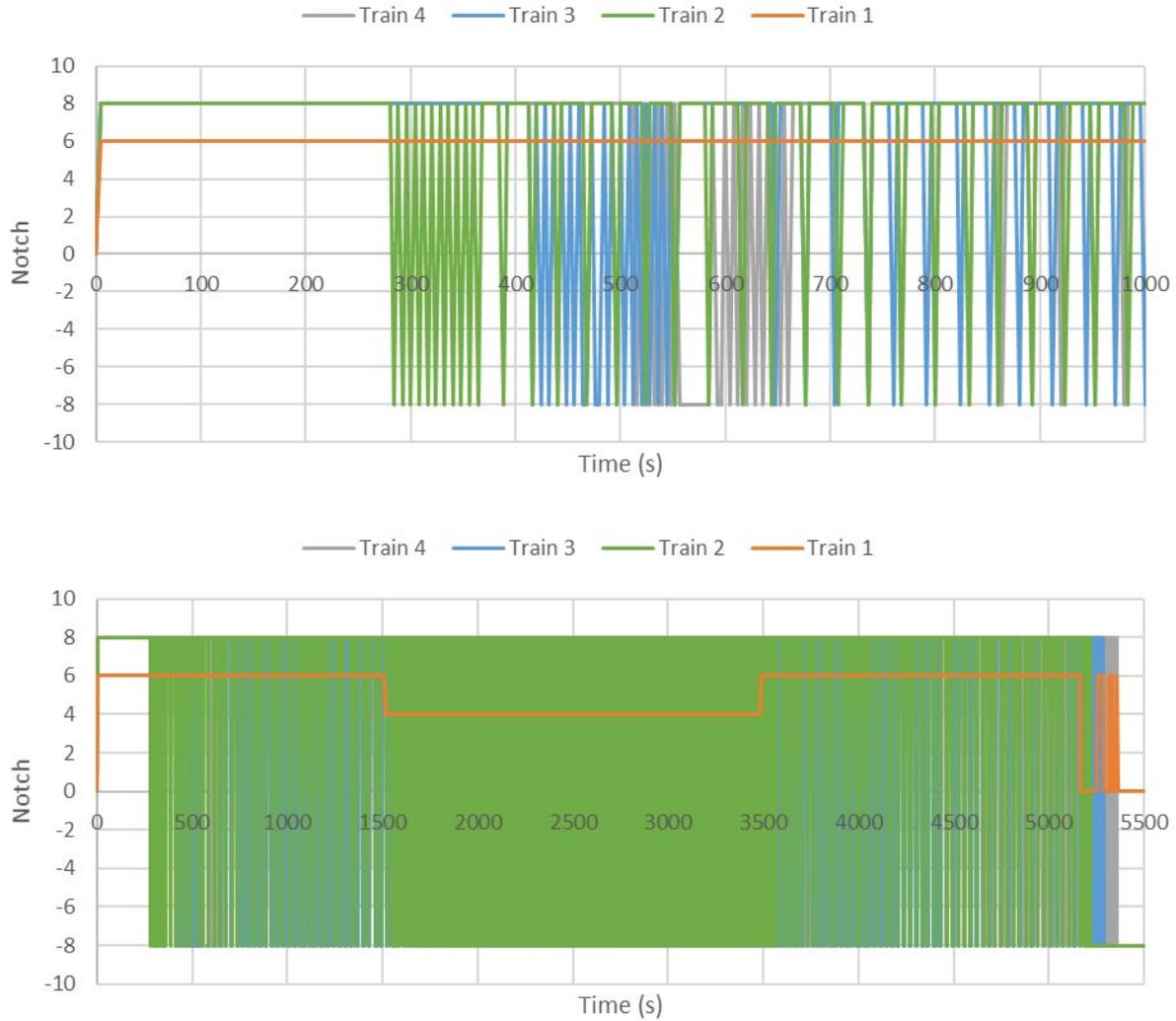
**Figure 2. Stringline Diagram for Four-Train Baseline Simulation**

All headways between trains appeared to be low and consistent, indicating good performance. However, the scale of the string-line diagram was quite coarse, possibly obscuring many important features. Thus, the absolute speed of each train and the relative speed between each pair of trains in sequence were also plotted (Figure 3).



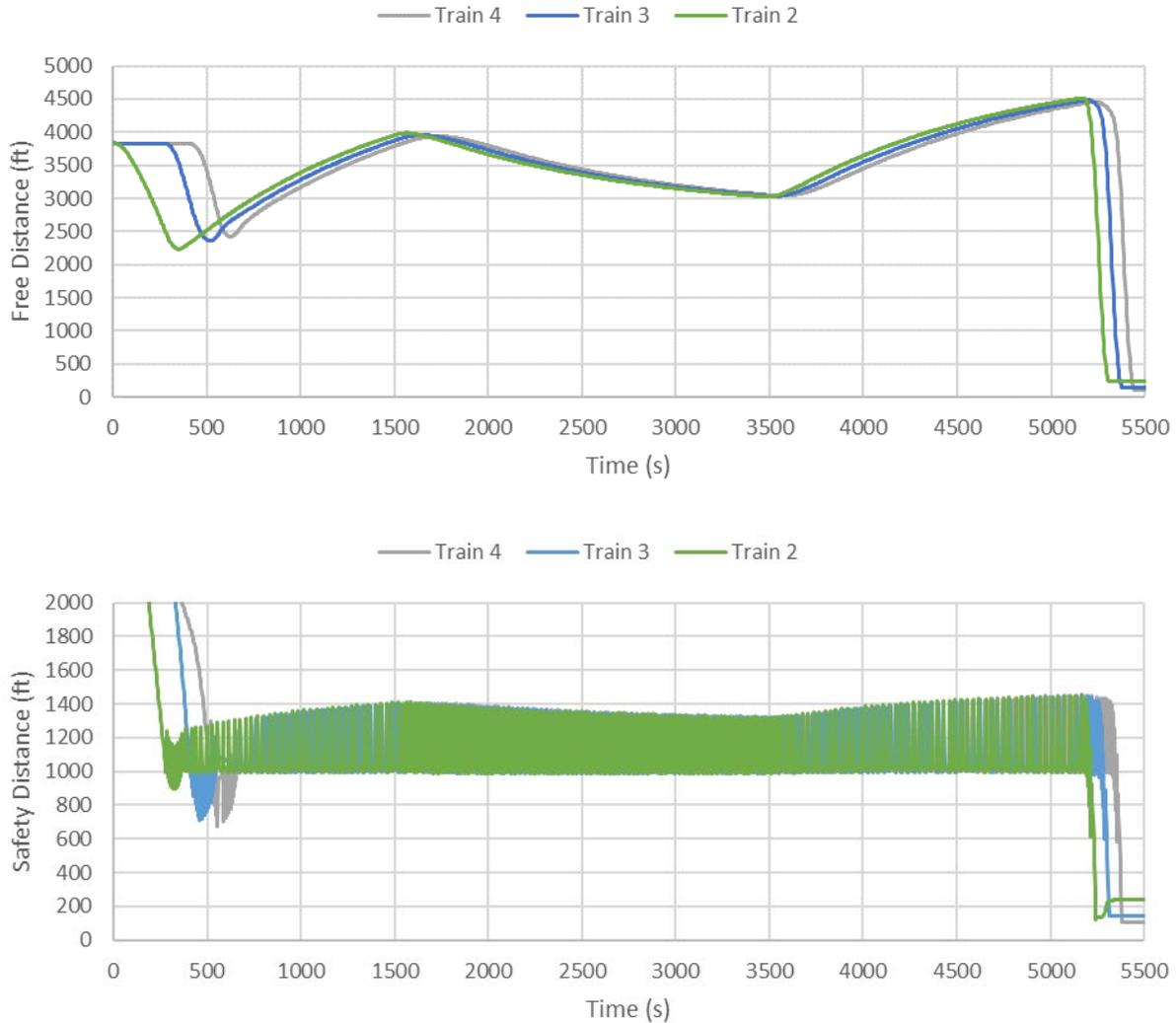
**Figure 3. Absolute and Relative Train Speed over Time for Baseline Simulation**

The results now looked much worse for the controller. At the start of the simulation, there were progressively increasing speed differences between each pair of following trains. These increasing speed differences may indicate that the controller was string unstable. Additionally, there were clear high-frequency oscillations in the speed difference graph all throughout the rest of the simulation. This sort of behavior is not good for minimizing energy consumption. To determine the source of this oscillation, the combined throttle and dynamic braking notch for each train were graphed versus time (Figure 4).



**Figure 4. Throttle Notch Position for First 1,000 Seconds and Full Baseline Simulation**

As expected, considering the design of the naïve train controller, the notch setting switched rapidly between power throttle 8 and dynamic brake -8. This throttle and brake behavior approximated a pulse-width modulation (PWM) signal evaluated every 4 seconds, that varied based on the distance between and relative speed of the following and preceding train. This signal was the source of the speed difference oscillations seen in Figure 3. Since these throttle inputs were generated by the control algorithm, there may also have been some signature of the oscillations in the inputs to the control algorithm. The main input to the control algorithm was the distance remaining before PTC enforcement for each train, also referred to as the safety distance (Figure 5). Additionally, the entire train spacing, including the PTC braking distance (free distance), is plotted over time for comparison.



**Figure 5. Train Spacing and Distance to PTC Enforcement for Baseline Simulation**

Substantial oscillations quite like those shown by the controller were clearly visible in the safety distance metric for each train. The graph of free distance shows only very minor oscillations, indicating that most of the safety distance oscillations were due to changes in the PTC braking distance. The rapid changes arose because the PTC braking distance was calculated by applying a full-service brake application on top of the current level of dynamic braking. When a train switched from notch 8 to notch -8 and began to slow down, the PTC braking distance decreased because there was more available braking effort. This strongly amplified the small change in train spacing caused by slowing down, thus giving the high-frequency and relatively high-amplitude oscillations shown above in the safety distance plot. It also accounted for at least some of the high-frequency oscillation in the notch output from the controller.

The plot of safety distance in [Figure 5](#) also illustrates the difficulty of maintaining a minimum safe braking distance headway under moving blocks. The naïve control algorithm took actions that attempted to maintain an additional 1,000 feet of train separation above the minimum safe braking distance. Due to the time required to implement brake actions and achieve the required

deceleration, following trains are often separated by less than 1,000 feet of safety distance (in addition to the safe braking distance), as the fluctuating relative speed changes braking distance.

Near the start and end of the simulation, subsequent trains followed at even closer distances. The behavior between 300 and 600 seconds – as each successive train begins to encounter the controlling effect of its respective preceding train – was particularly illustrative of the need for an additional safety buffer. Each successive train encountered larger speed differentials and less safety distance as the control actions of preceding trains were amplified through the train fleet. Had the trains attempted to follow at a minimum safe braking distance without any additional safety distance, subsequent trains would have encountered PTC enforcements, since the dynamic brakes were not sufficient to provide the required deceleration to match the speed of the preceding train. The need for additional safety distance to allow for these control actions and relative train speeds increased the overall headway required under moving blocks. These increased headways reduced the potential capacity benefits of moving blocks under this naïve train control algorithm.

### **3.2.3 Baseline Simulation Conclusions**

There is clearly room for improved train control algorithms that offer better performance than the naïve algorithm. Although the naïve algorithm successfully managed train headways to typically follow a 1,000-foot safety distance beyond minimum train braking distance, this was achieved by consistent rapid oscillations between full throttle and full dynamic braking. These train control inputs were not fuel-efficient and may place excessive strain on the mechanical and electrical components of diesel-electric locomotives. Although not modeled in the simulation in detail, the rapid throttle and brake inputs may also create resonant in-train forces that pose a train separation or derailment hazard. The lack of control over air braking and the substantial high-frequency oscillations generated in the system also create situations where following trains amplify the behavior of preceding trains and begin to incur further and further into the safety distance. With a sufficient number of following trains, trains may quickly encounter a PTC enforcement because the naïve train control algorithm cannot command sufficient decelerating force in a timely manner.

This concept of “actuator delay” is central to light-duty highway vehicle and heavy-truck platooning control algorithms. Thus, as described in Section 4, the project team adapted these highway approaches to following vehicle control to the heavy-haul freight rail domain and formulated various high-level control frameworks. Section 5 implements these improved control algorithms into the multi-train-following simulation to compare how they performed relative to the baseline for various combinations of train and route characteristics and leading train behavior. Implementation of the various high-level controller algorithm designs also required development of a low-level controller to transform the acceleration (or deceleration) specified by the analytical equations of the high-level controller into specific throttle and brake commands (including dynamic brakes and/or train air brakes). Both the high- and low-level controllers were implemented in the C++ framework to provide a consolidated multi-train-following simulation model.

## 4. Following Train Control Algorithm Development

---

Based on the literature review, multiple following train control algorithms were formulated and analyzed for their stability. A total of six control algorithms were fully formulated for investigation and evaluation with the multiple-train-following simulation model:

- Naïve Controller
- Proportional Derivative (PD) Controller
- Modified Proportional Derivative (MPD) Controller
- Cooperative Adaptive Cruise Control (CACC) Controller
- Predictive Cooperative Adaptive Cruise Control (PCACC) Controller
- Kinematic Adaptive Cruise Control (KACC) Controller

The naïve controller described in Section 3.2.1 represents the baseline condition of a train crew attempting to manually manage the relative speed and distance to the train ahead based solely on its last reported position. The naïve controller formulation will not be repeated here. The other five controllers were formulated to make more intelligent use of position, speed, acceleration, and other information on the status of the train ahead when specifying the throttle or brake setting of a following train. The following sections describe the general control objective and then present the analytical formulation and stability analysis of the remaining five train-following control algorithms.

### 4.1 Following Train Control Problem and Objectives

The dynamics of train  $i$  in a fleet of  $n$  trains can be expressed as follows:

$$\begin{cases} \dot{x}_i(t) = v_i(t) \\ M_i \dot{v}_i(t) = F_i(t) - f_b(v_i) - f_e(v_i, x_i, t) \end{cases} \quad (3)$$

where  $x_i(t)$ ,  $v_i(t)$ , and  $M_i$  are the position, speed, and mass of train  $i$  at time  $t$ .  $F_i$  is the force imposed (traction or braking force),  $f_b(v_i)$  is the specific basic and aerodynamic resistances, and  $f_e(v_i, x_i, t)$  is a combination of external resistance forces caused by track gradient and curve alignment (Gao et al., 2016). Generally, the control input to be designed contains two parts: an inner-loop part responsible for the compensation of the nonlinear basic and aerodynamic resistances and external disturbances, and an outer-loop in charge of inner-separation control among trains. From a technical point of view, the target is to design cooperative control laws for multiple trains such that the following are achieved:

- (i) For the following trains ( $i = 2, \dots, n$  in the first equation), based on the safety principle of moving-block operations, the target is to maintain the position of train  $i$  with a desired separation from the preceding train, the minimum value of which should be the sum of the train length, braking distance, redundant safety allowance, and positioning error distance.
- (ii) Stability for each train is guaranteed in the sense that the control instructions in the closed-loop system for each train are kept bounded and do not exceed the maximum acceleration or braking performance capabilities of the train.

(iii) String stability of the train fleet (platoon) is guaranteed.

From the energy and economic perspectives, it may also be desired to minimize train energy consumption through throttle and brake control instructions. However, this objective may not be entirely in line with the three objectives above.

#### 4.1.1 Stability and Control Architecture

The train-following control law is said to provide individual train stability if the *space gap error* of trains converges to zero when the preceding train is operating at a constant speed (Rajamani, 2011). The *space gap* ( $\varepsilon$ ) in this definition means the distance between the back of the preceding train and front of the control train. Also, *space gap error* ( $\delta$ ) refers to the difference between space gap ( $\varepsilon$ ) and the desired space gap ( $S_d$ ).

Consider a string of trains on the railway using a longitudinal control system for train-following. Let  $x_i$  be the location of the front of the  $i$ th train measured from an inertial reference. Consider train  $i$  as the one we want to control. Its preceding train is train  $i - 1$ . Also, the length of the preceding train is  $L_{i-1}$ . The space gap between the two trains is thus  $\varepsilon_i = x_{i-1} - x_i - L_{i-1}$ . Note that both  $\varepsilon_i$  and  $L_{des}$  are measured between the back of the preceding train and front of the following one. Then the space gap error relative to the desired space gap ( $S_{d_i}$ ) for the  $i$ th train is  $\delta_i = S_{d_i} - \varepsilon_i = S_{d_i} + x_i - x_{i-1} + L_{i-1}$ . The control law is said to provide individual train stability if the following condition is satisfied:

$$\ddot{x}_{i-1} \rightarrow 0 \implies \delta_i \rightarrow 0 \quad (4)$$

If the train-following control law ensures individual train stability, the space gap error should converge to zero when the preceding train tends to move at a constant speed. However, the space gap error is expected to be non-zero during the acceleration or deceleration of the preceding train. It is important then to describe how the space gap error would propagate from one train to the next train in a string of trains that use the same space gap policy and control law (Rajamani, 2011). The string stability of a string of trains refers to a property in which space gap errors are guaranteed not to amplify as the space errors propagate toward the tail of the string (for a detailed explanation, see Feng et al., 2019). To envision a causal understanding, string stability ensures that any error in space gap between the second and third trains does not amplify into a larger space gap error between the seventh and eighth trains in the string.

There are multiple architectures to shape the information transmission between following trains. In the setup of train-following under a moving-block signaling system with a wayside unit, it may be possible to make information on the location, speed, and acceleration of all trains available to all other trains. We begin with the simplest architecture, in which the control train has just received the location and speed of its preceding train. For that, the input variables are the inter-train spacing, the speed of the train under control, and the speed of the preceding train.

Within this architecture, designing the controller for train-following is much like the one for ACC systems. An ACC system is an extension of the standard cruise control system where the driver sets a constant desired vehicle moving speed. The cruise control system then automatically controls the throttle to maintain the desired speed. An ACC-equipped train uses the space gap information (i.e., the space between the back of the preceding train and front of the following train) on a track. In the absence of a preceding train, an ACC train travels at a user-set speed

under a *speed control* using throttles and brakes, much like a train with a standard cruise control system. However, if a preceding train is detected, the ACC system determines whether the train can continue to travel safely at a pre-specified speed. If the preceding train is too close to the following train or moving too slowly, then the ACC system of the following train switches from speed control to *spacing control*, to maintain the minimum safety spacing from the preceding train. Since train fleeting is under consideration here, discussions focus on spacing control.

A train-following control system architecture is typically designed hierarchically, with an upper-level controller and a lower-level controller. The upper-level controller determines the *desired acceleration* for each train. The lower-level controller determines the *throttle and/or brake commands* required to track the desired acceleration. The train dynamic model is used by the lower-level controller to calculate real-time brake and throttle inputs to track the desired acceleration. Since it is at the core of this research, this section focuses on the upper-level control design.

The objective of the upper-level controller is to determine desired train acceleration such that two performance requirements are met: individual stability and string stability. *Individual stability* aims to asymptotically achieve and maintain a desired spacing from the preceding train. String stability in our context aims to ensure that, when many trains with the same controller move in a fleet on a rail corridor following the same control law, the disturbances in the state (e.g., speed and acceleration) of the leading train of a string will attenuate as one moves to later trains in the string (Swaroop, 1995). As far as the upper-level controller is concerned, the *plant model* (i.e., the train dynamics model) used for the control design is:

$$\ddot{x}_i = u \tag{5}$$

In Eq. (5), subscript  $i$  denotes the  $i$ th train in the string. The acceleration of the train,  $u$ , is the control input. This acceleration should be realized by using lower-level control. An important variable for lower-level control is the *bandwidth of the control system*, which determines how fast the train responds to changes in the input command. The bandwidth of the control loop determines how quickly the system responds to changes in the variables being controlled (e.g., dynamic brake and throttle notches). Due to the finite bandwidth associated with the lower-level controller, some actuation delay is expected. Thus, a train will track its desired acceleration only imperfectly and have deviations from the set output in the upper-level controller.

The objective of the upper-level controller design is stated as the objective of meeting performance specifications robustly in the presence of a first-order actuation delay,  $T$ , from the lower-level controller performance, which will be further explained in Section 4.2.1:

$$\ddot{x}_i = \frac{1}{Ts + 1} \ddot{x}_{i_{des}} = \frac{1}{Ts + 1} u_i \tag{6}$$

Here,  $\ddot{x}_i$  is the actual acceleration of the controlled train,  $\ddot{x}_{i_{des}}$  is the desired acceleration of the train,  $T$  is the actuation delay in implementing the control law, and  $s$  is a complex variable.

The performance requirements (i.e., individual and string stability) must be met under the actual plant model given by Eq. (6). For example, one can assume an actuation delay of  $T = 4$  seconds for the analysis and simulation. The maximum possible acceleration and deceleration must be pre-specified. Since in this part the team has not considered any communication delay between

trains, there is no need for any communication delay parameter related to the input. In this part, the control inputs (i.e., train position) are assumed to transmit instantaneously.

## 4.2 Proportional Derivative Controller

Desired space gap ( $S_{d_i}$ ) can be characterized either as a constant value or a function of speed. It can be shown that considering a constant  $S_{d_i}$  would yield control laws which are not string stable (Rajamani, 2011). For that, one can consider a desired space gap policy that may ensure both individual train stability and string stability, named the constant time-gap (CTG) policy. In the CTG policy, the desired inter-train spacing is not constant but varies linearly with speed:

$$S_{d_i} = h\dot{x}_i + S_{0_i} \quad (7)$$

where  $S_{0_i}$  is the safety distance (space gap at initial time) and the parameter  $h$  is referred to as the constant-time-gap. Eq. (7) means that the inter-train spacing is the safety distance plus the distance traveled by the control train at speed  $\dot{x}_i$  for time  $h$ . Based on  $S_{d_i}$ , the space gap error  $\delta_i$  varies with the speed and is defined as

$$\delta_i = S_{d_i} - \varepsilon_i = h\dot{x}_i + S_{0_i} - \varepsilon_i \quad (8)$$

where  $\varepsilon_i$  is the space gap between the rear of the preceding train and the front of the control train:  $\varepsilon_i = x_{i-1} - x_i - L_{i-1}$ .

The controller based on the CTG policy was developed by Ioannou and Chien (1993). The CTG policy can be represented by the following control law:

$$\ddot{x}_{i_{des}} = -\frac{1}{h}(\dot{\varepsilon}_i + \lambda\delta_i) \quad (9)$$

where  $\ddot{x}_{i_{des}}$  is desired acceleration of controlled train,  $h$  is the headway between the controlled train and its preceding train,  $\dot{\varepsilon}_i$  is the first derivative of  $\varepsilon_i$ ,  $\delta_i$  is the space gap error and  $\lambda$  is a design parameter. Factoring out the input parameters for control law gives the following function representation of the law:

$$PD(h, \tau_s, s_0) \quad (10)$$

where  $h$  is headway,  $\tau_s = \frac{1}{\lambda}$  is the spacing weight, and  $s_0 = S_{0_i}$  is the global safety distance.

This control law includes proportional and derivative terms of the space gap error in addition to the headway between two successive trains. With this control law, it can be shown that the space gap errors of successive train  $\delta_i$  and  $\delta_{i-1}$  are independent of each other (Rajamani, 2011). This suggests that in a string of more than two trains, we do not need the space gap error of the preceding train to calculate the desired acceleration of the following train, i.e., the control law is autonomous (Rajamani, 2011). Under this control law, we can show the string stability of the system.

### 4.2.1 String Stability Analysis of PD Controller

In the presence of the lower controller and actuator dynamics, the desired acceleration is not obtained instantaneously but instead satisfies the dynamics approximated by Eq. (6):

$$T\ddot{x}_i + \dot{x}_i = \dot{x}_{ides} \quad (11)$$

Substituting for  $\dot{x}_{ides}$  from Eq. (7), we obtain

$$T\ddot{x}_i + \dot{x}_i = -\frac{1}{h}(\dot{\varepsilon}_i + \lambda\delta_i) \quad (12)$$

Also, differentiating  $\delta_i$  twice from Eq. (8), we obtain

$$\ddot{\delta}_i = \ddot{\varepsilon}_i + h\ddot{x}_i \quad (13)$$

Substituting for  $\ddot{x}_i$  from Eq. (11), we find that the relation between  $\varepsilon_i$  and  $\delta_i$  is given by

$$\ddot{\varepsilon}_i = \ddot{\delta}_i + \frac{1}{T}(\dot{\varepsilon}_i + h\dot{x}_i + \lambda\delta_i) \quad (14)$$

or

$$\ddot{\varepsilon}_i = \ddot{\delta}_i + \frac{1}{T}(\dot{\delta}_i + \lambda\delta_i) \quad (15)$$

The difference between errors of successive trains can be written as

$$\delta_i - \delta_{i-1} = \varepsilon_i - \varepsilon_{i-1} + h(\dot{x}_i - \dot{x}_{i-1}) \quad (16)$$

or

$$\delta_i - \delta_{i-1} = \varepsilon_i - \varepsilon_{i-1} + h\dot{\varepsilon}_i \quad (17)$$

Using Eq. (16) to substitute in Eq. (17) to calculate  $\varepsilon_i$  in terms of  $\delta_i$  and  $\varepsilon_{i-1}$  in terms of  $\delta_{i-1}$ , dynamic relation between  $\delta_i$  and  $\delta_{i-1}$  can be obtained. In the transfer function domain, this relation is

$$H(s) = \frac{\delta_i}{\delta_{i-1}} = \frac{s + \lambda}{hTs^3 + hs^2 + (1 + \lambda h)s + \lambda} \quad (18)$$

The string stability of this system can be analyzed by looking at the above transfer function and checking if its magnitude is always less than 1. Substituting  $j\omega$  for  $s$  and evaluating the magnitude of the above transfer function, it is shown in Swaroop & Hedrick (1995) that the magnitude is always less than or equal to unity at all values for  $s$  only if:

$$h \geq 2T \quad (19)$$

Further, if Eq. (19) is satisfied, then it is guaranteed that one can find a value of  $\lambda$  such that  $\|\widehat{H}(s)\|_{\infty} \leq 1$ . Thus, the condition of Eq. (19) is both necessary and sufficient with an appropriate value of  $\lambda$  (Swaroop & Hedrick, 1995). In practice, string stability can be maintained only if the time-gap is larger than  $2T$ .

#### 4.2.2 Pseudocode of PD Controller

The pseudocode for the PD controller is presented in Figure 6.

---

Calculate  $T$  from the below equation numerically

$$T\ddot{x}_i + \dot{x}_i = \dot{x}_{i_{des}}$$

For  $i = 2:n$  use below definitions:

$x_i$ =location of train  $i$

$\dot{x}_i$ =speed of train  $i$

$L_{i-1}$ =length of train  $i-1$  (unchanging)

$S_{0_i}$ =prespecified unchanging constant safety spacing

$h$ =prespecified constant time gap (we can change it,  $h > 2T$ )

$\lambda$ =prespecified design parameter (we can change it)

$\dot{x}_{i_{des}}$ =desired acceleration (control input)

Read  $x_i, x_{i-1}, \dot{x}_i, \dot{x}_{i-1}$

Set  $\ddot{x}_{i_{des}} = -\frac{1}{h}(\dot{x}_{i-1} - \dot{x}_i + \lambda(h\dot{x}_i + S_{0_i} + L_{i-1} + x_i - x_{i-1}))$

End

**Figure 6. PD Controller Pseudocode**

#### 4.3 Modified Proportional Derivative Controller

The main drawback of the PD controller is that the controller formulation has headway and speed weighting terms as the same. Consequently, we cannot change headway without changing the speed weight. Moreover, when the train ahead is going faster, the PD controller will instruct the following train to go faster, which may not be desirable for the following train movement.

The MPD controller is proposed to address these two issues and also solve the issues about PTC enforcement laws. In this controller, the spacing is defined as:

$$s = x_{i_t} - s_0 - h\dot{x}_i - x_{i_s} \quad (20)$$

where  $x_{i_t}$  is the location of the back of train  $i-1$  interpolated over the network update interval,  $x_{i_s}$  is the stop position for train  $i$ , and  $s_0$  is the stopped safety distance.

By modifying the PD controller as described above, the MPD controller formulation is:

$$\ddot{x}_{i_{des}} = \begin{cases} \frac{1}{\tau_v}(\dot{x}_{i-1} - \dot{x}_i) + \frac{1}{\tau_v\tau_s}s, \dot{x}_{i-1} < \dot{x}_i \\ \frac{1}{\tau_v\tau_s}s, \dot{x}_{i-1} \geq \dot{x}_i \end{cases} \quad (21)$$

The function representation of this controller is as follows:

$$MPD(\tau_v, \tau_s, h, s_0) \quad (22)$$

In this controller, the headway  $h$  and the spacing weight  $\tau_v$  are separate variables. As with the PD controller,  $\tau_s$  is the spacing weight and  $s_0$  is the minimum safety distance. With these modifications to the PD controller, the issue of PTC enforcements during acceleration is solved without affecting the string stability of the controller.

#### 4.4 Cooperative Adaptive Cruise Control Controller

CACC is a natural extension to ACC based on obtaining other vehicle information via wireless communication. We first consider a constant time-gap policy in our CACC controller for cooperative train-following. To improve train platooning safety, one should ensure that the CACC controller design and corresponding parameter specification for a platoon of trains is string stable. Consider the following model of train  $i$  within a platoon of  $n$  trains:

$$\begin{pmatrix} \dot{d}_i \\ \dot{v}_i \\ \dot{a}_i \end{pmatrix} = \begin{pmatrix} v_{i-1} - v_i \\ a_i \\ -\frac{1}{\tau}a_i + \frac{1}{\tau}u_i \end{pmatrix}, i \in S_n \setminus \{1\} \quad (23)$$

Here,  $d_i = x_{i-1} - x_i - L_{i-1}$  is the distance between train  $i$  and train  $i - 1$ , where  $x_i$  and  $x_{i-1}$  are the position of the front of trains  $i$  and  $i - 1$ , respectively, and  $L_{i-1}$  is the length of train  $i - 1$ .  $v_i$  is the speed and  $a_i$  is the acceleration of train  $i$ . Moreover,  $u_i$  is the train input, which can be interpreted as the actual acceleration for train 1 (based on the platoon maneuver) and as the desired acceleration for the following trains.  $\tau$  is the time constant representing the locomotive traction drive dynamics, including actuation lag. Also, the following constant headway policy is adopted for the spacing:

$$d_{d,i}(t) = r + hv_{i-1}(t), i \in S_n \setminus \{1\} \quad (24)$$

where  $d_{d,i}(t)$  is the desired distance between train  $i$  and  $i - 1$ ,  $h$  is the time headway (to be determined), and  $r$  is the standstill distance. The main objective is to regulate the distance  $d_i$  to  $d_{d,i}(t)$ , i.e.,

$$e_i(t) = d_i(t) - d_{d,i}(t) \rightarrow 0 \text{ as } t \rightarrow \infty \quad (25)$$

where  $e_i(t)$  is the spacing error. This equation may only be satisfied if the leading train moves with a constant speed, i.e.,  $a_1 = 0$ . In Ploeg et al. (2011), it is shown that the following dynamic controller achieves this train-following objective:

$$\dot{u}_i = -\frac{1}{h}u_i + \frac{1}{h}(k_p e_i + k_d \dot{e}_i + k_{dd} \ddot{e}_i) + \frac{1}{h}u_{i-1} \quad (26)$$

where  $k_p$ ,  $k_d$ , and  $k_{dd}$  are the controller coefficients. The function representation of this controller is as follows:

$$CACC(\tau_v, \tau_s, k_{dd}, h, s_0) \quad (27)$$

where  $\tau_v$  is the speed weight,  $\tau_s$  is the spacing weight,  $h$  is the headway,  $s_0 = r$  is the minimum spacing,  $k_p = \frac{1}{\tau_v \tau_s}$ , and  $k_d = \frac{1}{\tau_v}$ . Ploeg et al. (2011) showed that for a bounded  $u_{i-1}$ , the spacing error reached zero for  $a_1 = 0$  if the following constraints held:  $k_p, k_d > 0, k_{dd} + 1 > 0$ , and  $(1 + k_{dd})k_d - k_p \tau > 0$ .

The transfer functions in the block diagram (Figure 7) for this controller are as follows:

$$G(s) = \frac{X_i(s)}{U_i(s)} = \frac{1}{s^2(\tau s + 1)} \quad (28)$$

$$H(s) = hs + 1 \quad (29)$$

$$K(s) = k_p + k_d s + k_{dd} s^2 \quad (30)$$

$$D(s) = e^{-\theta s} \quad (31)$$

Here,  $X_i(s)$  and  $U_i(s)$  are the Laplace transforms of the train position  $x_i(t)$  and the desired acceleration  $u_i(t)$  respectively. The train transfer function  $G(s)$  follows the form  $\ddot{x}_i = -\frac{1}{\tau}\dot{x}_i + \frac{1}{\tau}u_i$ , as in Eq. (23). The spacing policy transfer function  $H(s)$  related to Eq. (24) and the controller  $K(s)$  represents the error feedback in Eq. (26). Also,  $\theta$  is the time delay induced by the wireless communication network. In order to consider the actuation delay in our formulation, we need to add another term to the transfer function  $G(s)$ . Hence, we modify the transfer function into:

$$G(s) = \frac{1}{s^2(\tau s + 1)} e^{\phi s} \quad (32)$$

where  $\phi$  is the train actuation delay.

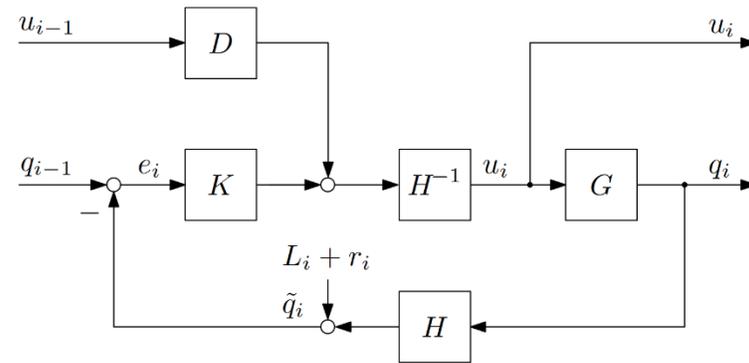


Figure 7. Block Diagram of CACC Controller

#### 4.4.1 String Stability Analysis of the CACC Controller

Let the train acceleration be taken as a basis for string stability, i.e.,  $y_i(t) = a_i(t) \forall i \in S_n$ . The corresponding string stability measure is given by:

$$\Gamma_{CACC}(s) = \frac{A_i(s)}{A_{i-1}(s)} = \frac{1}{H(s)} \frac{G(s)K(s) + D(s)}{1 + G(s)K(s)} \quad (33)$$

where  $A_i(s)$  and  $A_{i-1}(s)$  are the Laplace transforms of  $a_i(t)$  and  $a_{i-1}(t)$ , respectively. Without loss of generality, we may assume  $r_i = L_{i-1} = 0, \forall i \in S_n \setminus \{1\}$ . The corresponding string stability criterion is then:

$$|\Gamma_{CACC}(j\omega)|_{H_\infty} \leq 1 \quad (34)$$

We can numerically show that with some parameters  $k_p, k_d, k_{dd}, \tau, \theta, \phi, h$ , the requirement of Eq. (34) would be satisfied.

#### 4.4.2 Pseudocode of CACC Controller

The pseudocode for the CACC controller is presented in [Figure 8](#).

#### 4.5 Predictive Cooperative Adaptive Cruise Control Controller

Based on the original CACC algorithm, a new modified CACC controller following a constant spacing policy was proposed by Sybis et al. (2019) to improve performance of CACC systems in highway vehicles. In this approach, the instantaneous acceleration values were replaced by the desired ones in the controller formulation. According to their published results, this modification over instantaneous controller (which does not have predictive terms) had a significant impact on the performance of platoons of highway vehicles.

---

For  $i = 2:n$  use below definitions:

$x_i$ =front location of train  $i$

$\dot{x}_i$ =speed of train  $i$

$\ddot{x}_i$ =acceleration of train  $i$

$L_{i-1}$ =length of train  $i-1$  (unchanging)

$b_i$ =braking distance

$r$ =some extra prespecified safety distance

$h$ =prespecified constant time gap (we can change it)

$k_p, k_d, k_{dd}$ =prespecified design parameter (we can change it)

$a_{ptc}, t_b$ , and  $t_r$ =prespecified PTC enforcement safety distance parameters ( $t_b \approx 35 \text{ sec}, t_r \approx 10 \text{ sec}, a_{ptc} \approx 0.3 \text{ m/s}^2$ )

$u_i$ =desired acceleration (control input)

---

Read  $x_i, x_{i-1}, \dot{x}_i, \dot{x}_{i-1}, \ddot{x}_i, \ddot{x}_{i-1}, u_1$

Set

$$d_i = x_{i-1} - x_i - L_{i-1},$$

$$\text{If } \dot{x}_i \leq \frac{a_{ptc} t_b}{2},$$

$$b_i(\dot{x}_i) = \frac{2}{3} \dot{x}_i \sqrt{\frac{2t_b \dot{x}_i}{a_{ptc}}} + \dot{x}_i t_r$$

$$\frac{d}{dt} b_i = \dot{x}_i^{\frac{1}{2}} * \ddot{x}_i * \left(\frac{2t_b}{a_{ptc}}\right)^{\frac{1}{2}} + \ddot{x}_i * t_r$$

$$\frac{d^2}{dt^2} b_i = \frac{1}{2} \dot{x}_i^{-\frac{1}{2}} * \ddot{x}_i^2 * \left(\frac{2t_b}{a_{ptc}}\right)^{\frac{1}{2}} + \ddot{x}_i * \dot{x}_i^{\frac{1}{2}} * \left(\frac{2t_b}{a_{ptc}}\right)^{\frac{1}{2}} + \ddot{x}_i * t_r$$

Else,

$$b_i(\dot{x}_i) = \frac{\dot{x}_i^2}{2a_{ptc}} + \dot{x}_i \left(t_r + \frac{t_b}{2}\right) - \frac{a_{ptc} t_b^2}{24}$$

$$\frac{d}{dt} b_i = \frac{\dot{x}_i \times \ddot{x}_i}{a_{ptc}} + \ddot{x}_i \left(t_r + \frac{t_b}{2}\right)$$

$$\frac{d^2}{dt^2} b_i = \frac{\ddot{x}_i \times \dot{x}_i + \ddot{x}_i^2}{a_{ptc}} + \ddot{x}_i \left(t_r + \frac{t_b}{2}\right)$$

$$d_{d,i} = r + b_i(\dot{x}_i),$$

$$e_i = d_i - d_{d,i},$$

$$\dot{u}_i = -\frac{1}{h} u_i + \frac{1}{h} (k_p e_i + k_d \dot{e}_i + k_{dd} \ddot{e}_i) + \frac{1}{h} u_{i-1}.$$

End

### Figure 8. CACC Controller Pseudocode

In this formulation,  $x_{i_t}$  is the target position for train  $i$  and  $x_{i_s}$  is the stop position for train  $i$ . Under the constant spacing policy, the desired spacing between successive trains is defined as  $r_{des}$ . The formula for PCACC is:

$$\begin{aligned} \ddot{x}_{i_{des}} = & (1 - C_1) \ddot{x}_{(i-1)_{des}} + C_1 \ddot{x}_{i_{des}} - \left(2\xi - C_1 \left(\xi + \sqrt{\xi^2 - 1}\right)\right) \omega_n \dot{\delta}_i \\ & - \left(\xi + \sqrt{\xi^2 - 1}\right) \omega_n C_1 (\dot{x}_i - \dot{x}_l) - \omega_n^2 \delta_i \end{aligned} \quad (35)$$

where  $\delta_i$  is the spacing error of the  $i$ -th train and defined as:  $\delta_i = x_{i_t}(t) - x_{i_s}(t) - r_{des}$ .

Consequently, the first derivation of spacing error is  $\dot{\delta}_i = \dot{x}_i - \dot{x}_{i-1}$ .

In this formulation, the subjects of tuning are the constants  $C_1$ ,  $\xi$  and  $\omega_n$ . Gain  $C_1$  is the weight of the preceding and leading train accelerations and speeds. Gain  $\xi$  is the damping ratio and can be set to one for critical damping, i.e., the threshold between overdamping and underdamping in which the system returns to the equilibrium position as quickly as possible, passing it at most once without oscillating. Gain  $\omega_n$  is the bandwidth of the controller. The function representation of this controller is as follows:

$$PCACC(t_r, C_1, \xi, s_0) \quad (36)$$

where  $t_r = \frac{1}{\omega_n}$  is the resolution time and  $s_0$  is the minimum spacing.

Here, the modifications over instantaneous CACC controller are replacements of  $\ddot{x}_{i-1}$  and  $\ddot{x}_i$  with  $\ddot{x}_{(i-1)des}$  and  $\ddot{x}_{ides}$ , respectively.

Although the introduced modification changes the instantaneous CACC controller formula only slightly, its impact on the platoon performance is meaningful. This is because the predictive formula considers not only the current time but also how dynamics of the platoon should evolve in the close future. In the instantaneous CACC controller, transmitted acceleration values express the state of the train at the time of transmitting the message (neglecting sensor delay and processing time), while in the predictive version of CACC each train disseminates the information on its desired acceleration values. As a result, significantly improved information is broadcast as compared with information on the current acceleration value. Thus, the following train can update its behavior and adapt in advance to that what will presumably occur in the close future. In this scenario, it has been assumed that the acceleration value transmitted to the following trains will be achieved in the near future. However, such an assumption does not always hold, due to the residual error in tracking the acceleration command by the lower-level controller. This issue can be regarded as another degree of freedom in the simulation scenario.

#### 4.5.1 String Stability Analysis of the PCACC Controller

To analyze the string stability of the PCACC control law, recall Eq. (35) and formulate a similar equation for train  $i - 1$ . The subtraction of the latter one side-by-side from Eq. (35) lead us to the formula for the desired acceleration difference between the  $i$ -th and  $(i - 1)$ -th trains:

$$\begin{aligned}\ddot{\delta}_{ides} &= \ddot{x}_{ides} - \ddot{x}_{(i-1)des} \\ &= (1 - C_1)\ddot{\delta}_{(i-1)des} - \left(2\xi - C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\right)\omega_n(\dot{\delta} - \dot{\delta}_{i-1}) \\ &\quad - \left(\xi + \sqrt{\xi^2 - 1}\right)\omega_n C_1 \dot{\delta}_i - \omega_n^2(\delta_i - \delta_{i-1})\end{aligned}\quad (37)$$

Assuming that the train engine reacts to the desired acceleration signal as a first order inertial system, the lag is modeled as a low-pass filter applied to the output of the control law:

$$P(s) = \frac{A_i(s)}{A_{ides}(s)} = \frac{1}{\tau s + 1}\quad (38)$$

where  $A_i(s) = \mathcal{L}[a_i]$ ,  $A_{ides}(s) = \mathcal{L}[a_{ides}]$  and  $\tau$  is the actuation lag. Based on Eq. (38), we can write the following equation:

$$\begin{aligned}\ddot{\delta}_i + \tau\ddot{\delta}_i &= (1 - C_1)(\ddot{\delta}_{i-1} + \tau\ddot{\delta}_{i-1})\left(2\xi - C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\right)\omega_n(\dot{\delta} - \dot{\delta}_{i-1}) \\ &\quad - \left(\xi + \sqrt{\xi^2 - 1}\right)\omega_n C_1 \dot{\delta}_i - \omega_n^2(\delta_i - \delta_{i-1})\end{aligned}\quad (39)$$

According to Sybis et al. (2019), the calculation of the Laplace transform of Eq. (39) leads to the following equation reflecting dependency  $E_i(s) = \mathcal{L}[\delta_i]$  on  $E_{i-1}(s) = \mathcal{L}[\delta_{i-1}]$

$$E_i(s) = E_{i-1}(s)H(s) + I(s) \quad (40)$$

where

$$H(s) = \frac{(1 - C_1)(\tau s^3 + s^2) + (2\xi - C_1(\xi + \sqrt{\xi^2 - 1}))\omega_n s + \omega_n^2}{\tau s^3 + s^2 + 2\xi\omega_n s + \omega_n^2} \quad (41)$$

and

$$I(s) = \delta(0) \frac{\tau s^2 + s + 2\xi\omega_n}{\tau s^3 + s^2 + 2\xi\omega_n s + \omega_n^2} - \delta_{i-1}(0) \times \frac{(1 - C_1)(\tau s^2 + s) + (2\xi - C_1(\xi + \sqrt{\xi^2 - 1}))\omega_n}{\tau s^3 + s^2 + 2\xi\omega_n s + \omega_n^2} \quad (42)$$

The above calculations have been performed assuming that in a steady state of the train platoon for  $t = 0$  the derivatives of the distance errors are zero.

The train platoon is string stable if:

$$|H(j\omega)| \leq 1 \quad (43)$$

Finding the ranges of parameters  $\xi$ ,  $\omega_n$ ,  $\tau$ , and  $C_1$  for which Eq. (43) holds is not straightforward, as the numerator and denominator in Eq. (41) are third-order polynomials, and setting requirement Eq. (43) would lead to a complicated inequality in which all the parameters appear concurrently. Instead, we can use numerical and graphical solutions. Based on Sybis et al. (2019) calculations, the platoon is string stable if the actuation lag  $\tau$  does not exceed certain limits. For example, with parameters  $\xi = 1$ ,  $C_1 = 0.5$ , and  $\omega_n = 0.5$ , the platoon is string stable if  $\tau < 1.5$  sec.

#### 4.6 Kinematic Adaptive Cruise Control (KACC) Controller

Previous controllers had no guarantees of safety, and gain parameters depended at least partially on the specific safe braking distance formulation. Ideally, the same parameters would work independently of the particular function  $b_i$  to ensure that the formulation and parameters will work with the real PTC braking distance function.

A system of equations was developed to solve this problem based on the following assumptions: If trains have identical performance, safety is guaranteed so long as the train behind can match the speed of the train in front within the current spacing between them. This can be converted into a desired acceleration. The train accelerates from its current position and velocity at the desired acceleration for some specified look ahead time and then immediately decelerates at the maximum rate to reach the desired velocity after covering the desired spacing. The following equations formulate this idea mathematically:

$$\begin{cases} s_1 = v_0 t_l + \frac{a_{des}}{2} t_l^2 \\ v_f^2 - v_m^2 = 2a_{min}(s_f - s_1) \\ v_m = v_0 + a_{des} t_l \end{cases} \quad (44)$$

where  $t_l$  is the look ahead time,  $s_f$  is the full spacing,  $v_0$  is the starting velocity of the following train,  $v_f$  is the target velocity for the following train,  $v_m$  is the max velocity reached,  $a_{min}$  is the maximum deceleration, and  $a_{des}$  is the desired acceleration. After testing, the following equations were used to define these parameters.

$$s_f = x_{i_t} - x_i - s_0 - h\dot{x}_i \quad (45)$$

$$s_e = \min(s_f, x_{i_t} - x_{i_s} - s_0) \quad (46)$$

$$t_l = \max\left(t_{l_{min}}, \frac{s_f - s_e}{\dot{x}_i}\right) \quad (47)$$

Here,  $x_{i_t}$  is the target position for train  $i$  (i.e., the location of the back of train  $i - 1$  interpolated over the network update interval),  $s_0$  is the stopped safety distance, and  $x_{i_s}$  is the stop position for train  $i$ , which is the location of the front of train  $i$  plus its braking distance:  $x_{i_s} = x_i + b_i(\dot{x}_i)$ . Additionally,  $s_f$  is the full spacing,  $s_e$  is the excess spacing,  $t_l$  is the look ahead time, and  $t_{l_{min}}$  is the minimum look ahead time. To calculate the desired acceleration, the intermediate term  $\ddot{x}_{i_p}$  is calculated as follows:

$$\ddot{x}_{i_p} = \begin{cases} \frac{\ddot{x}_{i_{min}}}{2} - \frac{1}{t_l} \left( \dot{x}_i - \sqrt{\dot{x}_{i-1}^2 + \ddot{x}_{i_{min}} \left( t_l \dot{x}_i + \frac{t_l^2 \ddot{x}_{i_{min}}}{4} - 2s_f \right)} \right), s_f > t_l \frac{\dot{x}_i + \dot{x}_{i-1}}{2} \\ \frac{\dot{x}_{i-1}^2 + \dot{x}_i^2}{2s_f}, s_f \leq t_l \frac{\dot{x}_i + \dot{x}_{i-1}}{2} \end{cases} \quad (48)$$

To reach the final desired acceleration value, a spacing-based acceleration limit derived from the MPD controller is applied for when the following train going slower than the preceding train, as follows:

$$\ddot{x}_{i_{des}} = \min\left(\ddot{x}_{i_p}, \frac{s_e}{\tau_s^2}\right) \quad (49)$$

where  $\tau_s$  is spacing weight. The function representation of this controller is as follows:

$$KACC(h, \tau_s, t_{l_{min}}, s_0) \quad (50)$$

where  $h$  is the headway,  $\tau_s$  is the spacing weight,  $t_{l_{min}}$  is the minimum look ahead time, and  $s_0$  is the minimum spacing.

With this controller, we have the same parameters for all trains. Moreover, parameters are independent of braking distance function  $b_i$ , which ensures that the formulation and parameters will work with actual PTC braking distance functions (not known exactly to the research team) in practice. In fact, a virtual coupling scenario (PTC braking distance of zero) was tested to validate and no collisions or PTC enforcements occurred.

#### 4.6.1 String Stability Analysis of the KACC Controller

Taking the train acceleration as a basis for string stability, the corresponding string stability measure is given by:

$$\Gamma_{KACC}(s) = \frac{A_i(s)}{A_{i-1}(s)} \quad (51)$$

where  $A_i(s)$  and  $A_{i-1}(s)$  are the Laplace transforms of  $a_i(t)$  and  $a_{i-1}(t)$ , respectively. The corresponding string stability criteria is then:

$$|\Gamma_{KACC}(j\omega)|_{H_\infty} \leq 1 \quad (52)$$

Similar to CACC and PCACC, we may find numerically the parameters for  $h, s_0, \tau_s$  which satisfy the requirement of Eq. (52).

#### 4.6.2 Pseudocode of KACC Controller

The pseudocode for the KACC controller is presented in [Figure 9](#).

---

```

KACC( $h, \tau_s, t_{lmin}, s_0$ )
 $s_f = x_{i_t} - x_i - s_0 - h\dot{x}_i$ 
 $s_e = \min(s_f, x_{i_t} - x_{i_s} - s_0)$ 
 $t_l = \max\left(t_{lmin}, \frac{s_f - s_e}{\dot{x}_i}\right)$ 
If  $s_f > t_l \frac{\dot{x}_i + \dot{x}_{i-1}}{2}$ :
     $\ddot{x}_{i_p} = \frac{\ddot{x}_{i_{min}}}{2} - \frac{1}{t_l} \left( \dot{x}_i - \sqrt{\dot{x}_{i-1}^2 + \ddot{x}_{i_{min}} \left( t_l \dot{x}_i + \frac{t_l^2 \ddot{x}_{i_{min}}}{4} - 2s_f \right)} \right)$ 
Else:
     $\ddot{x}_{i_p} = \frac{\dot{x}_{i-1}^2 + \dot{x}_i^2}{2s_f}$ 
 $\ddot{x}_{i_{des}} = \min\left(\ddot{x}_{i_p}, \frac{s_e}{\tau_s^2}\right)$ 
 $t_l$ =look-ahead time
 $s_f$ =full spacing
 $s_e$ =excess spacing
 $h$ =headway
 $s_0$ =minimum spacing
 $t_s$ =controller time step

```

---

---

$x_{i_t}$ =target offset for the following train  
 $x_{i_s}$ =offset of the start of PTC enforcement for the following  
 $\ddot{x}_{i_{min}}$  and  $\ddot{x}_{i_{max}}$  are computed based on the current state of  
train  $i$   
 $x_{i_t}$  and  $\dot{x}_{i-1}$  are smoothed over the network time step based on  
current and previous values

---

**Figure 9. KACC Controller Pseudocode**

## 5. Simulated Performance of Following Train Control Algorithms

---

To effectively evaluate each proposed control algorithm, a battery of tests was developed based on a factorial design. This was accomplished by testing values for scenario parameters which seemed most likely to produce undesirable results from the control algorithms and then iteratively determining which of these values consistently resulted in undesirable behavior. By focusing on these values only, the total number of simulations needed to fully evaluate control algorithm behavior was greatly reduced, especially because the set of scenarios to be tested is defined by all possible combinations of the scenario parameters. The following sub-sections detail the baseline setup, the set of scenarios used, the metrics used to evaluate control algorithm performance, and the plan to compare control algorithms. Afterwards, the results of each test plan will be described and analyzed, with each test plan getting its own sub-section.

### 5.1 Experimental Design

Two main parameters defined a scenario. The first of these was the train plan. This parameter fully specified each train within a single simulation. To further simplify the requirements for the control algorithm, train plans were required to have all trains be identical (except for starting position). This ensured that following trains would have enough braking effort to match that of the trains in front so long as they were traveling the same speed. In a real-world scenario, this limitation could be imposed by artificially limiting the deceleration rate of all trains in a platoon to the worst performing one. Three different train plans, comprising a range of train performance, were used to evaluate each controller. These are summarized in [Table 2](#) below.

**Table 2. Summary of Train Plans**

<b>Train Type</b>	<b>Power (hp)</b>	<b>Weight (tons)</b>	<b>Length (ft)</b>	<b>HP/Ton</b>
Loaded Grain	8,800	14,992	6,150	0.59
Manifest	8,800	8,415	4,385	1.05
Intermodal	17,600	9,050	5,599	1.94

The second parameter was the front train profile. This parameter defined the notch setting and train brake setting that the lead train would follow based on its location. Train brakes were not used by the lead train to ensure the following trains can match performance. To make the train wait after stopping, another parameter was used to indicate how long to keep the control setting once the train stops. A total of six front train profiles were used to evaluate each controller. The first set of three profiles had two cycles of control while the second set had eight cycles of control. Within these sets, the first profile had a control cycle of notch 8 followed by notch 0, the second profile had a control cycle of notch 8 followed by notch -8, and the third profile had a control cycle of notch 5 followed by notch -8. Each of these control cycles required 120 seconds of dwell when changing from the notch 0 or notch -8 component and a required dwell of 2,000 seconds at the end of the profile to standardize the total fuel usage across controllers. Each control cycle was spaced approximately evenly over 100 miles.

All scenarios were set up on a 200-mile segment of perfectly straight and level track. This was a long enough segment of track to conduct all the desired tests since the control actions were placed within the first 100 miles. Straight and level track was used to ensure that only dynamic braking would be necessary to control train speed. Note that all trains were equipped with AC

traction locomotives that had effective dynamic braking down to 0.5 mph. Trains started at 10,000-foot, front-to-front spacing with the back of the last train placed at the start of the track segment. Note that this initial spacing was longer than each of the trains to be tested.

Combining the 3 train plans and the 6 front train profiles gives a grand total of 18 simulations per controller and set input values. The values for each of the 18 simulations are summarized in [Table 3](#) below. Additionally, several metrics were developed to summarize controller performance over the set of 18 simulations. These are detailed in the next section.

**Table 3. Summary of Simulation Scenarios**

<b>Scenario</b>	<b>Train Type</b>	<b>Max Notch</b>	<b>Min Notch</b>	<b>Cycle Count</b>
1	Loaded Grain	8	-8	2
2	Loaded Grain	8	-8	8
3	Loaded Grain	8	0	2
4	Loaded Grain	8	0	8
5	Loaded Grain	5	-8	2
6	Loaded Grain	5	-8	8
7	Manifest	8	-8	2
8	Manifest	8	-8	8
9	Manifest	8	0	2
10	Manifest	8	0	8
11	Manifest	5	-8	2
12	Manifest	5	-8	8
13	Intermodal	8	-8	2
14	Intermodal	8	-8	8
15	Intermodal	8	0	2
16	Intermodal	8	0	8
17	Intermodal	5	-8	2
18	Intermodal	5	-8	8

### **5.1.1 Performance Metrics**

Four main metrics were used to compare controller performance and to find the best set of input values for a given controller. This “best set of input values” was based on testing a range of values for the most influential parameters and then balancing the importance of the four metrics. These four metrics are defined below and are presented in order of decreasing importance.

The first of these was whether the simulation had a PTC failure, defined as any PTC enforcement event. Since the original assumptions specified that a good controller could avoid causing any PTC enforcements, the ideal controller and set of input values would have no PTC failures across all simulations tested. Thus, this metric was used to immediately disqualify sets of input values and to filter subsequent results tables.

The second of these metrics was average separation. This was defined by averaging the distance between the front of the train and the start of the PTC braking curve for each following train at each time step. The result of each simulation was averaged with the others to aggregate. The

ideal controller would have a low average separation, corresponding to increased train throughput.

The third of these metrics was fuel consumption. Since the simulator had detailed fuel consumption calculations, this was simply the total amount of fuel consumed by all trains in the specified simulation. To aggregate across simulations, the total fuel consumption was simply averaged. The ideal controller would have a low average fuel consumption to save on fuel costs. However, since moving-block systems will typically be implemented on very dense corridors, average separation was given a higher importance than fuel consumption.

The fourth and final metric was the standard deviation of notch difference. This was defined by taking the standard deviation of the difference between all time-adjacent notch settings for each following train. To aggregate across the 18 simulations, these standard deviation values were simply averaged. This metric will be large when there are large and frequent changes in the notch setting. Thus, the ideal controller would have a low standard deviation of notch difference. Since the standard deviation of notch difference is not a standard metric used to evaluate train performance, it was given the lowest importance.

### **5.1.2 Test Plans**

To efficiently determine which controller has the best performance, a series of four test plans were developed. First, the naïve controller was compared to the PD controller to show that large gains in performance are possible even with a quite simple controller design. Second, the PD, MPD, CACC, and KACC controllers were compared while allowing only position information to be communicated from preceding train to following train. The research team found that the MPD and KACC controllers performed quite similarly to each other and better than the other two. Third, the MPD, KACC, and PCACC controllers were compared while allowing any information to be communicated between any trains. The team found that the MPD and KACC controllers performed similarly to each other and better than the PCACC controller. Fourth, the performance impact of varying controller time step and interpolation methods was determined.

For each test plan, the best set of input values was first derived for each controller to be compared by balancing the importance of the four metrics described above. Next, the values for these metrics were compared across all the controllers to be compared to find which performs the best. Lastly, the behavior of each controller versus time was compared for a selected scenario. The values compared were excess spacing, defined for each following train as the difference between the target offset and the front of the train (like the average separation metric), and the notch setting of each train.

To limit the number of values that needed to be tested while keeping a level playing field, the minimum spacing parameter,  $s_0$ , was set to be at least 2 meters for all controllers. While  $s_0$  should be minimized in general, several controllers had issues when using  $s_0 = 0$  m that did not occur when using  $s_0 = 2$  m.

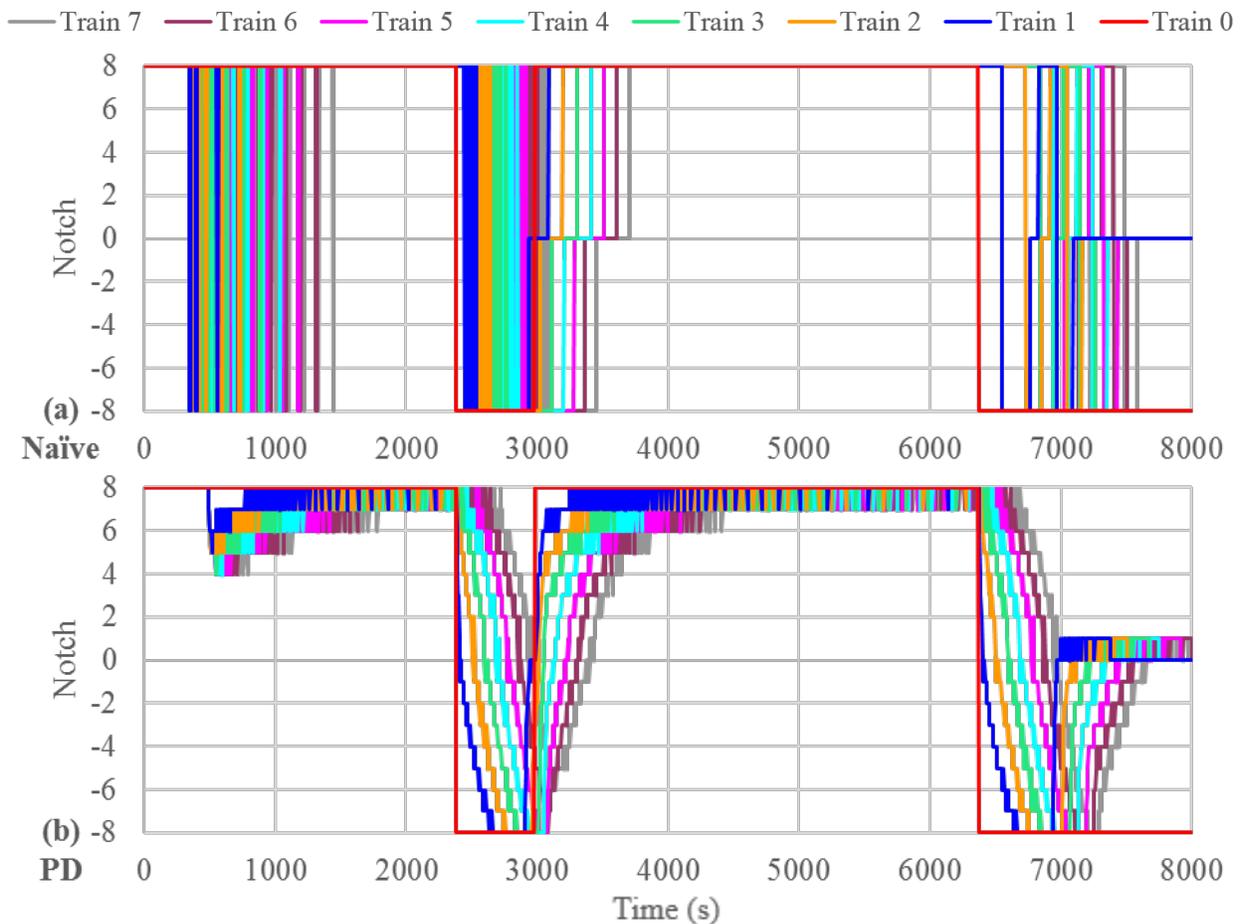
## **5.2 Simulation Experiment Results**

### **5.2.1 Naïve versus Proportional Derivative Controller**

In this test plan, the network time step was set to 4 seconds and the controller time step was also set to 4 seconds, meaning that no interpolation was needed. The only communication path

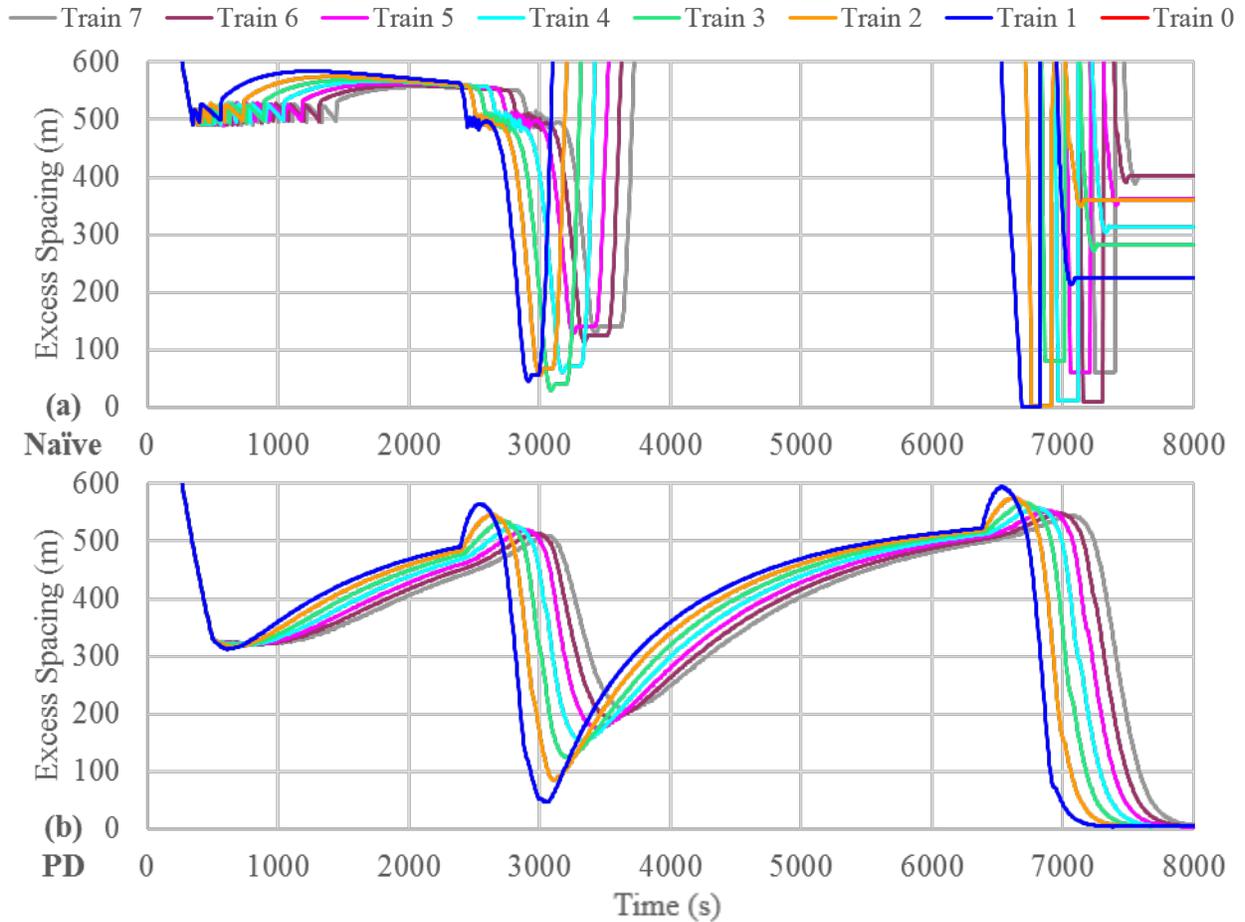
allowed was from preceding train to adjacent following train and only position was communicated. Change in position over one network time step was used to by the following train to derive the approximate velocity of the preceding train. Similarly, change in approximate velocity over one network time step was used as approximate acceleration of the preceding train.

Because the naïve controller was very simple, it was not possible to select a minimum spacing under 1,000 meters that avoids PTC enforcements for all 18 scenarios. This is exemplified in scenario number 1, which had eight loaded grain trains and a front train profile of two notch 8 to notch -8 cycles, because the speed difference between preceding and following trains during a braking event was larger than the specified spacing was capable of handling.  $N(s_0 = 500\text{ m})$  was used as the input for the naïve controller. For comparison, the PD controller with parameter values  $PD(h = 20\text{ sec}, \tau_s = 60\text{ sec}, s_0 = 2\text{ m})$  is shown. This set was the best at balancing the various performance metrics, as will be discussed in the next section, while also not causing any PTC enforcements in any of the 18 scenarios. The notch performance is shown in Figure 10 below and the excess spacing is shown afterwards in Figure 11.



**Figure 10. Naïve Controller versus PD Controller Notch Performance**

The PD controller had superior notch stability, which is quite important to both reduce fuel consumption and manage in-train forces. This was as expected because the naïve controller can only output either full power or full deceleration, which is simply not enough granularity for effective train control.



**Figure 11. Naïve Controller versus PD Controller Excess Spacing**

For nearly the entire simulation run, the excess spacing of the PD controller was lower than that of the naïve controller, which corresponded to lower average separation and increased train throughput. Additionally, PTC enforcements (a PTC failure) occurred at around 6,800 seconds for the naïve controller.

Overall, it was clear the naïve controller had quite poor performance and that even a control algorithm as simple as the PD controller could greatly improve performance. Because of this, the naïve controller was included in any subsequent comparisons.

### 5.2.2 Position-only Comparison

Of all the controllers developed, only the naïve, PD, MPD, CACC, and KACC controllers were designed to operate using only preceding train position information. Like the naïve versus PD controller test plan, the network time step was set to 4 seconds and the controller time step was also set to 4 seconds, meaning that no interpolation was needed. The only communication path allowed was from preceding train to adjacent following train and only position was communicated. Approximate velocity and acceleration were derived as described in the previous test plan. The tables used to determine the best set of input values for each controller are shown below.

For the PD controller, variation of parameters  $h$  and  $\tau_s$  are shown. The last parameter,  $s_0$  was set as  $s_0 = 2$  m. Thus, the set of results shown below in Table 4 were generated from inputs of the form  $PD(h, \tau_s, s_0 = 2 \text{ m})$ . Using the table, the research team determined that the best set of input values for the PD controller was  $PD(h = 20 \text{ sec}, \tau_s = 60 \text{ sec}, s_0 = 2 \text{ m})$ . These results are bolded to highlight them.

**Table 4. Position-Only Test Plan, PD Controller Parameter Variation**

**PTC Failures**

(a)		$\tau_s$ (sec)					
		30	40	50	60	70	80
h (sec)	10	<b>5</b>	<b>6</b>	<b>6</b>	<b>6</b>	<b>6</b>	<b>6</b>
	15	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>2</b>
	20	<b>2</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	25	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	30	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Average Separation (m)**

(b)		$\tau_s$ (sec)					
		30	40	50	60	70	80
h (sec)	10						
	15						
	20			359.5	<b>361.2</b>	363.0	364.8
	25		429.8	431.7	433.6	435.4	437.3
	30		502.2	504.2	506.2	508.1	510.1

**Fuel Used (kg)**

(c)		$\tau_s$ (sec)					
		30	40	50	60	70	80
h (sec)	10						
	15						
	20			10610	<b>10560</b>	10530	10510
	25		10560	10500	10460	10430	10420
	30		10460	10400	10370	10350	10340

**SD Notch Difference**

(d)		$\tau_s$ (sec)					
		30	40	50	60	70	80
h (sec)	10						
	15						
	20			0.5630	<b>0.5460</b>	0.5344	0.5296
	25		0.5541	0.5291	0.5192	0.5144	0.5113
	30		0.5299	0.5137	0.5040	0.5018	0.5000

For the MPD controller, the set of results shown below in Table 5 were generated from inputs of the form  $MPD(\tau_v, \tau_s, h = 0 \text{ sec}, s_0 = 2 \text{ m})$ .  $h = 0 \text{ sec}$  was chosen for testing because lower  $h$  results in lower average separation. Using the table, the team determined that the best set of input values for the MPD controller was  $MPD(\tau_v = 12 \text{ sec}, \tau_s = 90 \text{ sec}, h = 0 \text{ sec}, s_0 = 2 \text{ m})$ , which has its results bolded.

**Table 5. Position-Only Test Plan, MPD Controller Parameter Variation**

<b>PTC Failures</b>						
<b>(a)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8	3	4	3	0	0
	10	2	1	0	0	0
	12	1	2	0	0	0
	14	1	1	0	0	0
	16	2	1	1	0	0

<b>Average Separation (m)</b>						
<b>(b)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8				107.8	111.1
	10			104.8	108.3	111.7
	12			<b>105.2</b>	108.8	112.4
	14			105.7	109.5	113.1
	16				110.2	114.0

<b>Fuel Used (kg)</b>						
<b>(c)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8				11180	11170
	10			11150	11140	11140
	12			<b>11140</b>	11140	11140
	14			11150	11150	11150
	16				11160	11170

<b>SD Notch Difference</b>						
<b>(d)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8				0.6844	0.6553
	10			0.6065	0.5857	0.5739
	12			<b>0.5497</b>	0.5407	0.5304
	14			0.5177	0.5093	0.5017
	16				0.4808	0.4761

For the KACC controller, the set of results shown below in Table 6 were generated from inputs of the form  $KACC(h, \tau_s, t_m = 10 \text{ sec}, s_0 = 2 \text{ m})$ . The value of  $t_m = 10 \text{ sec}$  was chosen because it was a small value that avoided PTC failures for most combinations of  $h$  and  $\tau_s$  values. Using the table, it was determined that the best set of input values for the KACC controller was  $KACC(h = 8 \text{ sec}, \tau_s = 22 \text{ sec}, t_m = 10 \text{ sec}, s_0 = 2 \text{ m})$ , which has its results bolded.

**Table 6. Position-Only Test Plan, KACC Controller Parameter Variation**

(a)		PTC Failures							
		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
h (sec)	7	7	4	2	3	3	3	2	3
	8	6	1	0	0	0	0	0	0
	9	6	3	1	0	1	0	0	0
	10	5	3	2	1	0	0	0	0
	11	6	1	2	1	0	0	0	0

(b)		Average Separation (m)							
		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
h (sec)	7								
	8			106.4	<b>106.8</b>	107.3	107.8	108.4	109.2
	9				107.1		108.2	108.8	109.5
	10					108.0	108.5	109.2	109.9
	11					108.4	109.0	109.6	110.2

(c)		Fuel Used (kg)							
		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
h (sec)	7								
	8			11170	<b>11160</b>	11140	11140	11140	11140
	9				11140		11130	11130	11130
	10					11120	11120	11120	11120
	11					11110	11110	11110	11110

(d)		SD Notch Difference							
		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
h (sec)	7								
	8			0.4709	<b>0.4566</b>	0.4480	0.4416	0.4374	0.4329
	9				0.4594		0.4451	0.4394	0.4377
	10					0.4549	0.4482	0.4426	0.4391
	11					0.4572	0.4523	0.4458	0.4417

For the CACC controller, the set of results shown below in Table 7 were generated from inputs of the form  $CACC(\tau_v, \tau_s, k_{dd} = 0.3, h = 15 \text{ sec}, s_0 = 450 \text{ m})$ . The values for  $k_{dd}$ ,  $h$ , and  $s_0$  were also chosen by testing many different combinations. Note that because there were five total parameters to optimize, these values for  $k_{dd}$ ,  $h$ , and  $s_0$  were very probably not optimal. However, the results were much worse than those for the other controllers in this test plan. Thus, there was not a combination of  $k_{dd}$ ,  $h$ , and  $s_0$  values that could match the performance of the other controllers. Using the table, the team decided that the best set of input values for the CACC controller was  $CACC(\tau_v = 20 \text{ sec}, \tau_s = 110 \text{ sec}, k_{dd} = 0.3, h = 15 \text{ sec}, s_0 = 450 \text{ m})$ , which has its results bolded.

**Table 7. Position-Only Test Plan, CACC Controller Parameter Variation**

**PTC Failures**

(a)		$\tau_s$ (sec)			
		80	90	100	110
$\tau_v$ (sec)	10	2	2	2	2
	15	2	2	2	0
	20	2	1	1	0
	25	2	2	1	1

**Average Separation (m)**

(b)		$\tau_s$ (sec)			
		80	90	100	110
$\tau_v$ (sec)	10				
	15				
	20				
	25				

**Fuel Used (kg)**

(c)		$\tau_s$ (sec)			
		80	90	100	110
$\tau_v$ (sec)	10				
	15				
	20				
	25				

**SD Notch Difference**

(d)		$\tau_s$ (sec)			
		80	90	100	110
$\tau_v$ (sec)	10				
	15				
	20				
	25				

For reference, the best set of input values for each controller are summarized below in [Table 8](#). Additionally, [Table 9](#) summarizes the results for these inputs, facilitating direct comparison between controllers. Overall, the MPD controller and the KACC controller perform the best by far and perform very similarly to each other. The PD controller performed third-best and the CACC controller performed substantially worse than the other three for all metrics.

**Table 8. Position-Only Test Plan, Best Input Values Summary**

$PD(h = 20 \text{ sec}, \tau_s = 60 \text{ sec}, s_0 = 2 \text{ m})$
$MPD(\tau_v = 12 \text{ sec}, \tau_s = 90 \text{ sec}, h = 0 \text{ sec}, s_0 = 2 \text{ m})$
$KACC(h = 8 \text{ sec}, \tau_s = 22 \text{ sec}, t_m = 10 \text{ sec}, s_0 = 2 \text{ m})$
$CACC(\tau_v = 20 \text{ sec}, \tau_s = 110 \text{ sec}, k_{dd} = 0.3, h = 15 \text{ sec}, s_0 = 450 \text{ m})$

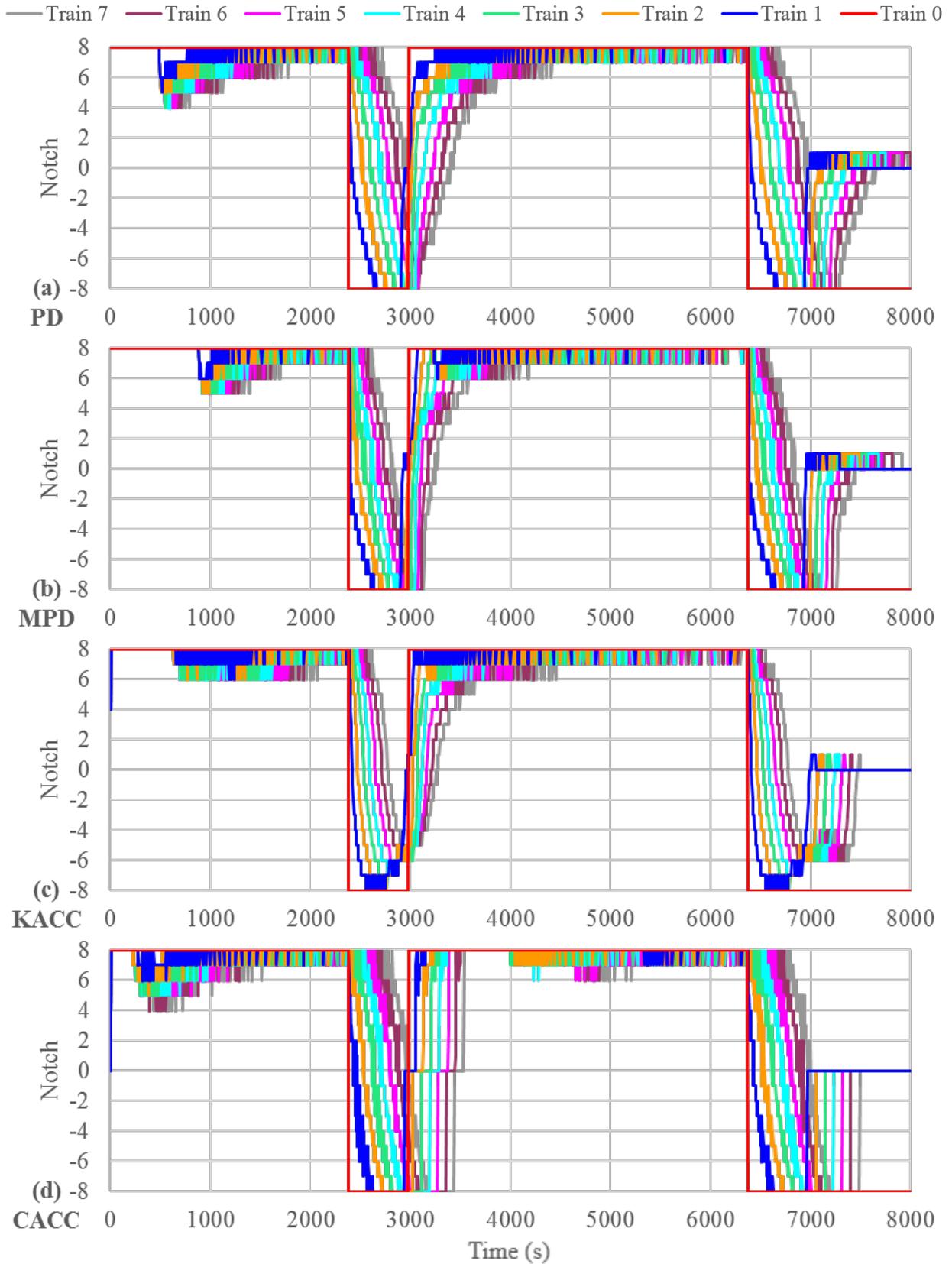
**Table 9. Position-Only Test Plan, Results Summary**

<b>Metric</b>	<b>PD</b>	<b>MPD</b>	<b>KACC</b>	<b>CACC</b>
<b>PTC Failures</b>	0	0	0	0
<b>Average Separation (m)</b>	361.2	105.2	106.8	554.3
<b>Fuel Used (kg)</b>	10560	11140	11160	11300
<b>SD Notch Difference</b>	0.5460	0.5497	0.4566	0.8936

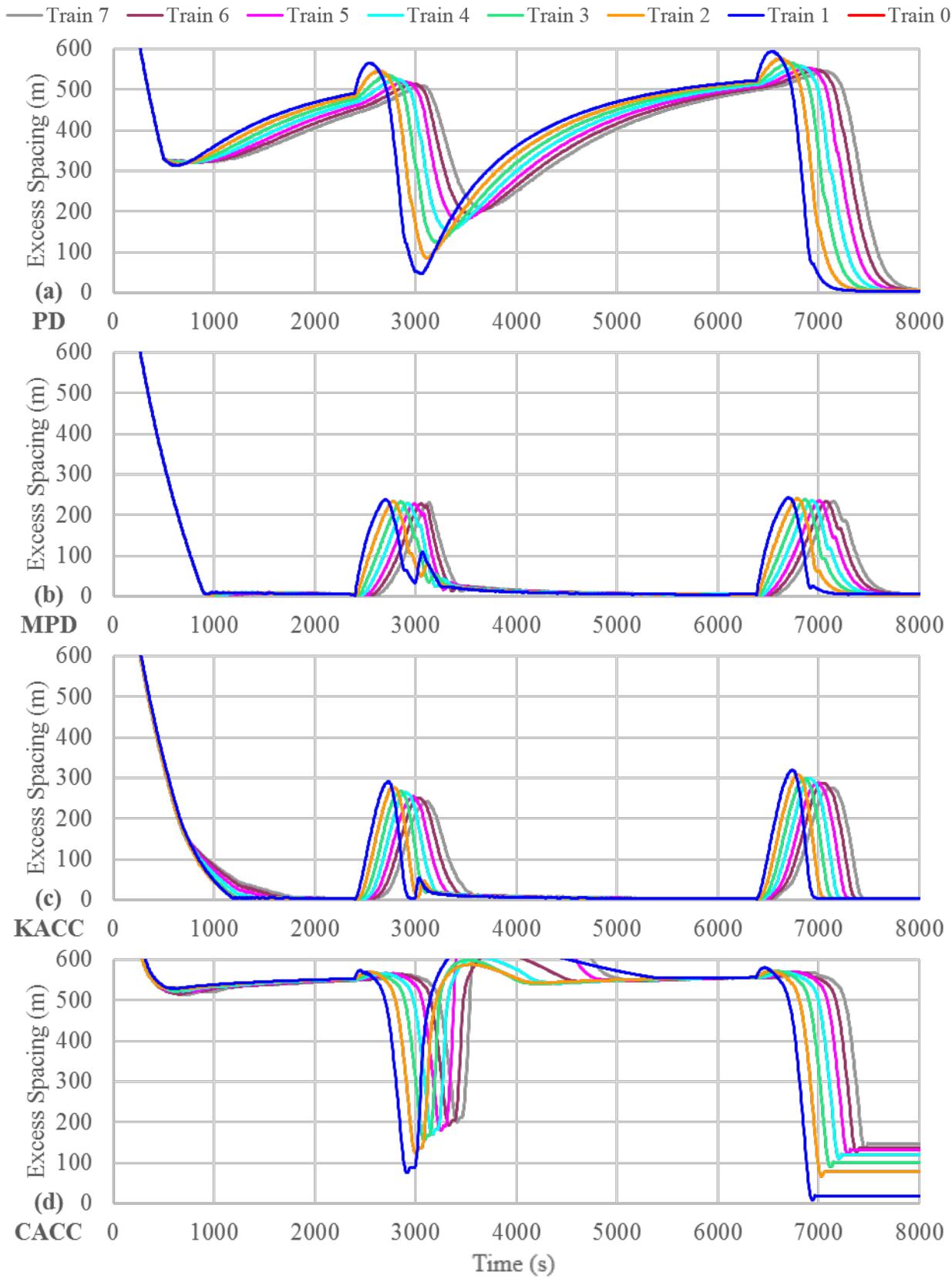
The PD controller had larger average separation and lower fuel consumption compared to MPD and KACC controller performance. However, there was no way to reduce the average spacing of the PD controller to anywhere near that of the MPD or KACC controllers, no matter what input values were used. Additionally, the lower fuel consumption of the PD controller should have been achievable by either the MPD controller or the KACC controller by adding an extra headway, as was done for the PD controller. Thus, the MPD and KACC controllers performed better than the PD controller.

The KACC controller had a substantially lower standard deviation of notch difference and a slightly higher average separation and fuel usage compared to the MPD controller. While the standard deviation of notch difference is not a typical metric used in evaluating train performance, it can be considered a proxy for effective in-train force management. Additionally, for the MPD controller to hit this standard deviation of notch difference, it needs to use the same amount of fuel as the KACC controller and have a slightly larger average separation than the KACC controller has. Thus, there was no clear overall performance advantage between the MPD controller and the KACC controller.

To compare the notch and spacing performance of the controllers versus time, one of the 18 scenarios must be selected. Scenario 1, which had eight loaded grain trains and a front train profile of two cycles of notch 8 to notch -8, did a good job of stressing controllers that lacked a velocity difference term. In fact, this scenario was the reason that  $s_0 = 450 \text{ m}$  had to be used for the CACC controller. [Figure 12](#) and [Figure 13](#) below show the notch and spacing performance respectively for the PD, MPD, KACC, and CACC controllers.



**Figure 12. Position-Only Test Plan, Notch Performance Comparison**



**Figure 13. Position-Only Test Plan, Spacing Excess Comparison**

Of the four controllers, the CACC controller had the worst performance because it had clear notch instability during all deceleration events and some acceleration events and because it required far more spacing at high speed as compared with the other controllers. Between the PD and the MPD controller, the MPD controller more closely matched the notch of the lead train. The MPD controller also had lower excess spacing for the entire simulation. Thus, as with the results from analysis of metrics, the CACC controller had the worst performance, followed by the PD controller.

Comparing the MPD controller and the KACC controller, the MPD controller had lower spacing everywhere except at the end of deceleration events. However, the KACC controller had better reactions during deceleration events in terms of notch performance. Additionally, the KACC controller generally tracked the lead train notch setting more closely. Thus, there was no clear overall performance advantage between the MPD controller and the KACC controller.

Combining the results from all comparisons in this test plan, the MPD and KACC controllers performed similarly to each other and substantially better than either the PD controller or the CACC controller. Therefore, only the MPD and KACC controllers were considered in subsequent test plans.

### **5.2.3 More Communication Comparison**

Once the infrastructure is in place to send authority updates to trains at a 4-second update interval, it should be straightforward to add extra information to each message. This could include current velocity, current acceleration, desired acceleration, train brake state, etc. Additionally, it should be feasible to send information along more paths than just from preceding train to following train.

To determine the benefits of this sort of setup, the MPD and KACC controllers were tested under the scenario where messages contain more information. The PCACC controller was also added to the test plan because this test plan was the first one so far that can support it. This was because the PCACC controller required information from the leading train in the platoon.

To change the minimal number of variables at once, the network time step was again set to 4 seconds and the controller time step was again set to 4 seconds, still meaning that no interpolation was needed. All possible communications paths were allowed, and any variable could be sent along any of these paths. In practice, only current position, velocity, acceleration, and desired acceleration were communicated, and the only new communication paths used were from the leading train to each following train, not quite doubling the original number communication links.

For the MPD controller, the set of results shown below in Table 10 were generated from inputs of the form  $MPD(\tau_v, \tau_s, h = 0 \text{ sec}, s_0 = 2 \text{ m})$ .  $h = 0 \text{ sec}$  was chosen for testing because a lower  $h$  resulted in lower average separation. Using the table, researchers determined that the best set of input values for the MPD controller was  $MPD(\tau_v = 12 \text{ sec}, \tau_s = 90 \text{ sec}, h = 0 \text{ sec}, s_0 = 2 \text{ m})$ , which has its results bolded. This was the same set of input values as for the position only test plan.

**Table 10. More Communication Test Plan, MPD Controller Parameter Variation**

<b>PTC Failures</b>						
<b>(a)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8	<b>4</b>	1	<b>2</b>	0	0
	10	1	1	0	0	0
	12	1	1	<b>0</b>	0	0
	14	1	1	0	0	0
	16	1	1	0	0	0

<b>Average Separation (m)</b>						
<b>(b)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8				108.8	112.2
	10			105.8	109.4	112.9
	12			<b>106.3</b>	110.0	113.6
	14			106.8	110.6	114.3
	16			107.4	111.3	115.2

<b>Fuel Used (kg)</b>						
<b>(c)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8				11150	11140
	10			11120	11100	11100
	12			<b>11100</b>	11100	11100
	14			11110	11100	11110
	16			11120	11120	11120

<b>SD Notch Difference</b>						
<b>(d)</b>		$\tau_s$ (sec)				
		70	80	90	100	110
$\tau_v$ (sec)	8				<b>0.7135</b>	<b>0.6798</b>
	10			0.6224	0.5999	0.5881
	12			<b>0.5600</b>	0.5485	0.5421
	14			0.5233	0.5134	0.5061
	16			0.4923	0.4833	0.4791

For the KACC controller, the set of results shown below in Table 11 were generated from inputs of the form  $KACC(h, \tau_s, t_m = 10 \text{ sec}, s_0 = 2 \text{ m})$ . The value of  $t_m = 10 \text{ sec}$  was the same as in the position only test plan. Using the table, researchers determined that the best set of input values for the KACC controller was  $KACC(h = 8 \text{ sec}, \tau_s = 22 \text{ sec}, t_m = 10 \text{ sec}, s_0 = 2 \text{ m})$ , which has its results bolded. This was the same set of input values as for the position only test plan.

**Table 11. More Communication Test Plan, KACC Controller Parameter Variation**

		<b>PTC Failures</b>							
<b>(a)</b>		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
<b>h (sec)</b>	7	5	4	3	3	3	4	3	4
	8	4	0	0	0	0	0	0	0
	9	4	1	0	0	0	0	0	0
	10	7	1	0	0	0	0	0	0
	11	6	2	0	0	0	0	0	0

		<b>Average Separation (m)</b>							
<b>(b)</b>		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
<b>h (sec)</b>	7								
	8		107.3	107.7	<b>108.1</b>	108.6	109.1	109.7	110.5
	9			108.1	108.5	109.0	109.5	110.2	110.9
	10			108.5	108.8	109.3	110.0	110.6	111.2
	11			108.8	109.2	109.7	110.3	111.0	111.7

		<b>Fuel Used (kg)</b>							
<b>(c)</b>		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
<b>h (sec)</b>	7								
	8		11160	11120	<b>11100</b>	11090	11090	11090	11090
	9			11110	11090	11080	11080	11080	11080
	10			11100	11080	11070	11060	11060	11070
	11			11090	11070	11060	11050	11050	11050

		<b>SD Notch Difference</b>							
<b>(d)</b>		$\tau_s$ (sec)							
		16	18	20	22	24	26	28	30
<b>h (sec)</b>	7								
	8		0.4987	0.4721	<b>0.4586</b>	0.4488	0.4438	0.4383	0.4340
	9			0.4776	0.4621	0.4518	0.4455	0.4408	0.4367
	10			0.4813	0.4638	0.4538	0.4481	0.4431	0.4388
	11			0.4842	0.4662	0.4559	0.4486	0.4436	0.4397

For the PCACC controller, the set of results shown below in Table 12 were generated from inputs of the form  $PCACC(t_r, C_1, \xi = 1, s_0 = 200 \text{ m})$ .  $\xi = 1$  corresponds to critical damping and was found to be the best from other testing.  $s_0 = 200 \text{ m}$  was the minimum workable value when  $\xi = 1$ . Using the table, researchers determined that the best set of input values for the PCACC controller was  $PCACC(t_r = 30 \text{ sec}, C_1 = 0.2, \xi = 1, s_0 = 200 \text{ m})$ , which has its results bolded.

**Table 12. More Communication Test Plan, PCACC Controller Parameter Variation**

		<b>PTC Failures</b>				
<b>(a)</b>		$C_1$				
		0	0.1	0.2	0.3	0.4
$t_r$ (sec)	15	1	2	2	2	2
	20	0	0	1	1	1
	25	0	0	0	0	1
	30	1	0	0	1	1
	35	2	2	2	1	2

		<b>Average Separation (m)</b>				
<b>(b)</b>		$C_1$				
		0	0.1	0.2	0.3	0.4
$t_r$ (sec)	15					
	20	264.1	264.0			
	25	264.3	264.6	264.9	265.2	
	30		265.6	266.3		
	35					

		<b>Fuel Used (kg)</b>				
<b>(c)</b>		$C_1$				
		0	0.1	0.2	0.3	0.4
$t_r$ (sec)	15					
	20	15490	15030			
	25	13700	13190	12640	12060	
	30		12160	11720		
	35					

		<b>SD Notch Difference</b>				
<b>(d)</b>		$C_1$				
		0	0.1	0.2	0.3	0.4
$t_r$ (sec)	15					
	20	4.1650	3.8750			
	25	3.0090	2.6480	2.2470	1.8060	
	30		1.8740	1.5410		
	35					

For reference, the best set of input values for each controller are summarized below in [Table 13](#). Additionally, [Table 14](#) summarizes the results for these inputs, facilitating direct comparison between controllers. Overall, the MPD controller and the KACC controller again performed very similarly to each other. The PCACC controller performed quite a bit worse than both the MPD controller and the KACC controller.

**Table 13. More Communication Test Plan, Best Input Values Summary**

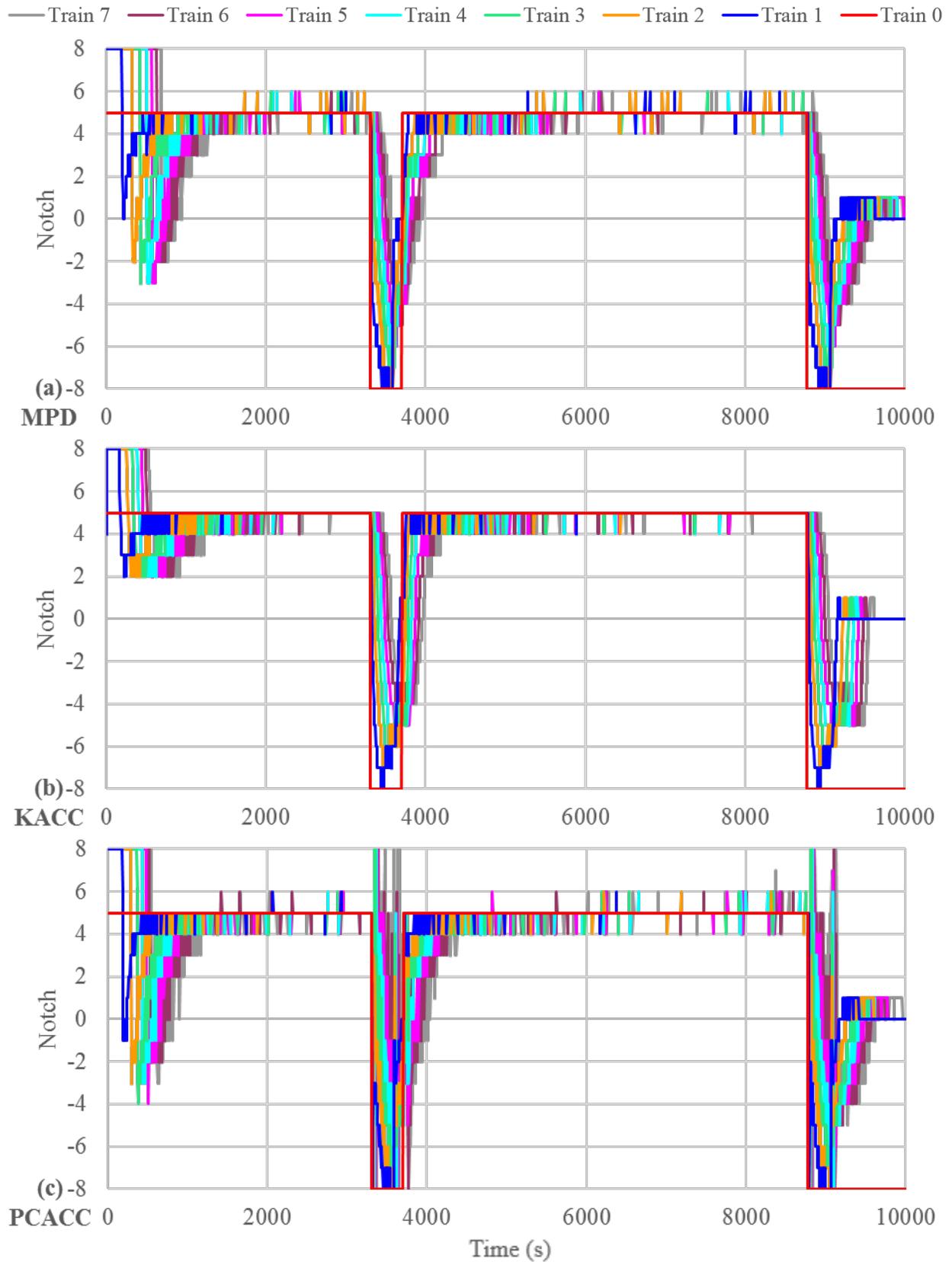
<i>MPD</i> ( $\tau_v = 12$ sec, $\tau_s = 90$ sec, $h = 0$ sec, $s_0 = 2$ m)
<i>KACC</i> ( $h = 8$ sec, $\tau_s = 22$ sec, $t_m = 10$ sec, $s_0 = 2$ m)
<i>PCACC</i> ( $t_r = 30$ sec, $C_1 = 0.2$ , $\xi = 1$ , $s_0 = 200$ m)

**Table 14. More Communication Test Plan, Results Summary**

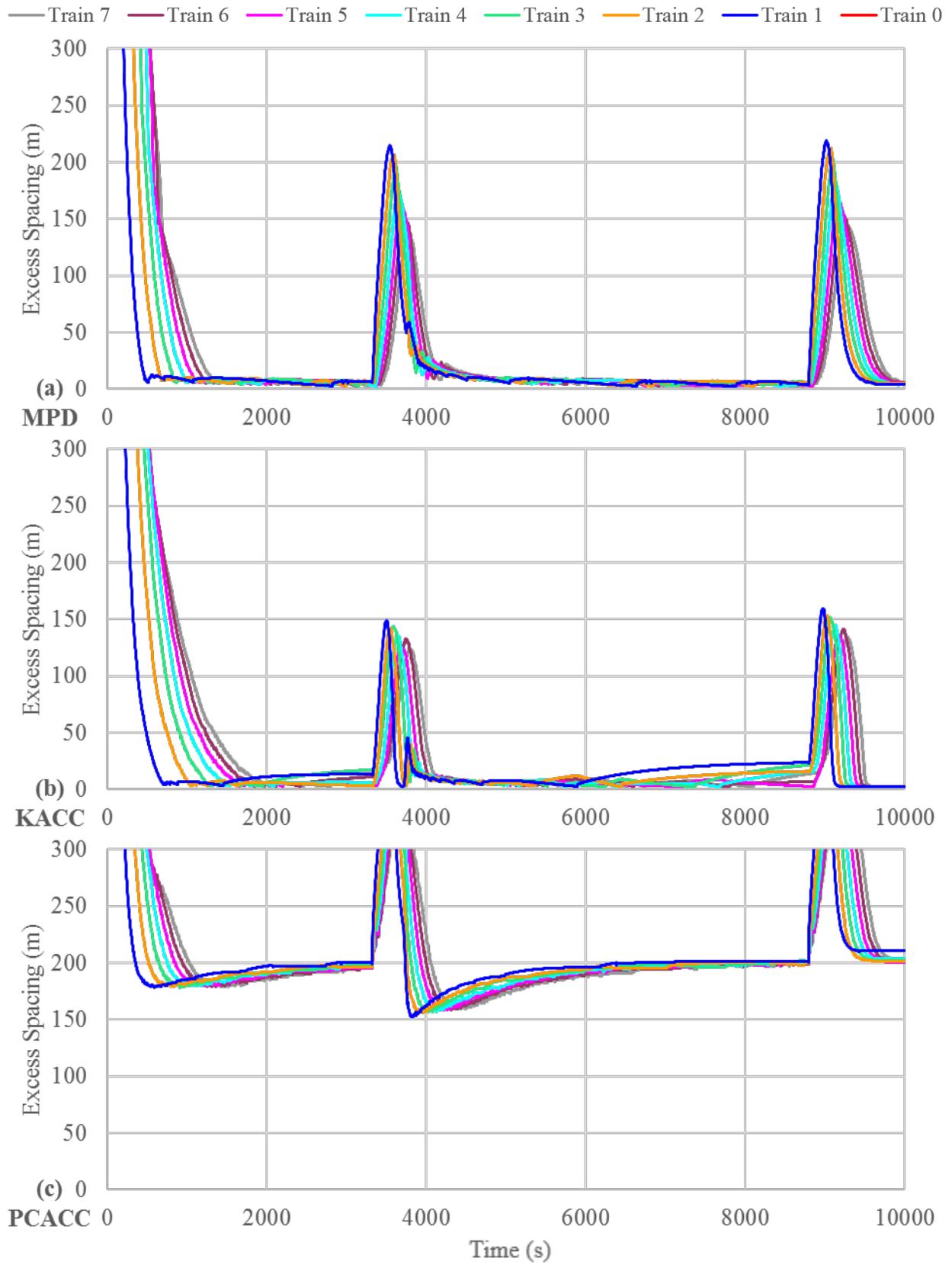
<b>Metric</b>	<b>MPD</b>	<b>KACC</b>	<b>PCACC</b>
<b>PTC Failures</b>	0	0	0
<b>Average Separation (m)</b>	106.3	108.1	266.3
<b>Fuel Used (kg)</b>	11100	11100	11720
<b>SD Notch Difference</b>	0.5600	0.4586	1.5410

Compared to the MPD controller, the KACC controller had a lower standard deviation of notch difference and the same fuel usage, but a higher average separation. Thus, the KACC controller should have been better at managing in-train forces but at the cost of increasing train separation slightly. Since managing in-train forces and minimizing train separation are both important goals, choosing between the MPD controller and the KACC controller will depend on the specific requirements of the railroad and the corridor that moving block is to be implemented on. In contrast, the PCACC controller had worse performance than either the MPD controller or the KACC controller because it had higher average separation, higher fuel usage, and a higher standard deviation of notch difference.

To compare the notch and spacing performance of the controllers versus time, one of the 18 scenarios must be selected. Scenario 5, which had eight loaded grain trains and a front train profile of two cycles of notch 5 to notch -8, showed all important operating situations for the controller, and so was used for comparison. This was because following trains can catch up to the leading train during acceleration but must react very quickly to deceleration events and decelerate at near the max rate. Additionally, using a different scenario than the one used for the position-only test plan means that new information can be learned about the MPD and KACC controllers. Note, however, that this scenario does not show why  $s_0 = 200$  m was necessary for the PCACC controller. That is scenario 13, which had eight intermodal trains and a front train profile of two cycles of notch 8 to notch -8. The point of minimum spacing occurred in the early stages of accelerating after the first intermediate stop. [Figure 14](#) and [Figure 15](#) below show the notch and spacing performance respectively for the MPD, KACC, and PCACC controllers.



**Figure 14. More Communication Test Plan, Notch Performance Comparison**



**Figure 15. More Communication Test Plan, Spacing Excess Comparison**

Of the three controllers, the PCACC controller had the worst performance because it had clear notch instability during all acceleration and deceleration events and because it required more spacing than either of the other two controllers during the entire simulation.

Between the MPD and the KACC controller, the KACC controller again had higher notch stability throughout the simulation and tracked much closer to the lead train. Additionally, the KACC controller used less spacing than the MPD controller during deceleration events. However, the MPD controller kept excess spacing near zero at high speed much more effectively than the KACC controller. This was because the acceleration specified by the KACC controller was insufficient to overcome the discretization method used by the lower level controller, which involves selecting the notch that produces acceleration as close to the target as possible without going over. An improved version of either the KACC controller or the lower level controller should be able to overcome this issue, though the authors were not successful in developing a solution that improved overall performance of the KACC controller.

In summary for this test plan, the MPD and KACC controllers performed similarly to each other and substantially better than the PCACC controller. Therefore, only the MPD and KACC controllers will be considered in subsequent test plans.

#### **5.2.4 Controller Time Step and Interpolation Method**

In addition to improving performance through changing the control law, the network time step, the allowed communication paths, and the variables communicated, there may be potential to improve performance by running the controller more often than once per network update. This has the potential to enable faster and more precise reactions by the train. However, to reduce the controller time step without a large performance regression, some sort of method must be developed to interpolate the communicated variables between network time steps.

Three interpolation methods were developed to do this. The first was the baseline method of independent linear interpolation of each variable. The second was designed for the position-only communication type and ensures that the target position is completely correct when the preceding train either has constant velocity or constant acceleration. The equation is as follows:

$$x_t = \max\left(x_0, \min\left(x_1, x_0 + t \frac{\dot{x}_t + \dot{x}_1}{2}\right)\right) \quad (53)$$

In this equation,  $t_n$  is the network time step,  $t$  is time and  $0 \leq t \leq t_n$ ,  $x_t$  is the interpolated offset at time  $t$ ,  $x_0$  is the previous communicated position of the preceding train,  $x_1$  is the current communicated position of the preceding train,  $\dot{x}_1$  is the approximate velocity of the preceding train calculated for the current network time step, and  $\dot{x}_t$  is the interpolated velocity for time  $t$  calculated using simple linear interpolation between  $\dot{x}_0$  and  $\dot{x}_1$ .

The third was designed for when position, velocity, and acceleration are communicated. The position function is the quintic spline polynomial defined by the boundary conditions of previous and current position, velocity, and acceleration. Interpolated velocity and acceleration are calculated as the derivative and second derivative of this polynomial respectively. The following four equations define this quintic spline polynomial:

$$\begin{aligned}
x_t &= a \left(\frac{t}{t_n}\right)^5 + b \left(\frac{t}{t_n}\right)^4 + c \left(\frac{t}{t_n}\right)^3 + \ddot{x}_0 \frac{t_n^2}{2} \left(\frac{t}{t_n}\right)^2 + \dot{x}_0 t_n \left(\frac{t}{t_n}\right) + x_0 \\
a &= \frac{t_n^2}{2} (\ddot{x}_1 - \ddot{x}_0) - 3t_n (\dot{x}_1 + \dot{x}_0) + 6(x_1 - x_0) \\
b &= -t_n^2 \left(\ddot{x}_1 - \frac{3\ddot{x}_0}{2}\right) + t_n (7\dot{x}_1 + 8\dot{x}_0) - 15(x_1 - x_0) \\
c &= \frac{t_n^2}{2} (\ddot{x}_1 - 3\ddot{x}_0) - 2t_n (2\dot{x}_1 + 3\dot{x}_0) + 10(x_1 - x_0)
\end{aligned} \tag{54}$$

In these equations,  $a$ ,  $b$ , and  $c$  are substituted into the top equation for  $x_t$  to get the full equation,  $t_n$  is the network time step,  $t$  is time and  $0 \leq t \leq t_n$ ,  $x_t$  is the offset at time  $t$ ,  $x_0$ ,  $\dot{x}_0$ ,  $\ddot{x}_0$ ,  $x_1$ ,  $\dot{x}_1$ , and  $\ddot{x}_1$  are the previous and current communicated position, velocity, and acceleration respectively. The derivative and second derivative of this equation that define  $\dot{x}_t$  and  $\ddot{x}_t$  respectively are not shown.

Both the MPD controller and the KACC controller were used to evaluate the effectiveness of decreasing controller time step and changing interpolation method. To keep the total number of simulations manageable, the only controller time step tested was 1 second, four times smaller than the network time step. For each controller for this time step, each of the two communication types were tested with each of their two interpolation methods, giving eight more simulation runs. To effectively summarize these results, only the results from the best set of inputs are shown. The same criteria as before was used to determine the best set of input values for each set of simulation runs. [Table 15](#) and [Table 16](#) below summarize the MPD and KACC controller results, respectively.

**Table 15. Time Step and Interpolation Method, MPD Controller Summary**

Communication Type	Position Only			All Variables		
	4	1		4	1	
Interpolation Type	N/A	Simple	Improved	N/A	Simple	Improved
PTC Failures	0	0	0	0	0	0
Average Separation (m)	105.2	99.7	99.7	106.3	100.6	100.6
Fuel Used (kg)	11140	11120	11120	11100	11090	11090
SD Notch Difference	0.5497	0.4712	0.4710	0.5600	0.4711	0.4625

**Table 16. Time Step and Interpolation Method, KACC Controller Summary**

Communication Type	Position Only			All Variables		
	4	1		4	1	
Interpolation Type	N/A	Simple	Improved	N/A	Simple	Improved
PTC Failures	0	0	0	0	0	0
Average Separation (m)	106.8	104.6	104.6	108.1	105.9	105.9
Fuel Used (kg)	11160	11160	11170	11100	11110	11120
SD Notch Difference	0.4566	0.4394	0.4395	0.4586	0.4397	0.4252

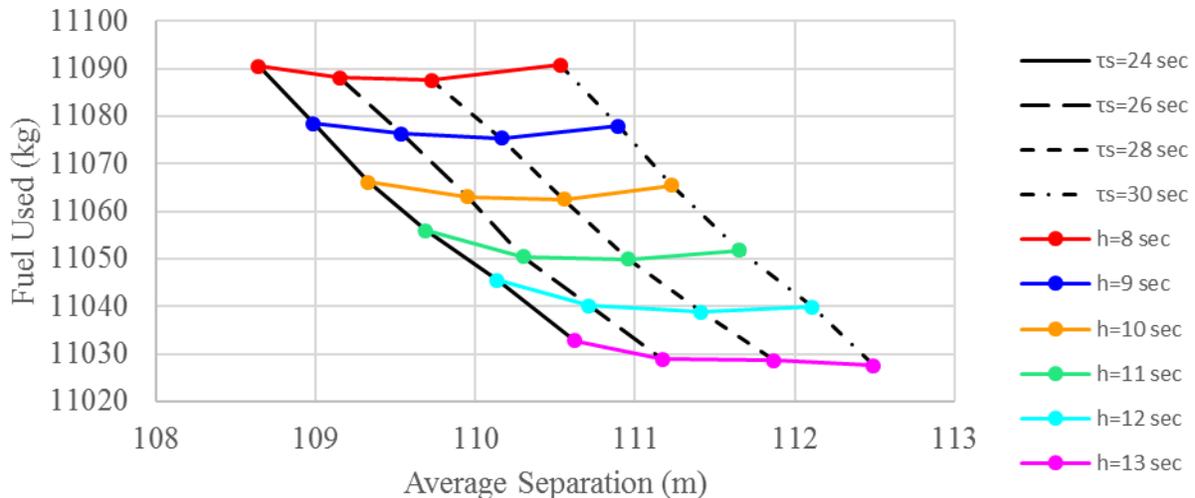
For both controllers, the position-only communication type had lower average separation and higher fuel consumption compared to the corresponding controller time step and interpolation type simulations. This was somewhat surprising, as it indicates that using more up-to-date velocity and acceleration information resulted in larger train separation and minimal change in the standard deviation of notch difference rather than resulting in an improvement to all metrics.

When moving to a smaller controller time step, both controllers showed substantial improvements in average separation and standard deviation of notch difference. For the MPD controller, fuel usage also decreased, while it stayed nearly constant for the KACC controller. Thus, there were clear benefits to reducing the controller time step. Note that the observed reduction in standard deviation of notch difference was likely due to bias in the metric itself. This was because it assigned a higher weight to large notch shifts over a single time step, which will typically be spread out more when using a smaller time step.

For the position-only communication type with a controller time step of 1 second, there was essentially no benefit to using the improved interpolation method (the second interpolation method). This indicates that it was simply not possible to do better than simple linear interpolation when receiving position-only information. In contrast, for the all variables communication scenario with a controller time step of 1 second, there was a measurable reduction in the standard deviation of notch difference, though average separation and fuel usage changed very little. It was valid to make this conclusion because the comparison was occurring between simulations that used the same controller time step.

### 5.2.5 Spacing and Fuel Consumption Tradeoff

To illustrate how controllers can be customized to meet specific business objectives, the fuel used was plotted versus average separation below in Figure 16. Data was generated from inputs of the form  $KACC(h, \tau_s, t_m = 10 \text{ sec}, s_0 = 2 \text{ m})$  using the more communication test plan.



**Figure 16. Average Separation and Fuel Consumption Tradeoff for KACC Controller**

For the KACC controller, the parameter  $h$  directly controlled the tradeoff between train separation and fuel consumption. This made sense because increasing train-following headway

should increase train separation while also decreasing fuel consumption by smoothing oscillations more aggressively. In contrast, the parameter  $\tau_s$  controlled the amount of spacing while only marginally affecting fuel consumption, with higher values corresponding to larger spacing. This also made sense because in the formulation,  $\tau_s$  was only used to moderate how fast following trains approached zero excess spacing to prevent PTC enforcements.

Together, the settings of these two control parameters defined a pareto-optimal frontier with respect to the objective of minimizing both fuel consumption and average train separation. Using this tradeoff, individual railroads and train operators can tune the performance of following train control algorithm depending on their desired business objectives. On a capacity-constrained corridor where minimum headway is a priority, lower values of the parameter  $h$  may be specified at the expense of increased fuel consumption. Larger values of parameter  $h$  and parameter  $\tau_s$  may be specified on corridors where increased train spacing can be tolerated but the cost of fuel consumption (and resulting emissions) is of primary importance.

### 5.3 Results Summary

Of the six controllers developed and evaluated through this research, the MPD controller and the KACC controller consistently exhibited similar performance and substantially better performance than all other controllers tested for all metrics considered. Of the four remaining controllers, the naïve controller had by far the worst performance because it required a large amount of excess spacing and rapidly commanded large changes in throttle and brake status. Between the PD controller and the CACC controller, the PD controller performed better in all metrics, as shown in [Table 9](#) for the position-only test plan. While this test plan did not include the PCACC controller, the results of [Table 14](#) for the more communication test plan can be considered directly by noting that the MPD and KACC controller performance is quite similar in both [Table 9](#) and [Table 14](#). Thus, the PCACC controller performed better than the CACC controller but worse than the PD controller. Overall, the naïve controller performed the worst, followed by the CACC controller, the PCACC controller, the PD controller, and both the MPD and KACC controllers, which performed the best.

Between the MPD controller and the KACC controller, both had advantages and disadvantages. While the KACC controller generally had slightly larger average train separation, this mostly occurred when trains were moving at high speed – rarely the limiting factor for capacity. Additionally, it had a lower standard deviation of notch difference, it was designed to ensure safe train separation for logically reasonable parameters, and it was successfully tested in virtual coupling scenarios where the MPD controller failed unless headway was increased to a value substantially higher than what was used for the KACC controller. Thus, a controller based on the KACC controller that fixes the train separation at high speed should be developed and should convincingly outperform the MPD controller in all metrics used.

As compared to ideal moving block with no excess spacing, the MPD and KACC controllers both performed very well because their average separation was on the order of 100 meters. Considering that moving-block calculations typically use a conservative value for braking distance, it was clear that well-designed moving-block systems will be able to very nearly reach the theoretical performance if calculated using the longest braking distance for any train that will traverse the network. More simulations and controller changes need to be conducted to determine how close real-world systems can come to theoretical moving block with fully customized braking distances.

## 6. Conclusion

---

Directly adapting highway vehicle platooning controllers to the heavy-haul freight and passenger railway domain is difficult due to the orders of magnitude difference in highway and rail vehicle performance. However, highway controllers do suggest families of control laws can be adapted to the train-following problem. The simulation results suggested that a series of control laws were better than others at managing the train separation and fuel consumption within train fleets. Certain controllers were fast-acting but demonstrated notch instability when attempting to minimize headways, while other controllers were slow-acting and required a large baseline train spacing to avoid an incursion into the minimum safe braking distance and a corresponding PTC enforcement. While moving blocks require additional train spacing beyond the minimum safe braking distance to account for train control actions, certain following train control algorithms can help minimize this distance. The developed control laws also exhibited a tradeoff that may allow railway operators to change algorithm parameters and balance fuel efficiency and train headway to meet their specific business objectives.

Relative to the scenario where only information on the position of the train ahead was known, the headway and fuel efficiency performance of the best following train control algorithms could be improved by communicating additional information on the speed and acceleration of the train ahead. These benefits were enhanced when the frequency of train position reports and controller updates was increased. This result suggests that enhanced communication may be essential to effectively managing train fleets and achieving the full capacity benefits of moving blocks.

The above conclusions are subject to the limitations of this research scope. The best-performing train following algorithms within the limited scope of this research should be subjected to more complex conditions involving actual rail corridor grade and curve topography as part of future research and development of following train control algorithms for moving blocks. While this research only included heavy-haul freight trains, future research and development work should consider passenger and commuter trains to determine if their distinct performance characteristics can still be managed with the same types of control algorithms but using different control algorithm parameters.

The train-following control algorithms developed through this research are critical to future attempts to increase capacity and efficiency through advanced train control systems with moving blocks. By allowing trains to maintain minimum headways, the best train-following algorithms will help the rail industry achieve the anticipated capacity benefits of advanced PTC with moving blocks without sacrificing energy efficiency. The algorithms developed through this research will also help maintain railway safety by reducing the number of repeated, successive brake applications made by closely following trains that fail to properly anticipate the speed of trains operating ahead of them in a fleet. With additional research and development, researchers anticipate that locomotive manufacturers, and other third-party vendors of locomotive and train control systems, will use the best-performing families of algorithms identified through this research as a starting point for their control laws as moving blocks enter test implementation.

## 7. References

---

- Albrecht, A., P. Howlett, P. Pudney, X. Vu and P. Zhou. (2016). The key principles of optimal train control—Part 1: Formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points. *Transportation Research Part B: Methodological*. 94: 482-508.
- Alikoc, B., I. Mutlu and A.F. Ergenc. (2013). Stability Analysis of Train-following Model with Multiple Communication Delays. *IFAC Proceedings Volumes*. 46(25): 13-18.
- Amoozadeh, M., H. Deng, C.N. Chuah, H.M. Zhang, and D. Ghosal. (2015). Platoon management with cooperative adaptive cruise control enabled by VANET. *Vehicular communications*. 2(2): 110-123.
- Bando, M., K. Hasebe, A. Nakayama, A. Shibata and Y. Sugiyama. (1995). Dynamical model of traffic congestion and numerical simulation. *Physical review E*, 51(2): 1035.
- Barney, D., D. Haley and G. Nikandros. (2001). Calculating Train Braking Distance. *Proceedings of the Sixth Australian Workshop on Safety Critical Systems and Software*. 3: 23-29.
- Brackstone, M. and M. McDonald. (1999). Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*. 2(4):181-196.
- Chandler, R.E., R. Herman and E.W. Montroll. (1958). Traffic dynamics: studies in car following. *Operations research*. 6(2):165-184.
- Chang, K.S., W. Li, P. Devlin, A. Shaikhbahai, P. Varaiya, J.K. Hedrick, and J. Olds. (1991). Experimentation with a vehicle platoon control system. *Vehicle Navigation and Information Systems Conference*. 2: 1117-1124.
- Chen, N., M. Wang, T. Alkim and B. van Arem. (2018). A robust longitudinal control strategy of platoons under model uncertainties and time delays. *Journal of Advanced Transportation*.
- Deng, Q. (2016). A general simulation framework for modeling and analysis of heavy-duty vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*. 17(11): 3252-3262.
- Dey, K.C., L. Yan, X. Wang, Y. Wang, H. Shen, M. Chowdhury and V. Soundararaj. (2015). A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC). *IEEE Transactions on Intelligent Transportation Systems*. 17(2): 491-509.
- Diaz de Rivera, A., C.T. Dick and L.E. Evans. (2020a). Improving railway operational efficiency with moving blocks, train fleeting and alternative single-track configurations. *Transportation Research Record: Journal of the Transportation Research Board*. 2674 (2): 146-157.
- Diaz de Rivera, A., C.T. Dick and L.E. Evans. (2020b). Potential for moving blocks and train fleets to enable faster train meets on single-track rail corridors. *Journal of Transportation Engineering, Part A: Systems*. 146 (8): 04020077.

- Diaz de Rivera, A., and C.T. Dick. (2021). Illustrating the implications of moving blocks on railway traffic flow behavior with fundamental diagrams. *Transportation Research Part C: Emerging Technologies*. 123: 102982.
- Dick, C.T. (2000). *Impact of Positive Train Control on Railway Capacity*. Independent Study Project, University of Illinois at Urbana-Champaign, Department of Civil and Environmental Engineering, Urbana, IL.
- Dick, C.T., D. Mussanov, L.E. Evans, G.S. Roscoe and T-Y. Chang. (2019). Relative capacity and performance of fixed and moving block control systems on North American freight railway lines and shared passenger corridors. *Transportation Research Record: Journal of the Transportation Research Board*. 2673(5): 250-261
- Dingler, M.H., Y-C. Lai and C.P.L. Barkan. (2009). Impact of CBTC and ECP Brakes on Capacity. In: *Proceedings of the American Railway Engineering and Maintenance-of-Way Association Annual Conference*.
- Dingler, M.H., Y-C. Lai and C.P.L. Barkan. (2010). Effects of communications-based train control and electronically controlled pneumatic brakes on railroad capacity. *Transportation Research Record: Journal of the Transportation Research Board*. 2159: 77-84.
- Durmus, M.S., K. Ucak, G. Oke and M.T. Soylemez. (2013). Train Speed Control in Moving-Block Railway Systems: An Online Adaptive PD Controller Design. *IFAC Proceedings Volumes*. 46(25): 7-12.
- Feng, S., Y. Zhang, S.E. Li, Z. Cao, H.X. Liu and L. (2019). String stability for vehicular platoon control: Definitions and analysis methods. *Annual Reviews in Control*. 47: 81-97.
- Fritz, H. (1999). Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic: experimental results from the CHAUFFEUR project. In *Proceedings of the 1999 IEEE International Conference on Control Applications* (Cat. No. 99CH36328). 2: 1348-1352.
- Gao, S., H. Dong, B. Min, C. Roberts, L. Chen and X. Sun. (2015). Cooperative adaptive bidirectional control of a train platoon for efficient utility and string stability. *Chinese Physics B*. 24(9).
- Gao, S., H. Dong, B. Ning and Y. Li. (2016). Composite Pi-like adaptive dynamic surface control for interval coordination of multiple trains. In *Proceeding of 12th IEEE International Conference on Control and Automation (ICCA)*: 437-442.
- Gazis, D.C., R. Herman and R.B. Potts. (1959). Car-following theory of steady-state traffic flow. *Operations research*. 7(4):499-505.
- Gehring, O., and H. Fritz. (1997). Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication. In *Proceedings of IEEE Conference on Intelligent Transportation Systems*. pp. 117-122.
- Gipps, P.G. (1981). Behavioral car-following model for computer simulation. *Transport. Res.* 15(2): 105-111.

- Gong, S., and L. Du. (2018). Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation research part B: methodological*. 116: 25-61.
- Gong, S., A. Zhou and S. Peeta. (2019). Cooperative adaptive cruise control for a platoon of connected and autonomous vehicles considering dynamic information flow topology. *Transportation Research Record*. 2673(10): 185-198.
- Gordon, S.P., D.G. Lehrer. (1998). Coordinated Train Control and Energy Management Control Strategies. *Proceedings of the 1998 ASME/IEEE Joint Railroad Conference*. 165-176.
- Guo, G., and W. Yue. (2014). Sampled-data cooperative adaptive cruise control of vehicles with sensor failures. *IEEE Transactions on Intelligent Transportation Systems*. 15(6): 2404-2418.
- Ho, M.T.K. (1998). A Multi-Train Movement Simulator with Moving Block Signalling. *Transactions on the Built Environment*. 34.
- Ioannou, P.A. and C.C. Chien. (1993). Autonomous intelligent cruise control. *IEEE Transactions on Vehicular technology*. 42(4): 657-672.
- Isaksson Palmqvist, M. (2016). *Model predictive control for autonomous driving of a truck*.
- Jaekel, B. and T. Albrecht. (2014). Comparative analysis of algorithms and models for train running simulation. *Journal of Rail Transport Planning and Management*. 4(1-2): 14-27.
- Jia, D., and D. Ngoduy. (2016a). Enhanced cooperative car-following traffic model with the combination of V2V and V2I communication. *Transportation Research Part B: Methodological*. 90: 172-191.
- Jia, D., and D. Ngoduy. (2016b). Platoon based cooperative driving model with consideration of realistic inter-vehicle communication. *Transportation Research Part C: Emerging Technologies*. 68: 245-264.
- Jiang, R., Q. Wu and Z. Zhu. (2001). Full velocity difference model for a car-following theory. *Physical Review E*. 64(1): 017101.
- Jin, I.G., and G. Orosz. (2014). Dynamics of connected vehicle systems with delayed acceleration feedback. *Transportation Research Part C: Emerging Technologies*. 46: 46-64.
- Johansen, T.A., K.J. Hunt, P.J. Gawthrop and H. Fritz. (1998). Off-equilibrium linearisation and design of gain-scheduled control with application to vehicle speed control. *Control Engineering Practice*. 6(2): 167-180.
- Karredla, G.V. and C.L. Srinivas. (2014). A Quasi-Moving Block Redundant Non Vital System Model for Conventional Single Line Absolute Block Railway System. *2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO)*. 127-132.
- Kato, S., S. Tsugawa, K. Tokuda, T. Matsui and H. Fujii. (2002). Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. *IEEE Transactions on intelligent transportation systems*. 3(3): 155-161.

- Köroğlu, H., M. Mirzaei, P. Falcone and S. Krajnović. (2018). Platoon Control Under a Novel Leader and Predecessor Following Scheme With the Use of an Advanced Aerodynamic Model. *Journal of Dynamic Systems, Measurement, and Control*. 140(4).
- Kraay, D., P.T. Harker and B. Chen. (1991). Optimal Pacing of Trains in Freight Railroads: Model Formulation and Solution. *Operations Research*. 39(1): 2-177.
- Li, K. and L. Guan. (2009). Simulating train movement in railway traffic using a car-following model. *Chinese Physics B*. 18(6).
- Li, K. and Z. Gao. (2007). An improved equation model for the train movement. *Simulation Modelling Practice and Theory*. 15(9): 1156-1162.
- Li, K. and Z. Gao. (2013). An improved car-following model for railway traffic. *Journal of Advanced Transportation*. 47(4): 475-482.
- Li, K., Z. Gao and B. Ning. (2005). Modeling the Railway Traffic using Cellular Automata Model. *International Journal of Modern Physics C*. 16(6):921-932.
- Li, K., Z. Gao and T. Tang. (2011). Modelling and Simulation for Train Movement Control Using Car-Following Strategy. *Communications in Theoretical Physics*. 55(1).
- Li, S., L. Yang and Z. Gao. (2015). Coordinated cruise control for high-speed train movements based on a multi-agent model. *Transportation Research Part C: Emerging Technologies*. 56: 281-292.
- Liang, C.Y. and H. Peng. (1999). Optimal adaptive cruise control with guaranteed string stability. *Vehicle system dynamics*. 32(4-5):313-330.
- Liang, K.Y., S. van de Hoef, H. Terelius, V. Turri, B. Besselink, J. Mårtensson and K.H. Johansson. (2016). Networked control challenges in collaborative road freight transport. *European Journal of Control*. 30: 2-14.
- Lima, P.F. (2018). *Optimization-based motion planning and model predictive control for autonomous driving: With experimental evaluation on a heavy-duty construction truck* (Doctoral dissertation, KTH Royal Institute of Technology).
- Liu, R. (2016). Simulation model of speed control for the moving-block systems under ERTMS Level 3. *2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT)*. 322-327.
- Ma, J., F. Zhou, Z. Huang and R. James. (2018). Hardware-in-the-loop testing of connected and automated vehicle applications: a use case for cooperative adaptive cruise control. *21st International Conference on Intelligent Transportation Systems (ITSC)*. pp. 2878-2883.
- Milanés, V., S.E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe and M. Nakamura. (2013). Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems*. 15(1): 296-305.
- Milanés, V., and S.E. Shladover. (2014). Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transportation Research Part C: Emerging Technologies*. 48: 285-300.

- Montanaro, U., M. Tufo, G. Fiengo, M. di Bernardo and S. Santini. (2014). On convergence and robustness of the extended cooperative cruise control. In *53rd IEEE Conference on Decision and Control*: 4083-4088.
- Naus, G.J., R.P. Vugts, J. Ploeg, M.J. van De Molengraft, and M. Steinbuch. (2010). String-stable CACC design and experimental validation: A frequency-domain approach. *IEEE Transactions on vehicular technology*. 59(9): 4268-4279.
- Nieuwenhuijze, M.R.I. (2010). *String stability analysis of bidirectional adaptive cruise control*. Eindhoven University of Technology.
- Ning, B. (1998). Absolute braking and relative distance braking train operation control modes in moving block systems. *Transactions on the Built Environment*. 34.
- Öncü, S., J. Ploeg, N. Van de Wouw and H. Nijmeijer. (2014). Cooperative adaptive cruise control: Network-aware analysis of string stability. *IEEE Transactions on Intelligent Transportation Systems*. 15(4): 1527-1537.
- Pan, D. and Y. Zheng. (2014). Dynamic Control of High-speed Train-following Operation. *Promet-Traffic and Transportation*. 26(4).
- Panou, K., P. Tzieropoulos and D. Emery. (2013). Railway driver advice systems: Evaluation of methods, tools and systems. *Journal of Rail Transport Planning and Management*. 3(4): 150-162.
- Ploeg, J., B.T. Scheepers, E. Van Nunen, N. Van de Wouw and H. Nijmeijer. (2011). Design and experimental evaluation of cooperative adaptive cruise control. *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. pp. 260-265.
- Ploeg, J., E. Semsar-Kazerooni, G. Lijster, N. van de Wouw and H. Nijmeijer. (2013). Graceful degradation of CACC performance subject to unreliable wireless communication. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*: 1210-1216.
- Polivka, A., B.M. Ede and J. Drapa. (2009). North American Joint Positive Train Control Project. *FRA. DOT/FRA/ORD-09/04*: 1-77.
- Qin, W.B., M.M. Gomez, and G. Orosz. (2016). Stability and frequency response under stochastic communication delays with applications to connected cruise control design. *IEEE Transactions on Intelligent Transportation Systems*, 18(2): 388-403.
- Rajamani, R. (2011). *Vehicle dynamics and control*. Springer Science & Business Media.
- Ramezani, H., S.E. Shladover, X.Y. Lu, and O.D. Altan. (2018). Micro-Simulation of Truck Platooning with Cooperative Adaptive Cruise Control: Model Development and a Case Study. *Transportation Research Record*. 2672(19): 55-65.
- Sheikholeslam, S., and C.A. Desoer. (1990). Longitudinal control of a platoon of vehicles. *IEEE American control conference*. pp. 291-296.
- Shladover, S.E., C.A. Desoer, J.K. Hedrick, M. Tomizuka, J. Walrand, W.B. Zhang and N. McKeown. (1991). Automated vehicle control developments in the PATH program. *IEEE Transactions on vehicular technology*. 40(1): 114-130.

- Shladover, S.E., D. Su and X.Y. Lu. (2012). Impacts of cooperative adaptive cruise control on freeway traffic flow. *Transportation Research Record*. 2324(1): 63-70.
- Shladover, S.E., C. Nowakowski, X.Y. Lu, and R. Ferlis. (2015). Cooperative adaptive cruise control: Definitions and operating concepts. *Transportation Research Record*. 2489(1): 145-152.
- Slotine, J.J.E. and W. Li. (1991). *Applied nonlinear control*. Prentice Hall, Englewood Cliffs, NJ. 199(1).
- Song, Q., Y. Song, T. Tang and B. Ning. (2011). Computationally Inexpensive Tracking Control of High-Speed Trains With Traction/Braking Saturation. *IEEE Transactions on Intelligent Transportation Systems*. 12(4): 1116-1125.
- Stern, R.E., S. Cui, M.L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, H. Pohlmann, F. Wu, B. Piccoli and B. Seibold. (2018). Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*. 89: 205-221.
- Su, S., T. Tang and C. Roberts. (2011). A Cooperative Train Control Model for Energy Saving. *IEEE Transactions on Intelligent Transportation Systems*. 16(2): 622-631.
- Sugiyama, Y., M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.I. Tadaki and S. Yukawa. (2008). Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam. *New journal of physics*. 10(3): 033001.
- Swaroop, D., J.K. Hedrick, C.C. Chien and P. Ioannou. (1994). A comparison of spacing and headway control laws for automatically controlled vehicles. *Vehicle system dynamics*. 23(1): 597-625.
- Sybis, M., V. Vukadinovic, M. Rodziewicz, P. Sroka, A. Langowski, K. Lenarska and K. Wesółowski. (2019). Communication aspects of a modified cooperative adaptive cruise control algorithm. *IEEE Transactions on Intelligent Transportation Systems*. 20(12): 4513-4523.
- Takagi, R. (2012). Synchronisation control of trains on the railway track controlled by the moving block signalling system. *IET Electrical Systems in Transportation*. 2(3): 130-138.
- Takeuchi, H., C.J. Goodman and S. Sone. (2003). Moving block signalling dynamics: performance measures and re-starting queued electric trains. *IEE Proceedings – Electric Power Applications*. 150(4): 482-492.
- Talebpour, A. and H.S. Mahmassani. (2016). Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*. 71: 143-163.
- Tang, H., C.T. Dick and X. Feng. (2015). A coordinated train control algorithm to improve regenerative energy receptivity in metro transit systems. *Transportation Research Record: Journal of the Transportation Research Board*. 2534: 48–56.
- Tang, T. and K. Li. (2007). Traffic Modelling for Moving-Block Train Control System. *Communications in Theoretical Physics*. 47(4): 601-606.

- Treiber, M., A. Hennecke and D. Helbing. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2): 1805.
- van Arem, B., C.J. Van Driel and R. Visser. (2006). The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on intelligent transportation systems*. 7(4): 429-436.
- van der Heijden, R., T. Lukaseder and F. Kargl. (2017). Analyzing attacks on cooperative adaptive cruise control (CACC). *IEEE Vehicular Networking Conference (VNC)*, pp. 45-52.
- Wang, C., and H. Nijmeijer. (2015). String stable heterogeneous vehicle platoon using cooperative adaptive cruise control. In *IEEE 18th International Conference on Intelligent Transportation Systems*. pp. 1977-1982.
- Wang, L. and B.K. Horn. (2019). On the chain stability of bilateral control model. *IEEE Transactions on Automatic Control*.
- Wang, M., W. Daamen, S.P. Hoogendoorn and B. van Arem. (2014). Rolling horizon control framework for driver assistance systems. Part II: Cooperative sensing and cooperative control. *Transportation research part C: emerging technologies*. 40: 290-311.
- Wang, M., J. Zeng, Y. Qian, W. Li, F. Yang and X. Jia. (2012). Properties of train traffic flow in a moving block system. *Chinese Physics B*. 21(7).
- Wang, P. and R.M.P. Goverde. (2016). Two-Train Trajectory Optimization with a Green-Wave Policy. *Transportation Research Record: Journal of the Transportation Research Board*. 2546: 112-120.
- Wang, Y. (2014). Optimal Trajectory Planning and Train Scheduling for Railway Systems. *Delft Center for Systems and Control*.
- Wang, Y., B. De Schutter, T.J.J van der Boom and B. Ning. (2013a). Optimal Trajectory Planning for Trains under a Moving Block Signaling System. *Proceedings of the 2013 European Control Conference*. 4556-4561.
- Wang, Y., B. De Schutter, T.J.J van der Boom and B. Ning. (2013b). A Suboptimal Control Scheme of Multiple Trains Based on Mode Vector Constraints. *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen)*. 1-12.
- Wang, Y., B. De Schutter, T.J.J van der Boom and B. Ning. (2014). Optimal trajectory planning for trains under fixed and moving signaling systems using mixed integer linear programming. *Control Engineering Practice*. 22: 44-56.
- Wilson, R.E. and J.A. Ward. (2011). Car-following models: fifty years of linear stability analysis—a mathematical perspective. *Transportation Planning and Technology*. 34(1): 3-18.
- Wu, C., A.M. Bayen, and A. Mehta. (2018). Stabilizing traffic with autonomous vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1-7.

- Wu, C., A. Kreidieh, K. Parvate, E. Vinitsky and A.M. Bayen. (2017). Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv preprint arXiv:1710.05465*.
- Xia, X. and J. Zhang. (2011). Modeling and Control of Heavy-Haul Trains. *IEEE Control Systems Magazine*. 31(4): 18-31.
- Xiao, L. and F. Gao. (2011). Practical string stability of platoon of adaptive cruise control vehicles. *IEEE Transactions on intelligent transportation systems*. 12(4): 1184-1194.
- Xu, L., W. Zhuang, G. Yin and C. Bian. (2018). Stable longitudinal control of heterogeneous vehicular platoon with disturbances and information delays. *IEEE Access*. 6: 69794-69806.
- Xu, X., K. Li and L. Yang. (2014). Discrete event model-based simulation for train movement on a single-line railway. *Chinese Physics B*. 23(8).
- Xun, J., B. Ning, T. Tang and H. Dong. (2013). Cooperative Control for Train Headway Adjustment in Railway Traffic. *IFAC Proceedings Volumes*. 46(25): 1-6.
- Yang, L., F. Li, Z. Gao and K. Li. (2010). Discrete-time movement model of a group of trains on a rail line with stochastic disturbance. *Chinese Physics B*. 19(10).
- Ye, J. and K. Li. (2013). Simulation optimization for train movement on a single-track railway. *Chinese Physics B*. 22(5).
- Ye, J., K. Li and X. Jin. (2013). Simulating train movement in an urban railway based on an improved car-following model. *Chinese Physics B*. 22(12).
- Ying, Y., T. Mei, Y. Song and Y. Liu. (2014). A sliding mode control approach to longitudinal control of vehicles in a platoon. In *2014 IEEE International Conference on Mechatronics and Automation*. pp. 1509-1514.
- Zhang, R., K. Li, Z. He, H. Wang and F. You. (2017). Advanced emergency braking control based on a nonlinear model predictive algorithm for intelligent vehicles. *Applied Sciences*. 7(5): 504.
- Zhao, S. and K. Zhang. (2018). A data-driven Model Predictive Control framework for robust Cooperative Adaptive Cruise Control using mobile sensing data. Presented at the *Transportation Research Board 97<sup>th</sup> Annual Meeting*, Washington, DC.
- Zhao, S. and K. Zhang. (2020). A distributionally robust stochastic optimization-based Model Predictive Control with distributionally robust chance constraints for Cooperative Adaptive Cruise Control under uncertain traffic conditions. *Transportation Research Part B: Methodological*, 138, pp144-178.
- Zhao, Y., P. Orlik, K. Parsons, K. Kataoka and T. Kalmar-Nagy. (2016). Improvement of train transportation performance by convoy signaling. *International Journal of Modern Physics C*. 27(7).
- Zheng, Y., S.E. Li, K. Li, F. Borrelli and J.K. Hedrick. (2016). Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Transactions on Control Systems Technology*. 25(3): 899-910.

- Zhou, J. and H. Peng. (2005). Range policy of adaptive cruise control vehicles for improved flow stability and string stability. *IEEE Transactions on intelligent transportation systems*. 6(2):229-237.
- Zhou, Y. and C. Mi. (2012). Modeling Train Movement for Moving-Block Railway Network Using Cellular Automata. *Computer Modeling in Engineering and Sciences*. 83(1): 1-22.
- Zhou, Y., S. Ahn, M. Chitturi and D.A. Noyce. (2017). Rolling horizon stochastic optimal control strategy for ACC and CACC under uncertainty. *Transportation Research Part C: Emerging Technologies*. 83: 61-76.
- Zhou, Y., M. Wang and S. Ahn. (2019). Distributed model predictive control approach for cooperative car-following with guaranteed local and string stability. *Transportation research part B: methodological*. 128: 69-86.
- Zhuan, X. and X. Xia. (2006). Cruise Control Scheduling of Heavy-haul Trains. *IEEE Transactions on Control Systems Technology*. 14(4): 757-766.

## Abbreviations and Acronyms

---

<b>ACRONYMS</b>	<b>EXPLANATION</b>
AAR	Association of American Railroads
ACC	Adaptive Cruise Control
CACC	Cooperative Adaptive Cruise Control
CAV	Connected and Automated Vehicles
ETCS	European Train Control System
FRA	Federal Railroad Administration
KACC	Kinematic Adaptive Cruise Control
LEADER	Locomotive Engineer Assist/Display & Event Recorder
MPD	Modified Proportional Derivative (controller)
PCACC	Predictive Cooperative Adaptive Cruise Control
PD	Proportional Derivative (controller)
PTC	Positive Train Control