CAN Coach: Human Cyber Physical System for Time-Space Control in Individual Vehicles

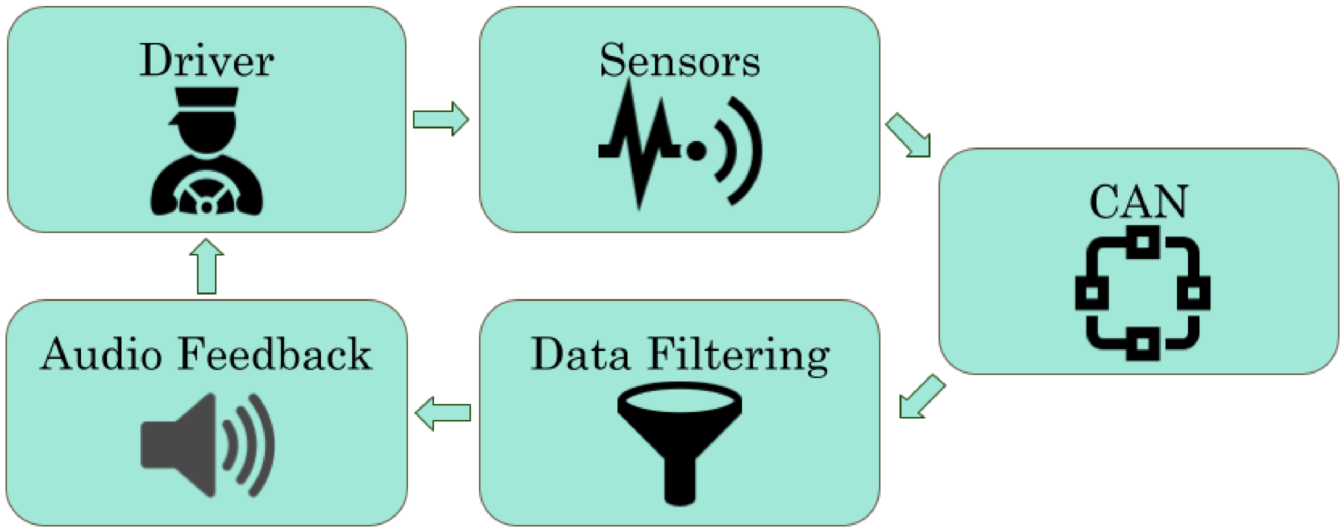


Figure 1. Diagram showing the flow of information in the CAN Coach.[MWN: the ultimate teaser figure would be the picture of the vehicles on the I-24 corridor from the cameras above, or something better if we can think of it]

Abstract

In this work, we address the question of whether a *human-in-the-loop cyber physical system* (HCPS) can be effective in controlling the time-space of an individual vehicle in a traffic flow. We introduce the *CAN Coach*, which is a system that gives feedback to the human driver from *controller area network* (CAN) data that is extrasensory to a human. Four sets of tests are conducted in which the study vehicle follows a lead vehicle driving with varied instruction, to assess the nature and limitations of the HCPS created by driving with CAN Coach. For each test except the last, there is a portion of *Instructed Driving* (driver is given only driving prompts, but is not given any real-time feedback from the CAN Coach), and *Coached Driving* (prompting and CAN Coach feedback). The last test is an 'ablation study' to compare the relative import [RB: should it be relative importance?] of the human perception and the CAN Coach. The driver is given CAN Coach feedback but the data used is from a 'ghost' car instead of based on the physical roadway; this removes the human perception of the lead vehicle from effecting control. The mean time gap errors, a measure of the vehicle's time-space distance from the control task, were 0.35 s (Instructed Driving), 0.13 s (Ghost Driving), and 0.03 s (Coached Driving). The standard deviation of the time gap errors are reduced by 54% and 44% when comparing Coached Driving to Instructed Driving and Ghost Driving respectively. Given these reductions of mean and standard deviation of the time gap error, we conclude that (1) to "coach" drivers using only data from the CAN is effective and (2) this is a true HCPS, since removing human perception from the control loop reduces performance at the given control task.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Keywords: human-in-the-loop, cyber physical systems, controller area network, vehicles

ACM Reference Format:
. 2020. CAN Coach: Human Cyber Physical System for Time-Space Control in Individual Vehicles. In *12th ACM/IEEE International Conference on Cyber Physical Systems, 19-21 May, 2021, Nashville,*

TN. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The goal of Lagrangian control is to control from within the flow of traffic. Outcomes of Lagrangian control could include either modification of a vehicle's velocity, or its position—or position and velocity. In the interest of controlling traffic, it is necessary to be able to control not just the velocity of a vehicle (which might leave significant space that would encourage lane changing) but would also control the distances between the control (ego) car with the lead car, in order to influence the velocities of the vehicles behind through normal driver behaviors.

Research has been done to evaluate this idea with automated vehicles [17]. It shows automation can smooth traffic waves, even with relatively low penetration rates, but can a human driver be enabled to drive in a manner that smooths traffic waves? If a human driver can be enabled to follow commands corresponding to Lagrangian control, this is possible.

In this paper, we ask drivers to follow a lead vehicle in their car (the ego vehicle) in order to determine how to positively improve time-gap following through the use of a CAN Coach. This CAN Coach uses information from the CAN bus such as instantaneous velocity along with estimates of the relative position and velocity of the lead vehicle. This data was collected using only data from the CAN in the following vehicle.[1]. Drivers are asked to follow a lead vehicle with several different objectives, as well as to follow an imaginary car (a "ghost" car) by using the CAN Coach to follow a space gap, but without the visual feedback to the human driver from the lead vehicle.

Our results show that with the CAN Coach, coached drivers can closely control the time gap between the lead vehicle and the ego car. Time gap is defined as the time separation between a lead vehicle's rear bumper and the following vehicle's front bumper when the follower travels at a constant speed[10], i.e. space as a function of speed. The driving behavior with the CAN Coach is significantly improved compared to when drivers are instructed to use their own perception to achieve the same level of control. When following the ghost car, the driver is more effective than when instructed, but not as effective as with the CAN Coach with a lead vehicle. This suggests that CAN Coach is an HCPS, since remov.

Modern cars have *controller area networks* (CANs) that are in place to facilitate communication between *electronic control units* (ECUs) on the vehicle. CAN uses a multi-master data bus where all messages are multicast. There are dozens of systems which use the CAN, featuring hundreds of sensors, that support everything in the car from optimal engine function to climate control to driver safety. Year by year, more electronic features are added (e.g., adaptive cruise control, blind spot monitoring, lane keep assist). The amount of data produced on the CAN will continue to accelerate as cars become increasingly automated.

Considering the ubiquity of CAN technology and the holistic nature of the data it provides, there is substantial potential for finding insights about drivers under various levels of vehicle automation [7, 19]. Moreover, considering the real-time nature of the CAN and the increased number of sensors reporting data on it, it is now possible to implement CAN applications that augment safety, improve eco-driving, or adjust car following behavior.

Since CAN data is available widely within today's vehicle fleet, these applications could scale broadly and at low cost. The demonstrable value of CAN data is already established in commercial aftermarket applications. Insurance companies offer discounts on rates in exchange for a relatively reduced set of data on a driver available through the *on board diagnostics* (OBD-II) interface (a subset of the data available on the CAN) to characterize driver behavior and ultimately risk. Major car manufacturers now collect data on their vehicles, and collaborate on CAN data research [3, 4, 16, 21]. Like the rapid introduction of mobility applications that were created during the transition from traditional cell phones to smartphones, we expect a similar transition to occur on vehicles using data available on the CAN bus. The goal of this work is to illustrate such an application: an HCPS that is effective in controlling the time-space profile of an individual vehicle in traffic flow using only CAN data. Though there has been discussion of the potential of CAN data to support application development [7], there have not yet been studies published showing real-time CAN feedback to a human driver is effective in controlling speed and spacing on the roadway.

1.1 Contribution and Outline

The contribution of this article is a demonstration of an HCPS that is effective in controlling the time-space of an individual vehicle in a traffic flow. The CAN-based feedback system that is used (CAN Coach) enables drivers to achieve precise control tasks they could not otherwise achieve by giving extrasensory feedback informing adjustments of velocity and spacing. In other words, the CAN Coach can coach drivers to achieve 'superhuman' driving tasks, and enables human drivers to drive in a manner that could smooth traffic waves. Furthermore, the technology can scale since it is low-cost and leverages common vehicle technology.[MWN: reel me in, but I wanted to make sure it stands out]

The remainder of this article is organized as follows. In Section 2, we discuss works informing the formation of CAN Coach. In Section 3, we provide an overview of CAN data, the CAN messages used at run-time, the hardware platform, and the software platform. In Section 4, we describe the data processing required to transform CAN messages into relevant

[1]This work was conducted under IRB approval #200343

data that is used as feedback to the driver. The implementation of the auditory feedback is also discussed, and the description of the driving experiments is provided. Section 5 describes how the validity of the CAN data is assessed, and provides the analysis of the experiments conducted to determine the impact of CAN Coach on driver behavior. Finally, Section 6 outlines the planned extensions from our work presented here.

2 Related Work

This section discusses related work that informed decisions on the feedback style, the intersectional interest in time gap for traffic flow and safety, CAN-based applications, and real-time driving applications.

Our goal in this work is to give feedback to drivers that will precisely inform their driving behavior. Continuous feedback is more appropriate in this regard than discrete feedback. The work in [16] informs drivers in a simulator about the time headway of the car to promote appropriate reliance on automation and support effective transitions between manual and ACC control. In the study, drivers who are given continuous information about time headway are better at managing time headway and braking interventions. The work [6] provides continuous auditory feedback to the driver whenever the time gap is below one second. Because of this feedback, drivers spend less time driving at time gaps below or near one second. This work has success in preventing time gaps lower than their specific threshold. In contrast to [6] and [16], [14] gives drivers discrete feedback on their time headway. Human subjects placed in a driving simulator change their behavior in the correct direction, with error of 0.3 s to 0.8 s [14]. The discrete feedback cedes control of precision. Precision is important for making impacts on the greater traffic flow, like smoothing traffic waves.

This work features experiments designed to test the efficacy of an HCPS for time gap control, with an eye toward traffic wave smoothing, but control of speed and spacing can be applied to improve driving goals other than time gap control. A shortcoming of research in driver behavior intervention is a lack real-world implementation and continuous feedback. For example, eco-driving experiments done with behavioral training as the method of intervention [18] [PN: grammar]. The work [18] measures driver's eco-driving in simulated environments, then does offline training on eco-driving, and puts the driver back in the simulator to measure the effects of behavioral intervention. A shortcoming of the evaluations in [18] and other work using simulations is that they aren't assessed in the real world. An HCPS such as driving with CAN Coach could provide the extrasensory feedback needed to coach a driver on their eco-driving on real roadways.

[MWN: maybe cut this part and/or others to make Related Work more brief. have much more important stuff to say later.] Since auditory or tactile driving warnings correspond with better driver reaction times than visual warnings or no warning at all [15], we select an auditory feedback approach. Choosing an apt sound for auditory feedback is an important design consideration in auditory warning systems (e.g., as is already used in lane departure warnings and blind spot monitoring systems). The work [3] examines which sounds help in discrete intervention (i.e., a single warning when a minimum time gap threshold is violated) to stop dangerously close following. It investigates which sounds are informative but least irritating and therefore effective as a feature in production systems designed to increase safety for long-haul truckers without annoyance. Because [4] uses continuous ambient sounds to give truckers complete information on the state of the vehicle, e.g., lane position, it was determined through truck driver surveys that the lack of sound is the preferred mechanism to give feedback to drivers. In contrast to [3] which only aims to avoid very low time gaps and therefore delivers infrequent auditory feedback, our work aims to coach drivers affirmatively to follow a specific time gap. Our more precise aim comes at the cost that it makes sounds too often to be usable in a production-level system.

Time gap is the subject of research interest in both safety and traffic flow. The time gap is a quantity that can be readily computed from sensor data on a follower vehicle. The time gap is the *space gap* (distance between lead vehicle rear bumper and follower vehicle front bumper) divided by the velocity of the follower car. Time gap is a surrogate safety metric for cars [2, 8, 9], in which larger time gaps are associated with increased safety. In freeway driving, drivers drive at nearly half the recommended safe time gap based on human reaction time [11]. It is also relevant to note that the average time gap and total throughput of the roadway are inversely related, so it is important to prevent the average time gap from becoming too large. Accordingly, feedback to promote a safe and overall throughput-efficient time gap is desired.

Some research has been done in post-processing CAN data. The literature features machine learning work characterizing data traces from CAN to categorize emergent driving behavior like changing lanes, making turns, and identifying drivers [7, 19]. These analyses done in post-processing contribute to support the assertion that CAN data has immense value in understanding drivers.

Some research has been done with real-time processing of CAN data, but without interaction with the driver. The work [5] presents a driver behavior identification tool that fuses CAN data, an IMU, and GPS to classify normal and aggressive maneuvers in real time. They took no steps to intervene and change driver behavior, but this work demonstrates the potential of using sensor fusion at run-time to understand the driver.

Some research has been done interacting with the driver in real-time, but rely on a 2-vehicle V2V system and installation of a LIDAR on the following vehicle. The work [21] shows a real-time speed advisory system with the intent to show that human-driven vehicles can contribute to traffic shock wave mitigation. Their velocity-based feedback is shown to reduce the standard deviation of the distance between two vehicles by as much as 40.7%. [MWN: really want to emphasize that this work doesn't achieve what is done our paper] In this paper, we evaluate a complete HCPS that is internally self-reliant.

3 CAN Coach System Background

This section provides an overview of CAN data, the CAN messages used at run-time, the hardware platform, and the software platform. For this study the precision, accuracy, and recency of the data provided for feedback contribute directly to the performance of the CAN Coach. The ceiling of the potential performance is set by the quality of the soft real-time CAN data processing that are the basis of the driver feedback (i.e., bad sensor data or incorrect processing will provide inaccurate feedback to the driver).

To use the CAN Coach, depending on the control task, we need measurements of the velocity of the following vehicle, velocity of the leading vehicle, and the distance from the front of the following vehicle to the tail of the leading vehicle. By accessing the CAN data in the following vehicle, this information can be found via wheel encoder and radar sensors. Radar sensors are increasingly common on stock vehicles today. Those sensors directly measure information needed to compute the space gap and relative velocity, so the relative trajectories of the follower and the leader are evident in the data. To have relevant feedback in real-time, there are two related challenges with radar data. First, incoming radar data tracks many objects in a wide field of view and needs to be processed to extract a high quality estimate of the location and relative velocity of the lead vehicle. Second, any processing on the radar data must be done quickly so that the feedback to the driver remains relevant. Latent data could substantially degrade the quality of control achievable using CAN Coach.

3.1 Collecting CAN Data

Each message recorded from the CAN includes a timestamp of record, the bus the message was sent on, the data signal(s) contained within the message, and the length of the data in bytes, which is critical for proper decoding (i.e., translating into human understandable values).

A sample of this data is shown in Table 1. From this data, we can understand what the vehicle's sensors are reporting after translation. The address gives the context to decipher the information embedded in the hexadecimal message. A separate *CAN database* (referred to as a DBC) provides information and instruction on how to read CAN signals. The DBC has stringent formatting rules that define the messages, the signals subset of the message, the start, size, scale, offset, min, max, units, endian-ness, and signed-ness of the message. Some messages are multiplexed too. The DBC is critical for successful decoding of the CAN data, since any errors in the DBC result in errors or complete failure of the ability to interpret the information contained in the CAN message.

Table 1. Example of CAN Messages that must be deciphered and translated to build CAN Coach.

CAN Messages				
Time (s)	*Bus*	*Address*	*Signal*	*Length (B)*
3.048295	0	945	0000...00ca	8
3.048318	1	865	8053...5c7c	8
3.048447	1	384	a8ff...0028	8
...	...	...	...	...

Table 2. Summary of CAN Messages Used for CAN Coach at run-time.

Message Name	*Signals*	*Frequency (Hz)*
Raw Radar 1	Position, Rel.Vel.	20
Raw Radar 1	Position, Rel. Vel.	20
...	...	...
Raw Radar 16	Position, Rel. Vel.	20
Pre-filtered Radar	Space Gap	5
Velocity	Speed	40

3.2 CAN Signals at Run-time

This section details the CAN signals that are used at run-time, summarized in Table 2. Many types of velocity signals are reported on the CAN. Each wheel reports an associated velocity, the speedometer has an associated velocity, and the driver support unit reports a velocity. In this work we use the velocity corresponding to the speedometer because it reflects the movement for the vehicle as a whole, and is reported for reference by other vehicle systems on multiple busses.

The radar data can inform the vehicle, and the CAN Coach, about the conditions ahead on the roadway, but is also the most challenging to interpret. The data is reported on a set of 16 *tracks*, each of which may contain an object in the field of view of the sensor. Each track reports the Cartesian (x,y) coordinates of the object, the relative velocity of the object with respect to the vehicle, the relative acceleration, a binary validity signal, and a score ranging from 0-100. A useful pre-filtered estimate of the lead vehicle's space gap is reported on the CAN, but it doesn't contain the relative velocity signal and is rounded down to the nearest meter. Processing this collection of data to determine the distance and relative velocity to the vehicle immediately ahead, critical for the CAN Coach, is explained later in Section 4.

3.3 Hardware

This section describes the hardware platform used. The base vehicle used in this work is a stock 2020 Toyota Rav4 Hybrid vehicle. The vehicle has as standard equipment an *Adaptive Cruise Control* (ACC) system and *Lane Tracing Assist* (LTA) system. The ACC system depends on a stock forward looking radar unit that transmits relevant data on the CAN. To access this data, we used a Gray Panda manufactured by Comma.ai as a data logging device. The Gray Panda is connected to a Raspberry Pi 4 via USB, where messages are decoded, processed, recorded, and used to generate an auditory feedback for the driver.

3.4 Software

In this section, we discuss software that was developed to implement CAN Coach system, execute driving scenario in a repeatable manner and perform analysis of the captured data. Figure 3 shows how the information flows through the system in a loop . [RB: Libpanda is recording all of the incoming data, how ROS pares it down to what is needed for CAN Coach, and what the CAN Coach does. Discussion convincing the reader that the software is polished, reliable, and scaleable.]

3.4.1 Libpanda Layer. The Comma.ai Panda devices are simply interfaces to allow for communication to the CAN bus through USB. Some variants of the Pandas also has a GPS module installed. Libpanda is a C++ based library to aid in the construction of custom software, abstracting the USB interface for easy reading and writing to the CAN bus and for easy reading of the GPS module [1]. Comma.ai also provides support software for their hardware, however their focus has been mainly focused on on self-driving applications and is intended to be run on particular computers and operating systems. For their purposes, a Python-based library is effective for their use case. In early testing on a Raspberry Pi 4 with Ubuntu, it was found that the Comma.ai Python library would miss nearly 50% of the CAN messages during logging. This performance is what lead to the initial construction of libpanda, to operate effectively with no drops in CAN packets during data logging.

The core of libpanda is dependent on libusb. In this regard libpanda acts as an adapter with libusb serving as a lower-level layer for handling the USB protocol. Defined in the library are various methods of configuration for the Panda devices. In order to prevent packet drops from filled buffers in the Panda device itself, libpanda is built with multi-threading in mind to ensure asynchronous data reading and transfer. To ensure easy use for software developers the threading is handled by the library itself, and data can be read through a methodology of callback functions in the form of the observer software design pattern. This observer pattern works for both GPS and CAN messages to reduce latency in applications.

Though libpanda is intended to be compiled into custom applications, a few simple example utilities are provided. One of these utilities dumps all data from the GPS and CAN bus to CSV-formatted data files. Another utility will set the system clock to align with GPS time. Libpanda also has installation scripts with the intention of installation on Ubuntu can install service scripts for automatic data collection. This means that only one script is needed to perform automatic time synchronization and data collection. Having such scripts greatly aids non-experts in getting started with vehicle data collection.

3.4.2 ROS Layer. The Robotic Operating System (ROS) is an open source framework for robotics. It provides the necessary tools and libraries to create and run peer-to-peer processes called nodes. ROS enables modular software development by utilizing reusable code packages. In ROS, the node is executable code designed to perform a specific task and is capable of communicating with other nodes at runtime. Nodes can communicate with each other as a result of publish and subscribe protocol. In this protocol, nodes need to register their information to ROS master, which functions much like DNS server, and then nodes utilize the ROS master to locate other nodes. When a node is in need of data, it must subscribe to relevant topics while a node that produces data publishes messages to a topic [13].

In this work, the ROS layer has the following main functions: utilize libpanda library for data acquisition, decode and publish relevant CAN bus messages on ROS topics, implement CAN coach system, and log and playback data. By taking advantage of the modular structure of ROS, each of these main functions is implemented with one or more nodes to minimize the code that goes into each node and make the debugging process easier. The workflow of the ROS system begins with a node that is integrated with libpanda library performing real-time reading and logging of raw CAN bus data and publishing of relevant messages, such as speed, acceleration, etc, to a topic. Next, another node is receiving the published raw data to process it by decoding it based on Toyota RAV4 DBC file and publishing the results to relevant topics. After decoding CAN bus data, nodes implementing the CAN coach system receive the processed CAN bus data needed for the system to function. In the experiment, the CAN coach system consists of CAN coach node, mode changer node[MWN: note: should we explain this?], ghost mode, and director node. The CAN coach node is core node in the CAN coach system where it processes CAN data and generates feedback. Mode changer node, as the name indicates, allows the driver to change from one test to another by toggling the high beams in the car. The ghost mode is where we use historical data to create virtual car, and the director node facilitates the communication between the driver, mode changer, and CAN coach. The ROS framework has various tools and features that assisted in developing, testing, and analyzing the whole ROS system. With ROS, we are capable of recording the system's data at run-time into a single file and play back the recorded data to simulate the behavior of the running system. This feature allows for recreating and analyzing the experiment without the need of a vehicle and other physical hardware.

3.4.3 Ghost Mode. The idea behind the ghost mode is to utilize recorded, or even fabricated, driving data to create a virtual (ghost) car. Then, the CAN coach can provide feedback to the driver to follow the virtual car. For the ghost mode to work, the CAN coach requires the space gap between the ego car and the ghost car; this can be achieved by keeping track of the distance traveled by the ghost car and the ego car (physical 'following' car). Since we do not have a CAN bus message for the distance traveled, we designed a ghost mode package in ROS that consists of two integrator nodes to produce distance traveled for each car, i.e the ego car and virtual car, using speed messages. Since the ghost car isn't governed by reality, the ego car can get really far ahead or behind due to road conditions, e.g. dense traffic or an acute traffic event. If the space gap gets really far off, like hundreds of meters, the usefulness of the remainder of the test is compromised. In order to solve this issue, we equipped the ghost car's speed integrator with a reset feature which gets triggered if the distance between the ghost car and the ego car is more than 100m (too far to follow) or if the distance between the two cars is less that -30m (dangerously close)[MWN: need to figure out a sane way to explain why we chose to do it this way]. During a reset, the ghost car distance traveled is set equal to the ego distance traveled.

The CAN Coach calculates the time gap in ghost mode from the ego velocity, and the difference in the distance traveled for both vehicles during ghost mode plus a constant of 65 meters. The constant is added because at the initialized speed, 29 m/s, and time gap set point, 2.25 s, the target space gap is 65 m. This ensures that in the first minutes of ghost mode, if the ghost car is reset it will return to the target state.

3.4.4 Director Node. For the precise, repeatable experiment with fixed duration for various segments of CAN Coach exercise, we developed a state-machine in conjugation with publish-subscriber mechanism of ROS. A schematic of the director node is provided in Figure 2. The director node consists of a subscriber that subscribes to changes in the mode changer. Mode changer allows user to asynchronously change the mode if a user wants to switch modes. A detailed discussion of modes is provided in Section 4.2.2 and Section 4.2.3 where broader definition of modes includes *Normal Driving*, *Instructed Driving*, *Coached Driving*, and *Ghost Mode* and three separate protocol namely *Velocity Matching*, *Constant Time Headway*, and *Dynamic Time Headway*. We used Stateflow/Simulink in MATLAB to implement the state-machine that allows to switch modes synchronously in time-triggered and event-trigerred manner. A change is triggered based on input from the user via mode changer or in a time-triggered manner. This allows the drivers to determine for how long to drive in a specific mode. For real-time implementation, overall Stateflow/Simulink block was used to generate a native C++ ROS node that runs along side other ROS node.

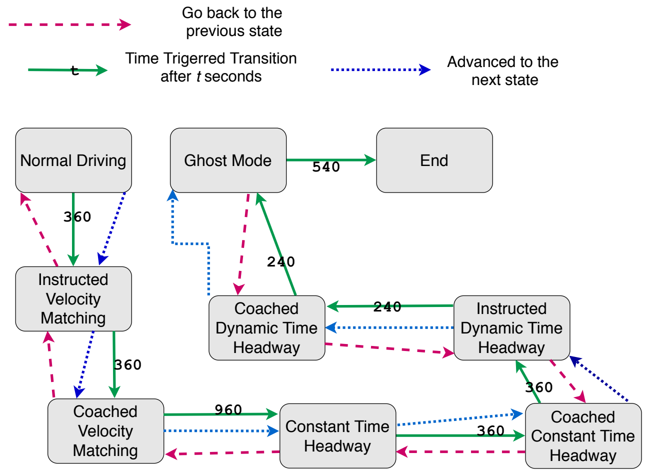

Go back to the previous state
t
Time Trigerred Transition after t seconds
Advanced to the next state
Normal Driving
Ghost Mode
End
540
360
240
Instructed Velocity Matching
Coached Dynamic Time Headway
240
Instructed Dynamic Time Headway
360
360
Coached Velocity Matching
960
Constant Time Headway
360
Coached Constant Time Headway

Figure 2. The director node with different experimental conditions + experiment protocol as separate states. Each state lasts for certain duration. Alternatively, state can be switched based on mode changer input that triggers state switching. The director node informs CAN Coach node about the set point, feedback type, current mode every Δt time step where Δt is the sampling time of the director node. Discussion about each state is provided in greater detail in Section 4.2.2 and Section 4.2.3. We used sampling time $\Delta t = 0.5s$.

3.4.5 CAN Coach. The CAN Coach, the eponymous ROS node, is what links the driver to the vehicle sensors by giving the driver audio feedback derived from CAN data in real time. The CAN Coach subscribes to the CAN data produced from the Section 3.4.2: ROS Layer, the ghost vehicle data from Section 3.4.3: Ghost Mode, and information on the control task and feedback type from Section 3.4.4: Director. With this information, the CAN Coach does two things.

1. Solve the data association problem between pre-processed radar data, and the unprocessed raw radar data that has relative velocity measurements. This is
2. Produce sound for the driver corresponding to the current control task and feedback type at the given time in the experiments.

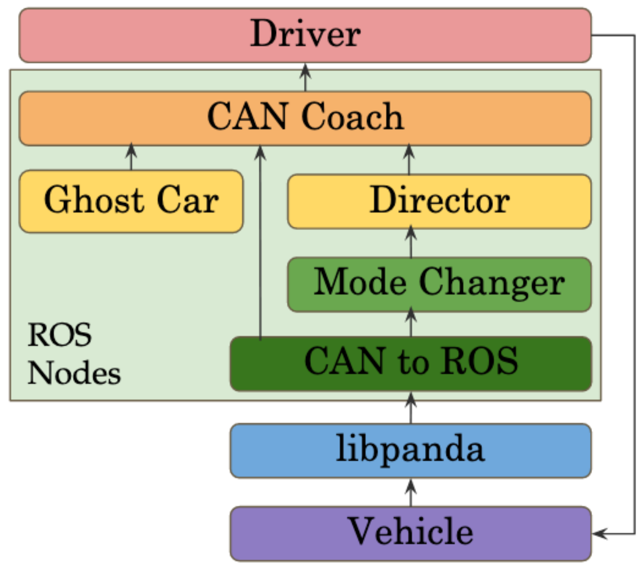


Figure 3. Diagram showing how data flows from the vehicle through the can coach system to the driver and loops.

[RB: Add a flow diagram showing how data flows all the way from CAR to libpanda, ROS Layer, mode changer, director node and CAN Coach][MWN: good idea, added it]

4 Methodology

The *Methodology* section covers the decisions made to design the feedback from the CAN Coach, and to design the experiments.

4.1 Feedback Design

The CAN Coach has simple feedback, because complexity can be a burden to the driver [4]. Specifically, when the driver is a threshold amount away from the target, a signal is output from the controller to inform the driver of the car's state. If time gap is outside of the set point ±0.05s, a sound is output to increase or shorten the time gap. If velocity matching is outside of the set point ±0.4 m/s, a sound is given as output to increase or decrease the velocity. There are only three discrete options for a human driver controlling a car following another vehicle: speed up, slow down, or do nothing. High pitch feedback is given to indicate the time gap is high and the driver should speed up to shorten the time gap, or just to speed up to match the lead vehicle velocity. Low pitch feedback is given to the driver to indicate the time gap is low and the driver should slow down to increase the time gap, or just slow down to match the lead vehicle velocity.

4.2 Experimental Design

4.2.1 Driving Baseline. To provide a baseline, each driver starts with *Normal Driving*. Normal Driving is regular driving with no prompt for time gap. The driver is given instructions to follow the vehicle ahead without changing lanes. As a safety precaution, the driver is alerted that the vehicle ahead may change speeds during the test. This establishes a baseline in absence of instruction and feedback. Some evidence shows that a mild cognitive task (such as driving for an experiment) induces longer time gap [12], so the time gap behavior shown experimentally likely represents a longer time gap than one observed candidly.

4.2.2 Feedback Type.

- *Instructed Driving*: The driver is given all information provided during Normal Driving. Additionally, they are asked to match velocity, or maintain a certain time gap to the vehicle ahead. This instruction establishes a baseline in absence of CAN feedback with which to compare the CAN feedback.
- *Coached Driving*: The driver is given the same instructions as Instructed Driving. Additionally, CAN Coach audio feedback is activated when driving. The driver is informed that high pitches indicate the time gap is too large or they need to speed up, and low pitches indicate the time gap is too small or they need to slow down. No sound indicates no adjustment is necessary.
- *Ghost Mode*: The driver is given the same instructions and feedback as for Coached Driving, but there is a ghost vehicle to follow instead of a real one. The drivers are simply informed to follow the feedback; instruction to maintain a 'time gap' to a vehicle that you can't see does not make sense.

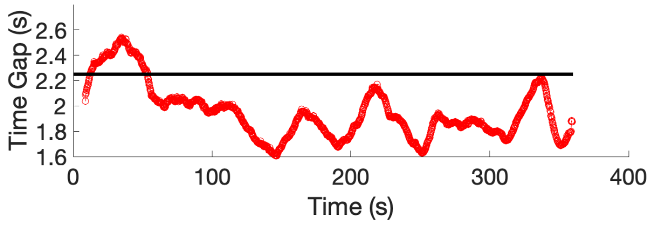
4.2.3 Control Tasks. The driver in following vehicle was asked to perform a series of control tasks with and without CAN Coach feedback. They are as follows:

- *Velocity Matching*: The driver matches the velocity of the leading vehicle. The CAN Coach gives feedback to guide the velocity profile of the driver for controlling the ego vehicle.
- *Constant Time Headway*: The driver matches a 2.25 s time gap. The CAN Coach gives feedback guiding the time gap, i.e. the space gap as a function of velocity, to the driver for controlling the ego vehicle.
- *Dynamic Time Headway* The driver matches a 2.25 s or 1.8 s time gap when indicated. The CAN Coach gives feedback guiding the time gap, i.e. the space gap as a function of velocity, to the driver for controlling the ego vehicle. With a varying set point, the spacing specified by the CAN Coach is even more varied.

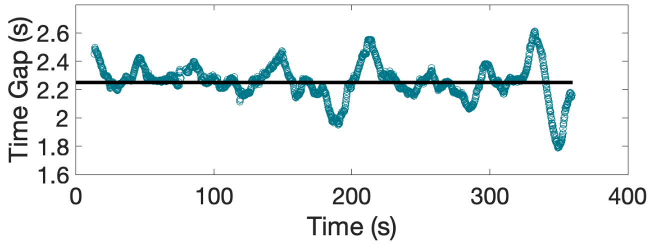
4.2.4 Details on the Field Experiments. The CAN Coach Experiments were a set of 2-vehicle driving tests that took place on the roadways in the Nashville, TN area. They consisted of a following vehicle with a driver testing the effects of the CAN Coach, and a leading vehicle with a driver to provide consistent and cross-comparable driving behavior. Each drive started and ended in approximately the exact same location, and occurred in the middle of the day, in an effort to produce as close as possible to identical input to each driver.[MWN: this is where the GPS figure will go]

Whenever instruction was needed for the following driver, it was provided on time and automatically with identical language from the Director, detailed in Section 3.4.4. Each driver was alone in their vehicle. Before the experiments, drivers were instructed to listen to instructions from the vehicle and follow to the best of their ability.

The lead vehicle driver was given a simple driving protocol: set the cruise controller to 29.0 m/s (65 mph) and allow the hilly terrain, among other factors, to vary the speed profile. This is enough variation of speed to require velocity and spacing adjustments to follow a specified time gap. In Ghost Mode, the Ghost Car is going a constant velocity, 29.0 m/s (65 mph), in the duration of the Ghost Mode.



(a) Time gap from Instructed Driving. Set point is horizontal line. [DBW: need to be more specific. What driver? What test? Units on x and. y axis? suggest redrawing all plots later for consistency]



(b) Time gap for Coached Driving. Set point is horizontal line.

Figure 4. A comparison of the time series data, showing time gaps for instructed driving and coached driving.

5 Results

First, we show that the CAN Coach is effective in getting a driver to closely follow a constant time gap. No matter how close or far the driver is from the mean when in Instructed driving, the driver is very close to the specified time gap when Coached. For every driver, the standard deviation of the time gap decreased. Second, we show that on average when comparing coached driving to ghost mode the mean time gap error increases, and so does the standard deviation of the time gap. Then, we consider some alternative implementations of CAN Coach and see if they are also effective in enabling a driver to match targeted velocities and space gaps. Velocity matching proved to be quite ineffective in getting drivers to be consistent in their space gap. Finally, we tested an extension of the constant time gap feedback; the set point of time gap is alternated every minute, so each change in time gap setting changes the target space gap at a given velocity. Though performance was slightly worse than when the set point remained static, the results mimic the outcome from the static constant time gap feedback. The mean time gap error is reduced for all drivers, and the standard deviation of the time gap is diminished on average.

5.1 Data Validity

5.1.1 Speed. The CAN velocity measurements corresponding to the speedometer are precise to 4+ significant figures [DBW: no clue what this means. How about x m/s or something human readable]. The speed measurements from the CAN bus are shown to be valid by comparing the integral of the velocity data with a cumulative distance travelled between data points from a high-precision GPS sensor used for validation. There was a 0.01% difference in the integrated velocity from the GPS distance travelled measurement. [DBW: no idea how you reason about error on velocity by the total distance. i.e., how do you know it is valid to 4 sig figs if you only integrate the signal?]

5.1.2 Space Gap. CAN radar messages are used to measure space gap accurately in [20], which are also confirmed with high-precision GPS measurements. Since these experiments were performed with a different year-model vehicle, here we take additional steps to confirm the space gap measurements are accurate. Ten measurements are made in approximately one meter intervals from 0-10 meters when the vehicle is at rest and an object is ahead at the set distance. All of the CAN measurements match the true object distance empirically measured with a tape measure to within +/- 2 centimeters. This confirms that the measurements were accurate and precise.

5.2 Constant Time Gap

The constant time headway test shows the impact of using CAN Coach feedback to enable the driver to meet a specific speed and spacing with precision. [DBW: i guess this section needs a setup, it wasn't clear to me you are going to compare instructed vs coached driving under the goal to reach a desired time headwday]

5.2.1 Effect of CAN Coach on Driver A. First, we examine the results from a driver representative of the set of all drivers. [DBW: is A actually representative? if not, just say we first describe the results of cancoach on a single driver.

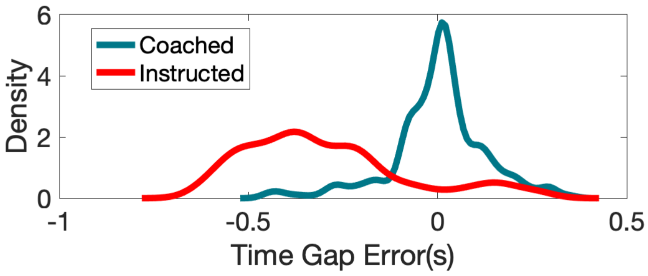


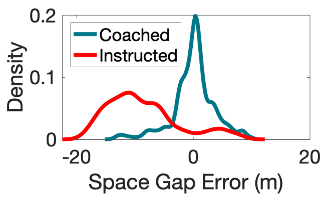
Figure 5. Kernel density estimate of time gap values for instructed driving and coached driving.

No explanation needed unless it really is representative. and in what sense?] Drivers simply instructed to maintain a 2.25 second time gap struggled to meet the set point. [DBW: exactly. Here you are comparing one driver under two different settings (instructed vs coached, but you didn't tell me that] Figure 4 shows a representative time series of time gap data from segments of Driver A's drive. Driver A's time gap varies during instructed driving, as shown in Figure 4a, however when given CAN Coach feedback (Figure 4b) the drivers' time gap hovers around the set point. Figure 5 shows the distribution of the time gap for Driver A during instructed and coached driving.

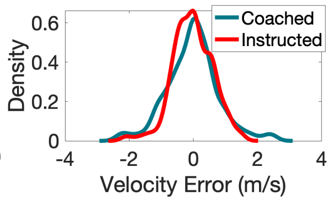
This is the same data presented in Figure 4, but in the form of a kernel density estimate (KDE) [DBW: of what? My suggestion is to just refer to these as the distribution of the time gap error, space gap error, etc. KDE is just a method you are using to plot the error distribution]. The distribution for instructed driving has wide variation and is not centered on the set point. In contrast, during Coached driving the time gap distribution is centered on the 2.25 s set point, and much more narrowly than in instructed driving. This shows that the CAN Coach is effective in getting Driver A to closely follow a velocity and spacing profile. To get a deeper look at relative importance of velocity and spacing for time gap control, we can look at the KDEs for the velocity error and spacing error. Figure 6 shows the time gap distribution broken down into its components. From comparing 6a and 6b we can see that though there are differences in relative velocity between instructed and coached driving, the patterns in space gap control are what dominate the time gap control results.these need to be recreated for Driver A [DBW: figure 4 needs to be cited /discussed here, right?]

5.2.2 Effect of CAN Coach on All Drivers.

Figure 7 shows the results comparing instructed driving to coached driving for all drivers. Figure 7a presents all 6 drivers' instructed driving time gap error[MWN: this plot isn't error]. The drivers' only tool to achieve the task is their ability to estimate by eye. It's clear from Figure 7a that the Instructed drivers vary greatly in their performance. Mean time headway in this condition ranged from 1.95 s to 3.61 s. The mean time



(a) KDE of Space Gap error.



(b) KDE of Relative velocity.

Figure 6. Comparing the contribution spacing control and velocity matching to the success of the time gap control task. [DBW: soemwhere in the paper we need to relate speed time headway and space gap. We need a figure and math]

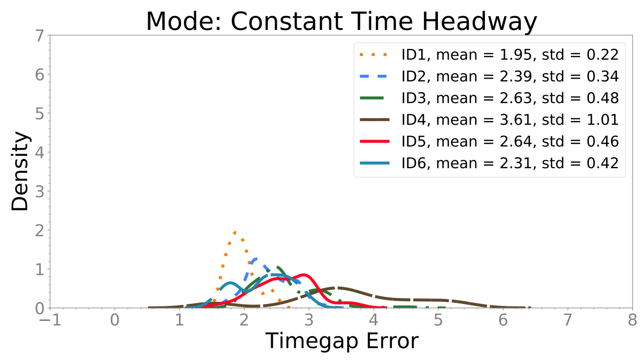
gap error is 0.44 s, with 0.43 s standard deviation of the mean error from driver to driver. In other words, on average the following drivers were 12.7 m from the target time-space profile [DBW: cool. but as a reader i have no clue how this conversion occurred], with an average standard deviation of 14.2 m. Essentially, the average driver is averaging 2 car lengths off and just as likely to be exactly on target as being 4 car lengths off of the desired time-space profile. [DBW: Figure 5 b is pretty impressive, we'll need to figure out how to drive this point home]

Figure 7b presents all 6 drivers' coached driving time gap error. No matter the mean error or variance in 7a, with the CAN Coach the all 6 drivers' time gap error is centered close to 0 with less variance. The mean time gap for coached drivers ranges only from 2.21 s to 2.35 s. This alignment is starkly clear in Figure 7b. The mean time gap error is 0.05 with 0.03 standard deviation of the mean error from driver to driver. In other words, the average coached driver is 1.4 m from the target time-space profile, with an average standard deviation of 5.6 m. Essentially, always within a car length of the desired time-space profile. [DBW: maybe a table with driver and percent reduction in mean error and percent reduction in std could help drive teh point home]

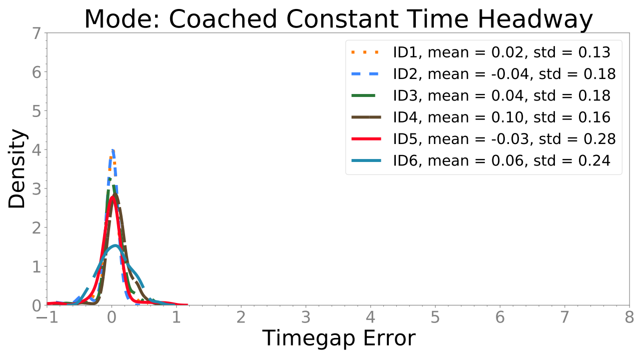
5.2.3 Comparing Driving Conditions.

[MWN: this subsection can be sucked into the above subsections, much is redundant] [DBW: yes, this section confuses me. 3 paragraphs in and i'm struggling to understand the point] Driving with CAN Coach clearly makes a difference in achieving the control aim. The feedback from the CAN Coach is effective at guiding the driver to a specific velocity and space gap behind the lead vehicle. In Figure 4b it's [DBW: general rule, avoid ' when possible] clear that the driver being coached hovers narrowly around the set point from CAN Coach, whereas in 4a the driver simply instructed has a time gap that wanders somewhat aimlessly in the general vicinity of the instructed set point. [DBW: too informal, we rephrase later]

Regardless of a driver's performance in the Instructed driving, with CAN Coach all drivers' performance in closely



(a) Estimated distribution of time gap values for instructed.



(b) Estimated distribution of time gap values for coached.

Figure 7. Comparing performance across all drivers with just instruction and CAN Coach feedback.

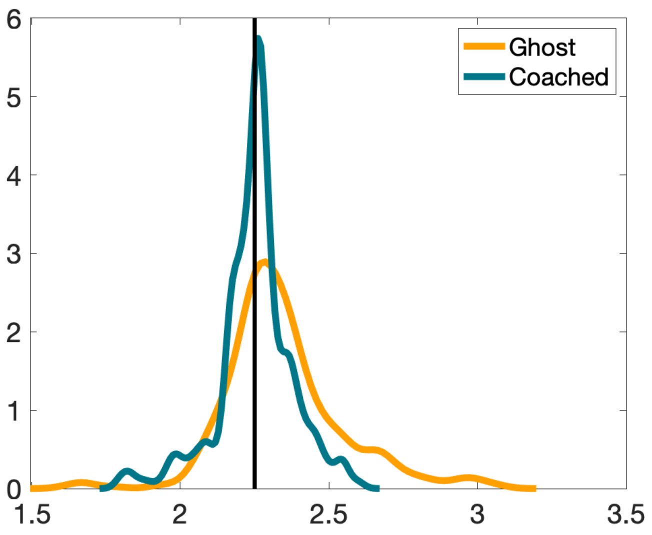


Figure 8. Example of ghost mode time gap with CTH time gap for comparison.[DBW: this caption is confusing. ghost mode is a feedback mechanism, cth is a control goal]

following the velocity and spacing communicated from CAN Coach feedback improves in the Coached driving condition.

Figure 6 is a component breakdown of the in Figure 4 that shows the space gap precision is what really drives the improvement in the driver's ability to follow the time gap. The comparison of relative velocity isn't remarkably different, but the comparison of the space gap is. This makes sense in the context of an HCPS. The human driver following the car ahead has fairly strict bounds on their velocity because they want to stay behind the car ahead, and of course never crash into the vehicle.

5.3 Reduced Performance in Ghost Mode

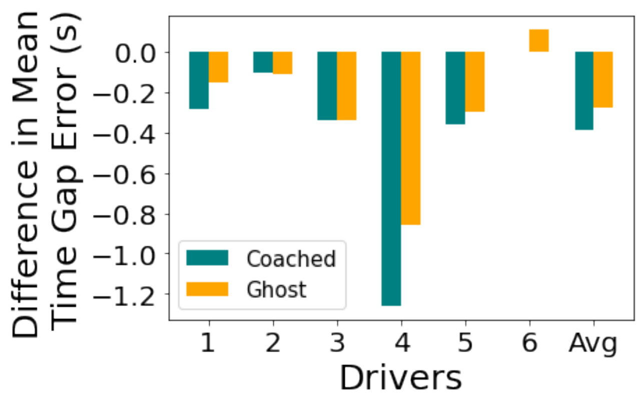
For the CAN Coach system to be an HCPS, there must be a contribution from the human driver to the effectiveness of the CAN Coach. If this is true, the expectation is that when removing the human sensory contribution from the perception loop, the effectiveness of the CAN Coach would decrease. We consider the ghost mode test an 'ablation study' because it removes the human perception of the lead vehicle by replacing it with an abstract 'ghost' vehicle. The ghost vehicle gives feedback to the driver in the same way a real vehicle would, but the driver can't see it.

Figure 8 shows that in ghost mode, Driver A goes a worse job of staying close to the set point compared to with Coached Driving. Going from coached to ghost, the mean is slightly further from the set point, and the standard deviation greater.

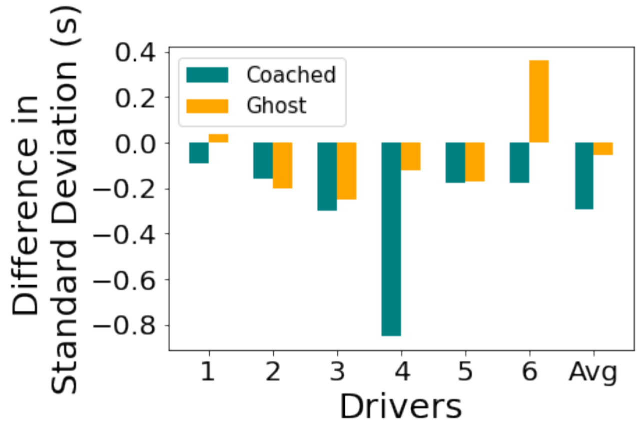
Figure 9 shows there is much variation from driver to driver, but across all drivers the trend from 8 holds. Figure 9a shows the change in the mean time gap error by driver, and comparing Instructed, Coached, and Ghost Mode driving. Positive values means that the mean error increases from X to Y driving condition, and negative means mean error decreases. On average, both coached driving and ghost mode driving have less mean time gap error than instructed driving, and coached has less error than ghost. The mean time gap error across all drivers is 0.05s during Coached Driving, compared to 0.13 s in Ghost Mode.

This theme continues in Figure 9b. Positive values mean that the standard deviation increases from X to Y driving condition, and negatives mean standard deviation decreases. On average, both coached driving and ghost mode driving have less standard deviation than instructed driving, and coached driving has less standard deviation than ghost. The mean standard deviation of Coached Driving is 0.19 s, compared to 0.40 s in Ghost Mode.

Figure 9 shows us that Ghost Mode has a similar effect on the drivers as Coached Driving with a real car ahead, just with a diminished magnitude.



(a) mean error of ghost



(b) stdev of ghost

Figure 9. mean error of ghost

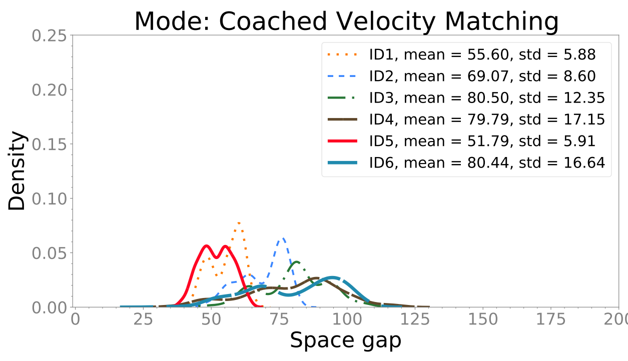


Figure 10. Velocity matching lacks control of space gap. [DBW: I like this plot but it confuses me]

5.4 Velocity Matching: Not Effective

[MWN: the point of this section is the show the reader that velocity matching isn't useful for time-space control of a vehicle because the spacing is unspecified]

There is a moderate reduction in standard deviation of the relative velocity for Coached Driving over Instructed Driving (CAN Coach feedback reduces mean relative velocity by 0.03 m/s, and reduces standard deviation by 0.16 m/s). This shows that the CAN Coach has an intended effect in the driving task: help match velocity. [DBW: more like matchign time headway also has the result of matching velocities]

Figure 10 shows that considering space gap distributions across drivers, velocity matching feedback clearly doesn't work to have fine control of space gap.

The CAN Coach giving feedback for velocity matching does not control the value of space gap. CAN Coach for Velocity Matching does encourage maintaining whatever your current space gap is, regardless of its value and your velocity.

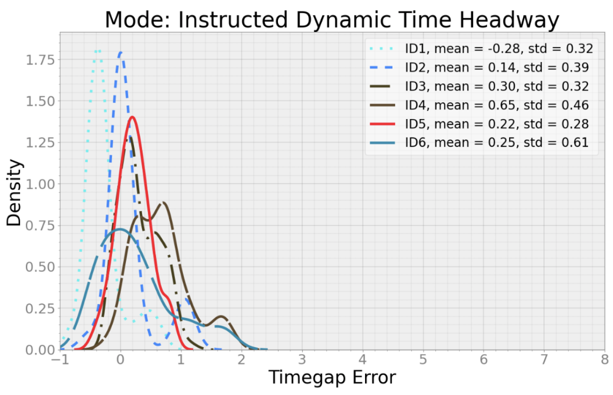
5.5 Dynamic Time Gap Settings

Drivers were instructed to maintain a 2.25 or 1.8 second time gap, with the set point changing on an automated timer from the Director node, according to experimental protocol. The drivers are enabled to closely follow the target speed and spacing with feedback from the CAN Coach. This is shown in Figure 11, which presents results similar to the results from coaching in 7. From 11a to 11b for each driver the mean error is decreased and across all drivers the standard deviation is much smaller. The mean time gap error is decreased from 0.21 s to 0.08 s when going from instructed to coached driving with dynamic time gap settings. The mean standard deviation is decreased from 0.40 s to 0.30 s when going from instructed to coached driving with dynamic time gap settings. Just as with the static constant time headway, the driver is enabled to closely follow the velocity and spacing even though the set point of the time gap is changing every minute. When comparing dynamic time gap settings to the static coached driving, we see performance was slightly worse than when the set point remained static. This is expected for a more difficult driving task. The mean time gap error increased from 0.05 s to 0.08 s when changing to time gap settings. The mean standard deviation increased from 0.19 s to 0.30 s when changing the time gap settings.

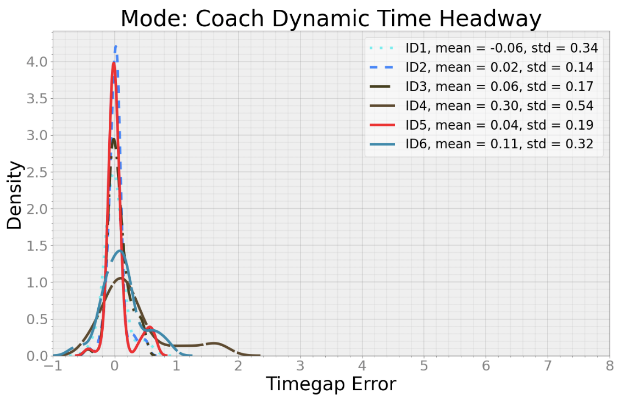
5.6 Driver Comparisons

[MWN: I don't know if I have anything interesting to say on this topic, so might just cut it.]

There are differences from driver to driver which hold throughout the experiments from test to test. Driver D has a higher variance than other drivers in the CAN Coach Experiments. Though the number of drivers is low, there seems to be a correlation between the mean in Normal Driving, and the mean in the drivers' estimation of a 2.25 second time headway. Across the board, Driver X performed best with the CAN Coach. Driver X's mean difference from the time



(a) Velocity matching relv KDE comparing instructed vs. feedback.



(b) need a caption.

Figure 11. Put a real caption here. [RB: Did we not decide that we are gonna put figures with same y-axis limits? I had provided figures with same y-axis limits later.]

headway set point was X.XX s, in comparison to X.XX s which is the average among all drivers.

6 Conclusions and Future Work

7 Acknowledgements

[MWN: Need to make sure everyone's mentioned their funding sources here.] This material is based upon work supported by the National Science Foundation under Grant No. CNS-1837652 (Work). This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Vehicle Technologies Office award number CID DE-EE0008872 (Nice). The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

[1] [n.d.]. *Libpanda: A software library and utilities for interfacing with vehicle hardware systems.*

[2] Vittorio Astarita, Demetrio Carmine Festa, Vincenzo Pasquale Giofrè, and Giuseppe Guido. 2019. Surrogate safety measures from traffic simulation models a comparison of different models for intersection safety evaluation. *Transportation research procedia* 37 (2019), 219–226.

[3] Pavlo Bazilinskyy, Pontus Larsson, Emma Johansson, and Joost CF de Winter. 2019. Continuous auditory feedback on the status of adaptive cruise control, lane deviation, and time headway: An acceptable support for truck drivers? *Acoustical Science and Technology* 40, 6 (2019), 382–390.

[4] Pavlo Bazilinskyy, Jork Stapel, Coert de Koning, Hidde Lingmont, TS de Lint, TC van der Sijs, FC van den Ouden, F Anema, and JCF de Winter. 2018. Graded auditory feedback based on headway: An on-road pilot study. *Varieties of interaction: from User Experience to Neuroergonomics* (2018).

[5] Juan Carmona, Fernando García, David Martín, Arturo de la Escalera, and José María Armingol. 2015. Data fusion for driver behaviour analysis. *Sensors* 15, 10 (2015), 25968–25991.

[6] Stephen H Fairclough, Andrew J May, and C Carter. 1997. The effect of time headway feedback on following behaviour. *Accident Analysis & Prevention* 29, 3 (1997), 387–397.

[7] Umberto Fugiglando, Emanuele Massaro, Paolo Santi, Sebastiano Milardo, Kacem Abida, Rainer Stahlmann, Florian Netter, and Carlo Ratti. 2018. Driving behavior analysis through CAN bus data in an uncontrolled environment. *IEEE Transactions on Intelligent Transportation Systems* 20, 2 (2018), 737–748.

[8] Carl Johnsson, Aliaksei Laureshyn, and Tim De Ceunynck. 2018. In search of surrogate safety indicators for vulnerable road users: a review of surrogate safety indicators. *Transport Reviews* 38, 6 (2018), 765–785.

[9] Aliaksei Laureshyn, Åse Svensson, and Christer Hydén. 2010. Evaluation of traffic safety, based on micro-level behavioural data: Theoretical framework and first implementation. *Accident Analysis & Prevention* 42, 6 (2010), 1637–1646.

[10] Amara Loulizi, Youssef Bichiou, and Hesham Rakha. 2019. Steady-state car-following time gaps: an empirical study using naturalistic driving data. *Journal of advanced transportation* 2019 (2019).

[11] John-Jairo Martinez and Carlos Canudas-de Wit. 2007. A safe longitudinal control for adaptive cruise control and stop-and-go scenarios. *IEEE Transactions on control systems technology* 15, 2 (2007), 246–258.

[12] Sanna M Pampel, Samantha L Jamson, Daryl L Hibberd, and Yvonne Barnard. 2018. Old habits die hard? The fragility of eco-driving mental models and why green driving behaviour is difficult to sustain. *Transportation research part F: traffic psychology and behaviour* 57 (2018), 139–150.

[13] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, Vol. 3. Kobe, Japan, 5.

[14] Malte Risto and Marieke H Martens. 2014. Supporting driver headway choice: The effects of discrete headway feedback when following headway instructions. *Applied ergonomics* 45, 4 (2014), 1167–1173.

[15] JJ Scott and Robert Gray. 2008. A comparison of tactile, visual, and auditory warnings for rear-end collision prevention in simulated driving. *Human factors* 50, 2 (2008), 264–275.

[16] Bobbie D Seppelt and John D Lee. 2007. Making adaptive cruise control (ACC) limits visible. *International journal of human-computer studies* 65, 3 (2007), 192–205.

[17] Raphael E Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, Hannah Pohlmann, Fangyu Wu, Benedetto Piccoli, et al. 2017. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C* 89, April 2018 (2017), 205–221.

[18] Mark JM Sullman, Lisa Dorn, and Pirita Niemi. 2015. Eco-driving training of professional bus drivers–Does it work? *Transportation Research Part C: Emerging Technologies* 58 (2015), 749–759.

[19] Toshihiro Wakita, Koji Ozawa, Chiyomi Miyajima, Kei Igarashi, Katunobu Itou, Kazuya Takeda, and Fumitada Itakura. 2006. Driver identification using driving behavior signals. *IEICE Transactions on Information and Systems* 89, 3 (2006), 1188–1194.

[20] Yanbing Wang, George Gunter, Matthew Nice, and Daniel B. Work. 2019. Estimating adaptive cruise control model parameters from on-board radar units. arXiv:1911.06454 [stat.AP]

[21] Hao-Ping Wu, Wen-Hsuan Shen, Yu-Lin Wei, Hsin-Mu Tsai, and Qianyan Xie. 2016. Traffic shockwave mitigation with human-driven vehicles: is it feasible?. In *Proceedings of the First ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services*. 38–43.