

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent of:      Koebler et al.  
U.S. Patent No.:      7,925,426                      Attorney Docket No.: 49649-0018IP1  
Issue Date:              April 12, 2011  
Appl. Serial No.:      11/283,137  
Filing Date:             November 17, 2005  
Title:                      POWER MANAGEMENT SYSTEMS AND DEVICES

**DECLARATION OF DR. NIKOLAOS PAPANIKOLOPOULOS**

## TABLE OF CONTENTS

|              |                                                                                                                  |            |
|--------------|------------------------------------------------------------------------------------------------------------------|------------|
| <b>I.</b>    | <b>QUALIFICATIONS AND BACKGROUND</b>                                                                             | <b>3</b>   |
| <b>II.</b>   | <b>LEGAL PRINCIPLES</b>                                                                                          | <b>13</b>  |
|              | A. Anticipation                                                                                                  | 13         |
|              | B. Obviousness                                                                                                   | 13         |
| <b>III.</b>  | <b>OVERVIEW OF CONCLUSIONS FORMED</b>                                                                            | <b>15</b>  |
| <b>IV.</b>   | <b>BACKGROUND KNOWLEDGE ONE OF SKILL IN THE ART WOULD HAVE HAD PRIOR TO THE PRIORITY DATE OF THE '426 PATENT</b> | <b>15</b>  |
| <b>V.</b>    | <b>INTERPRETATIONS OF THE '426 PATENT CLAIMS AT ISSUE</b>                                                        | <b>16</b>  |
| <b>VI.</b>   | <b>OVERVIEW OF THE '426 PATENT</b>                                                                               | <b>17</b>  |
| <b>VII.</b>  | <b>PROSECUTION HISTORY OF THE '426 PATENT</b>                                                                    | <b>21</b>  |
| <b>VIII.</b> | <b>THE CHALLENGED CLAIMS ARE UNPATENTABLE</b>                                                                    | <b>23</b>  |
|              | A. [GROUND 1] – Claims 1-2 are Rendered Obvious by Neiss-Ran                                                     | 23         |
|              | 1. Overview of Neiss (TESLA-1005)                                                                                | 23         |
|              | 2. Overview of Ran (TESLA-1006)                                                                                  | 35         |
|              | 3. Combination of Neiss and Ran                                                                                  | 43         |
|              | 4. Claim Analysis                                                                                                | 52         |
|              | (a) Claim 1                                                                                                      | 52         |
|              | (b) Claim 2                                                                                                      | 107        |
|              | B. [GROUND 2] – Claim 3 is Rendered Obvious by Neiss-Ran-Andersson.                                              | 109        |
|              | 1. Overview of Andersson (TESLA-1007)                                                                            | 109        |
|              | 2. Combination of Neiss, Ran, and Andersson                                                                      | 111        |
|              | 3. Claim 3 Analysis                                                                                              | 118        |
|              | C. [GROUND 3] – Claim 4 is Rendered Obvious by Neiss-Ran-Hosoda                                                  | 122        |
|              | 1. Overview of Hosoda                                                                                            | 122        |
|              | 2. Combination of Neiss, Ran, and Hosoda                                                                         | 124        |
|              | 3. Claim 4 Analysis                                                                                              | 128        |
| <b>IX.</b>   | <b>ADDITIONAL REMARKS</b>                                                                                        | <b>130</b> |

I, Nikolaos Papanikolopoulos, of Minneapolis, Minnesota, declare that:

## **I. QUALIFICATIONS AND BACKGROUND**

1. My name is Nikolaos Papanikolopoulos and I have been retained as an expert witness by Tesla in a petition for *inter partes* review challenging all claims (claims 1-4) of U.S. Patent No. 7,925,426 (the “’426 Patent”). My qualifications for forming these conclusions are summarized below.

2. I am the McKnight Presidential Endowed Professor and Distinguished McKnight University Professor in the Department of Computer Science and Engineering of the University of Minnesota in Minneapolis, Minnesota. I am also the Director for the Minnesota Robotics Institute.

3. I have been a professor at the University of Minnesota (originally as an assistant professor, and then as an associate professor) since the Fall of 1992. Between Fall 2001 and Spring 2004, and between Fall 2010 and Spring 2013, I was the Director of Undergraduate Studies of the College of Science and Engineering.

4. In 1992, I received my Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University. My thesis was entitled “Controlled Active Vision” and focused on using computer vision in a controlled fashion to detect, track, and manipulate objects in the environment. In 1988, I received my M.S. in Electrical and Computer Engineering from Carnegie Mellon University. My B.S. in Electrical

Engineering was received in 1987 from the National Technical University in Athens, Greece.

5. For over thirty years, my research and teaching work has focused on intelligent vehicles, computer vision, and robotics. This research has included detection and tracking of humans and objects such as vehicles using information from stationary or moving cameras. It has also included object detection and recognition, including work with artificial intelligence and pattern recognition systems for surveillance applications. Finally, my research involved using the sensory information to control the motion of vehicles in real-time.

6. I currently teach three courses related to intelligent vehicles: (i) CSci 5551 Introduction to Intelligent Robotic Systems, (ii) CSci 5561 Computer Vision, and (iii) CSci 5511 Artificial Intelligence.

7. I am currently serving as a Senior Editor in the IEEE Transactions on Intelligent Transportation Systems with responsibility in assigning papers to Associate Editors based on their expertise. I was Associate Editor in the same journal for twelve years.

8. I have received research grants for various projects involving the transportation industry, including vehicle tracking, assessing truck parking, detection of pedestrians, and obstacle avoidance. My industry experience includes

founding a robotics company in 2005 named ReconRobotics Inc. that develops different robotic platforms. I have also been a consultant since 1997 for numerous companies and provided technical expertise in developing computer vision algorithms for vehicle tracking and traffic control, camera calibration, inspection, precision agriculture, human tracking, object recognition, and fingerprint recognition. In one of my consulting assignments, I had to develop computer vision algorithms for inspection of industrial parts by using different illumination sources (Banner Engineering). I have licensed my algorithms to several companies.

9. My research has produced more than 380 journal and conference publications. More than 90 publications are in refereed journals. Many of my publications and research grants relate to vehicle technologies. Some examples include:

*Wang, Z., and Papanikolopoulos, N., “Estimating Pedestrian Crossing States Based on Single 2D Body Pose”, Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '20), pp 2205-2210, Las Vegas, NV, October 25-29, 2020.*

*Zermas, D., Izzat, I., and Papanikolopoulos, N., “Fast Segmentation of 3D Point Clouds: A Paradigm on LiDAR Data for Autonomous Vehicle Applications”, Proceedings of the 2017 IEEE Int. Conference on Robotics and Automation, pp 5067-5073, Singapore, May 29 – June 3, 2017.*

*Cook, D., Morris, T., Morellas, V., and Papanikolopoulos, N., “An Automated System for Persistent Real-Time Truck Parking Detection and Information Dissemination”, Proceedings of the 2014 IEEE Int. Conference on Robotics and Automation, Hong Kong, China, May 31 – June 7, 2014, PP. 3989-3994.*

*Atev, S., Miller, G., and Papanikolopoulos, N.P., "Clustering of Vehicle Trajectories", IEEE Trans. on Intelligent Transportation Systems, Volume 11, No. 3, September 2010, PP. 647-657.*

*Atev, S., Arumugam, H., Masoud, O., Janardan, R., and Papanikolopoulos, N.P., "A Vision-Based Approach to Collision Prediction at Traffic Intersections", IEEE Trans. on Intelligent Transportation Systems, Volume 6, No. 4, December 2005, PP. 416-423.*

*Masoud, O., and Papanikolopoulos, N.P., "A Novel Method for Tracking and Counting Pedestrians in Real-time Using a Single Camera", IEEE Trans. on Vehicular Technology, Volume 50, No. 5, September 2001, PP. 1267-1278.*

*Du, Y., and Papanikolopoulos, N.P., "Real-time Vehicle Following Through a Novel Symmetry-Based Approach", Proceedings of the 1997 IEEE Int. Conf. on Robotics and Automation, PP. 3160-3165, Albuquerque, NM, April 20-25, 1997.*

10. My work in vehicle detection and tracking has focused on using image and video processing techniques, including object detection, tracking, and recognition, to accurately track vehicles in real-time. These algorithms have been applied to different scenarios, such as urban traffic monitoring and surveillance. Moreover, machine learning techniques have helped in the improvement of these methodologies.

11. My research has also involved tracking autonomous vehicles. My interests include developing tracking algorithms and systems that can handle the complexities of autonomous vehicles, including predicting their future movements

and understanding their intentions. The work has been extended to monitoring the intentions of pedestrians.

12. These vehicle tracking systems enable continuous monitoring and tracking of vehicles, providing valuable information for traffic control and management, law enforcement, and security purposes.

13. My paper “Detection and Classification of Vehicles,” published in the IEEE Transactions on Intelligent Transportation Systems in March 2002, is one of the most cited papers in the history of this journal (more than a thousand citations). The paper, among other contributions, shows how to use detected headlights and taillights to extract vehicles from videos and images in low-light conditions. One should note that the majority of my papers in the broader area of intelligent vehicle systems dealt with the detection of vehicles, pedestrians, trucks, etc., by using a variety of techniques, which were mainly based on computer vision. I also have nine patents in pertinent areas.

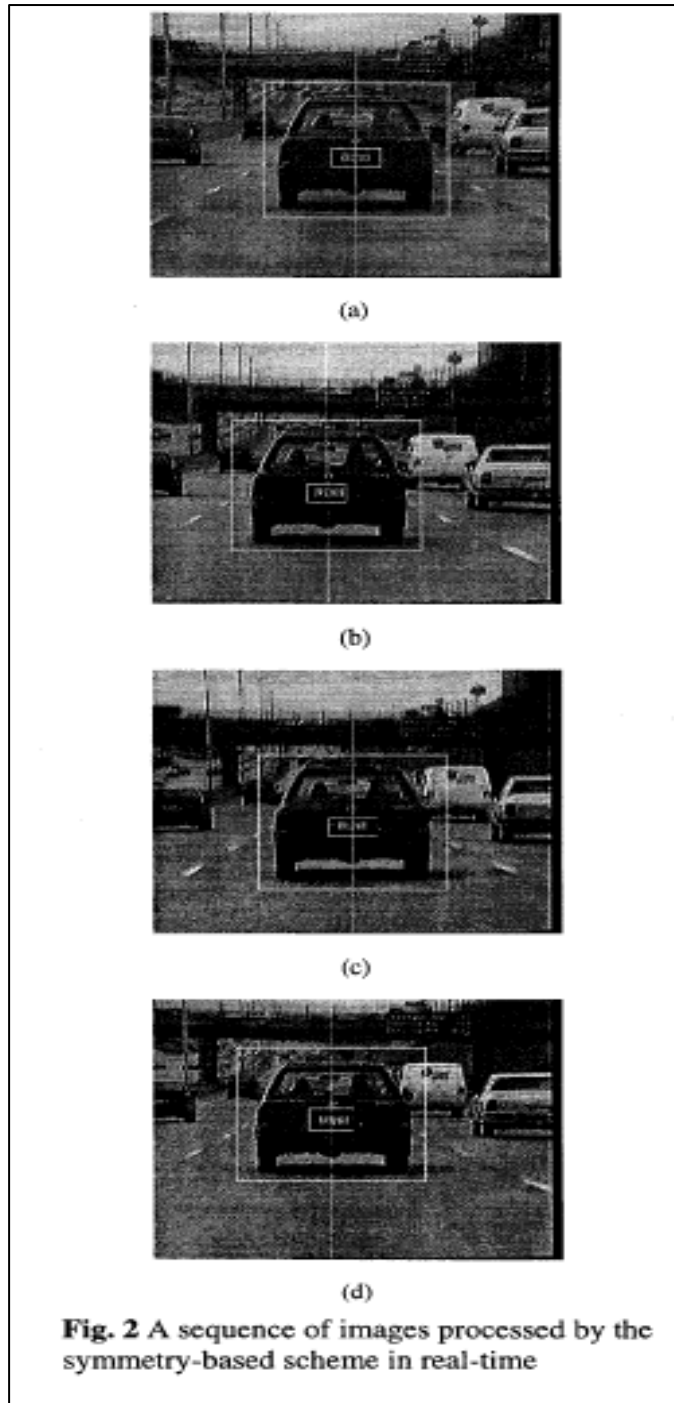
14. My research work has been funded by several government agencies. For example, I have received funding from the Department of Homeland Security to monitor threats at Mass Transit Sites, like the 30th Street Station in Philadelphia, PA, and the Light Rail Station in the Minneapolis-Saint Paul International Airport.

Monitoring parked vehicles was an essential task for which we developed a variety of computer vision algorithms and tools.

15. For more than a decade, I have been developing computer vision-based systems to assess truck parking availability at truck rest areas. The funding came from Federal Highway Administration (FHWA), Minnesota Dept. of Transportation (MnDOT), Wisconsin Dept. of Transportation (WiscDOT), and Kansas Dept. of Transportation (KansasDOT). The approach uses multiple cameras to detect trucks and other vehicles and determine empty spaces. The system operates in all weather conditions around the clock.

16. I have received numerous honors and awards for my research and contributions. I have been a Distinguished McKnight University Professor at the University of Minnesota since 2007 and have been a McKnight Presidential Endowed Professor in Computer Science since 2016. Also in 2007, I was nominated and became an IEEE Fellow. In 2016, I received the IEEE RAS George Saridis Leadership Award in Robotics and Automation as well as the Center for Transportation Studies Research Partnership Award. I have also received the IEEE VTS 2001 Best Land Transportation Paper Award for the paper “A Novel Method for Tracking and Counting Pedestrians in Real-Time Using a Single Camera” (with Osama Masoud).

17. The figures below show the detection and tracking of a vehicle by a camera mounted in a car that follows a target vehicle. These figures are from a paper that I had published in 1997: Du, Y., and Papanikolopoulos, N.P., “Real-time Vehicle Following Through a Novel Symmetry-Based Approach,” *Proceedings of the 1997 IEEE Int. Conf. on Robotics and Automation*, PP 3160-3165, Albuquerque, NM, April 20-25, 1997.



18. I have also worked on the use of sensors to monitor driver fatigue. I developed a computer vision-based system that analyzes facial features and eye

movements to detect signs of drowsiness and driver fatigue. By monitoring factors such as eye closure, head pose, and blink patterns, the system can alert drivers when they show signs of fatigue and suggest appropriate actions (this research took place in the late 1990s). The proposed algorithms and systems considered the continuously evolving light conditions that vehicular systems operate within.

19. In writing this Declaration, I have considered the following: my own knowledge and experience, including my work experience in the relevant field(s); my experience in teaching subjects in these field(s); and my experience in working with others involved in those fields. In addition, I have analyzed the following publications and materials, in addition to other materials I cite in my Declaration:

- U.S. Patent No. 7,925,426 (TESLA-1001), and its accompanying prosecution history (TESLA-1002)
- U.S. Patent Application Publication No. 2004/0068359 to Neiss et al. (“Neiss”) (TESLA-1005)
- U.S. Patent No. 6,317,686 to Ran (“Ran”) (TESLA-1006)
- U.S. Patent No. 6,836,719 to Andersson et al. (“Andersson”) (TESLA-1007)
- Japanese Patent Publication No. 11220807 (“Hosoda Original”) (TESLA-1008), and its accompanying certified translation (“Hosoda”) (TESLA-

1009) (all cites to Hosoda are to its certified translation, TESLA-1009)

- U.S. Patent No. 6,314,347 (“Kuroda”) (TESLA-1011)
- U.S. Patent Application Publication No. 2006/0195249 (“Fosseen”) (TESLA-1012)
- U.S. Patent Application Publication No. 2005/0228556 (“Kojima”) (TESLA-1013)
- U.S. Patent 6,843,337 (“Matsubara”) (TESLA-1014)
- U.S. Patent 6,895,320 (“Bauer”) (TESLA-1015)
- U.S. Patent Publication No. 2004/0084237 (“Petrie”) (TESLA-1016)
- U.S. Patent Publication No. 2005/0004742 (“Michi”) (TESLA-1017)
- U.S. Patent Publication No. 2005/0216793 (“Entin”) (TESLA-1018)
- U.S. Patent Application Publication No. 2004/0230376 (“Ichikawa”) (TESLA-1019)
- U.S. Patent No. 6,396,394 (“Suzuki”) (TESLA-1020)
- Excerpts from Heinle’s Newbury House Dictionary of American English (4<sup>th</sup> Ed.) (TESLA-1025)
- Excerpts from Webster’s College Dictionary (TESLA-1026)
- U.S. Patent No. 7,013,205 (“Hafner”) (TESLA-1027)

## **II. LEGAL PRINCIPLES**

### **A. Anticipation**

20. I have been informed that a patent claim is invalid as anticipated under 35 U.S.C. § 102 if each and every element of a claim, as properly construed, is found either explicitly or inherently in a single prior art reference. Under the principles of inherency, if the prior art necessarily functions in accordance with, or includes the claimed limitations, it anticipates.

21. I have been informed that a claim is invalid under 35 U.S.C. § 102(a) if the claimed invention was known or used by others in the U.S., or was patented or published anywhere, before the applicant's invention. I further have been informed that a claim is invalid under 35 U.S.C. § 102(b) if the invention was patented or published anywhere, or was in public use, on sale, or offered for sale in this country, more than one year prior to the filing date of the patent application (Critical Date). And a claim is invalid, as I have been informed, under 35 U.S.C. § 102(e), if an invention described by that claim was described in a U.S. patent granted on an application for a patent by another that was filed in the U.S. before the date of invention for such a claim.

### **B. Obviousness**

22. I have been informed that a patent claim is invalid as "obvious" under 35 U.S.C. § 103 in light of one or more prior art references if it would have been

obvious to a POSITA, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called “secondary considerations” of non-obviousness, which include: (i) “long felt need” for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) “copying” of the claimed invention by others. For purposes of my analysis above and because I know of no indication from the patent owner or others to the contrary, I have applied a date of November 17, 2005, as the date of invention in my obviousness analyses, although in many cases the same analysis would hold true even at an earlier time than November 17, 2005.

23. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference or references themselves, the knowledge or “common sense” of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been informed that the combination of familiar elements according to known

methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

### **III. OVERVIEW OF CONCLUSIONS FORMED**

24. This expert Declaration explains the conclusions that I have formed based on my analysis. To summarize those conclusions:

- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1-2 of the '426 Patent are obvious over Neiss in view of Ran.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 3 of the '426 Patent are obvious over Neiss in view of Ran, and in further view of Andersson.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 4 of the '426 Patent are obvious over Neiss in view of Ran, and in further view of Hosoda.

### **IV. BACKGROUND KNOWLEDGE ONE OF SKILL IN THE ART WOULD HAVE HAD PRIOR TO THE PRIORITY DATE OF THE '426 PATENT**

25. Based on the foregoing and upon my experience in this area, a person of ordinary skill in the art ("POSITA") relating to the subject matter of the '426

Patent as of November 17, 2005 (“Critical Date”) would have had a bachelor’s degree in electrical engineering, computer engineering, mechanical engineering (including automotive engineering), civil engineering (transportation systems), physics, or a related field, and at least four years of experience (or the academic equivalent) in the field of transportation systems. Additional graduate education could substitute for professional experience, and vice versa.

26. Based on my experiences, I have a good understanding of the capabilities of a POSITA as I possessed and exceeded qualifications of a POSITA at the time of the Critical Date. Moreover, I have taught, participated in organizations with, and worked closely with many such persons over the course of my career.

## **V. INTERPRETATIONS OF THE ’426 PATENT CLAIMS AT ISSUE**

27. I have been informed by Counsel and understand that the best indicator of claim meaning is its usage in the context of the patent specification as understood by one of ordinary skill. I further understand that the words of the claims should be given their plain meaning unless that meaning is inconsistent with the patent specification or the patent’s history of examination before the Patent Office. Counsel has also informed me, and I understand that, the words of the claims should be interpreted as they would have been interpreted by a POSITA at the time of the

invention was made (not today). I have been informed by Counsel that I should use November 17, 2005 as the point in time for claim interpretation purposes.

## **VI. OVERVIEW OF THE '426 PATENT**

28. The '426 Patent discloses and claims a method of “managing the power consumption of an automotive vehicle, and thereby for optimizing the power consumption of the vehicle.” TESLA-1001, 2:7-10, Claim 1.

29. The '426 Patent states that “optimizing the power supplied to the engine typically means minimizing the energy loss from the vehicle, thereby increasing fuel efficiency.” *Id.*, 6:14-17. The '426 Patent contemplates applicability of its techniques to any appropriate vehicle, including fuel-powered vehicles, hybrid vehicles, and electric vehicles. *See id.*, 6:20-30.

30. The claimed method includes “receiving a route” at “a computer device.” *Id.*, 26:27. The route can be “determined based on the current position of the vehicle and a final destination position,” which can be “explicitly provided or derived” from “information about the operator, the current location, the time of day, or the like.” *Id.*, 11:27-30, 11:46-47.

31. The claimed method further recites that the “computer device segment[s] said route into a plurality of segment destinations.” *Id.*, 26:28-29. The '426 Patent's specification does not use the claim phrase “segment destination.” *See*

*generally id.* In the context of route segmentation, the '426 Patent's specification states that the "route ... typically includes a starting position ..., an ending position ..., and any intermediate positions between the initial and the final position." *Id.*, 13:36-42. Each "segment may comprise any distance to be traveled, including the entire route, or small portions of the route." *Id.*, 13:45-47. The "route may be segmented in any appropriate manner" such as, e.g., based on "predetermined or anticipated changes in speed (e.g., switching from 65 to 55 mph), changing traffic patterns (e.g., turns, stops, yields, etc.), traffic or anticipated traffic, distance (e.g., x miles), terrain (e.g., the gradient or condition of a road), or the like." *Id.*, 13:49-56.

32. The claimed method further recites a series of operations to be performed by the computing device with respect to each segment destination. *See generally id.* The '426 Patent's specification describes performing operations with respect to route segments—and not segment destinations as recited in the claim. *See generally id.*

33. As described below, for each segment destination, the claimed method recites the computer device performing the following four operations: (1) calculating an optimized speed, (2) calculating a probable speed, (3) determining an efficient speed based on the optimized and probable speeds, and (4) applying calculated power corresponding to the determined efficient speed to the vehicle engine to travel

the segment destination. *See id.*, Claim 1.

34. The *first* claimed operation recites “calculating an optimized speed by minimizing power consumption over said segment destination,” where one or more of the following information is “received as input” for this calculation:

- (a) “information about the external environment of a vehicle,”
- (b) “information about the operational status of the vehicle,”
- (c) “one or more user command inputs,” or
- (d) “one or more operational parameters of the vehicle.” *See id.*, 26:30-39.

35. The claim phrase “optimized speed” is not used in the ’426 Patent’s specification. *See generally id.* Instead, the specification recites an “optimal speed” that can be calculated “by energy calculation logic,” where the logic uses “information inputs (e.g., information from the external environment of the vehicle, the operational status of the vehicle, information from one or more command inputs, and operational parameters of the vehicle).” *Id.*, 14:30-49.

36. The ’426 Patent provides examples of each type of the “information inputs.” *Id.*, 7:21-51, 8:21-40, 8:61-9:6, 9:55-10:2, 25:33-34. “Information from the extent environment of the vehicle” include, among others, “the current location of the vehicle, geographical information about the surrounding area, the elevation of the vehicle, upcoming elevations of the vehicle, the current slope/grade of road, the

predicted slope/grade of the next segments of road,” etc. *Id.*, 7:21-51.

37. “Information about the operational status of the vehicle” include, among others, “the vehicle’s current speed, the motor speed, the vehicle’s current orientation, the RPM of the vehicle’s motor,” etc. *Id.*, 8:21-40.

38. “Examples of command inputs” include “the acceleration applied by a driver, the braking applied by a driver, the vehicle’s known or predicted final destination, the vehicle’s known or predicted interim destination, preferred speed, maximum and minimum range over which speed should be adjusted, and preferred route.” *Id.*, 8:61-9:6.

39. “Information from one or more operational parameters” include, among others, “aerodynamic parameters (CDA), rolling resistance parameters (Crr1 and Crr2),” and “maximum fuel volume.” *Id.*, 9:55-10:2, 25:33-34.

40. The ***second*** claimed operation recites “calculating a probable speed” for the segment destination. *Id.*, 26:40. The ’426 Patent’s specification does not use the phrase “probable speed,” nor does the claim provide any requirements for what this phrase encompasses. *See generally* TESLA-1001.

41. The ***third*** claimed operation recites “determin[ing] an efficient speed based on said optimized speed and said probable speed.” *Id.*, 26:41-42. The claim does not specify how the efficient speed is determined based on the optimized speed

and the probable speed. In one example, the specification states that the “efficient speed” can be “determined by a combination of the calculated optimized speed and the probable optimized speed after they have been appropriately weighted.” *Id.*, 19:3-6.

42. The *fourth* and final claimed operation recites “applying the calculated power corresponding to said determined efficient speed” to “the vehicle engine to travel said segment destination.” *Id.*, 26:43-45. The specification does not disclose how “the calculated power corresponding to said determined efficient speed is applied.” *See generally id.* The specification, however, recognizes that an “efficient speed ... may also be expressed as the applied power that is provided to the engine to achieve the speed at which the fuel efficiency is optimal.” *Id.*, 18:62-65.

## **VII. PROSECUTION HISTORY OF THE '426 PATENT**

43. The originally-filed claims of the '426 Patent did not recite an optimized speed, a probable speed, and an efficient speed based on the optimized speed and the probable speed. TESLA-1002, 142-46 (the initially-filed independent claims were directed to calculating an applied power for the vehicle engine from information about the external environment of the vehicle, information about the operational status of the vehicle, one or more command inputs, and one or more operational parameters of the vehicle).

44. In the first issued action, the claims were initially rejected on §101 grounds. *Id.*, 75-76. In response, and to overcome this rejection, the applicant amended the claims to recite “one or more processors” for receiving and analyzing data, and to calculate “an applied power for the vehicle engine based on the received and processed data.” *Id.*, 62-70.

45. A second office action was then issued to reject the independent claims in view of U.S. Patent No. 6,843,337 (Matsubara et al.) (TESLA-1014), and to reject pending claim 19 (which recited the calculation of an energy efficient speed to travel to the segment destination, the determination of a target speed for the vehicle to travel to the segment destination, or calculating the applied power from the target speed) in view of Matsubara and U.S. Patent No. 6,895,320 (Bauer et al.) (TESLA-1015). *Id.*, 48-54.

46. In response, the applicant cancelled claims 1-16 and retained the method claim (then-pending claim 17), which was amended to recite, among other claim elements, the “optimized speed,” “probable speed,” and “efficient speed” limitations present in the issued claims. *Id.*, 41-44. In the corresponding remarks, the applicant stated—without additional reasoning—that the claims, as currently amended, were “non-obvious over Matsubara or over [] Matsubara in view of

Bauer.” *Id.*; *see id.* 44 (also stating that “the cited prior art of record does not teach or suggest the combination of elements defined in indepdent [sic] claim 17”).

47. The claims were subsequently allowed, with the Reasons for Allowance stating that “these prior art do not make obvious” the entirety of the pending and amended independent claim (which issued as claim 1). *Id.*, 21-25.

## **VIII. THE CHALLENGED CLAIMS ARE UNPATENTABLE**

### **A. [GROUND 1] – Claims 1-2 are Rendered Obvious by Neiss-Ran**

#### **1. Overview of Neiss<sup>1</sup> (TESLA-1005)**

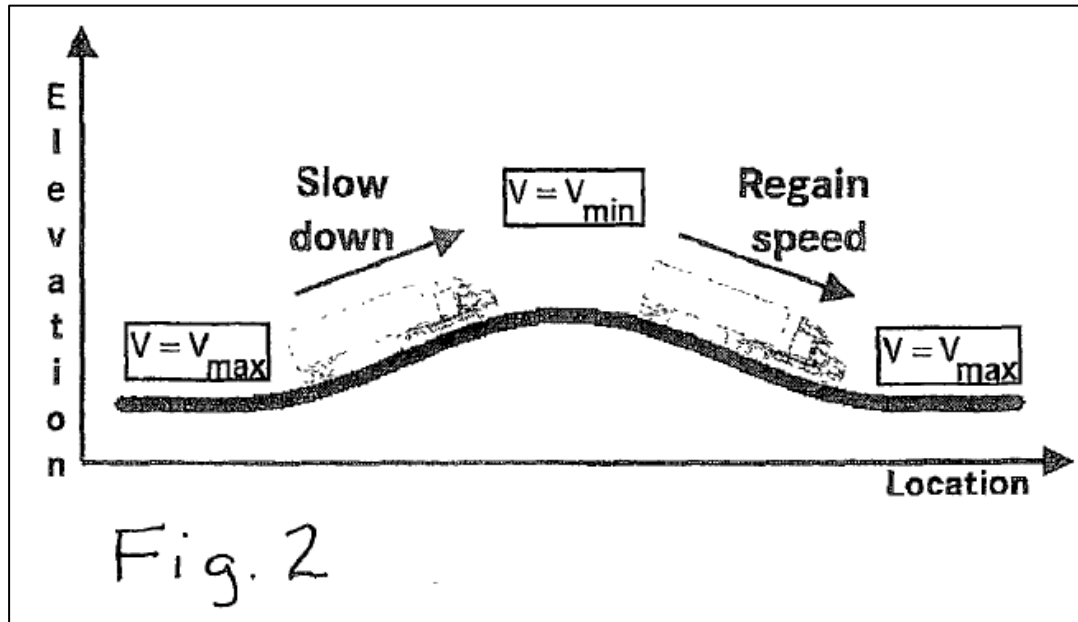
48. Neiss describes a “predictive cruise control system [that] utilizes information about the current vehicle position, as well as upcoming terrain... to save fuel and increase driving comfort.” TESLA-1005, Abstract.

49. Generally, Neiss’s cruise control system uses “[f]orward-looking terrain information ... to determine, as a function of position, a desired speed inside [a] speed band ... to maximize fuel savings.” *Id.*, Abstract, ¶[0031]. For instance, with reference to Figure 2, shown below, a vehicle implementing Neiss’s predictive

---

<sup>1</sup> Descriptions of the references and combinations thereof are incorporated into each mapping that includes citations to these references. All emphasis is added unless otherwise indicated.

cruise control may “slow[] down as it moves uphill until it reaches a minimum speed on top of the hill, and regains speed on the downhill part, thus completely converting the potential energy into kinetic energy.” *Id.*, ¶[0031].



*TESLA-1005, FIG. 2*

50. Neiss’s cruise control unit aims to “minimize[] vehicle operating costs” by optimizing “a vehicle operating cost function” that is defined based on “environmental parameters, vehicle parameters, vehicle operating parameters[,] and route parameters.” *Id.*, ¶[0001], [0015]. “As the vehicle proceeds” along its route, “an onboard computer iteratively calculates and stores in a memory vehicle control parameters that optimize the vehicle operating cost function for a predetermined distance ... along the route ahead of the vehicle.” *Id.*, ¶[0015]. “The vehicle is then

controlled by reading the optimized vehicle control parameters from the memory, corresponding to the current position of the vehicle.” *Id.*

51. Neiss describes an optimization problem with a cost function ( $J$ ) that depends on multiple subfunctions including fuel consumption ( $J_{fuel}$ ). Before the Critical Date, a POSITA would have known that optimization involves computing values for these variables that minimize the overall cost function, and that this formulation can incorporate constraints regarding these variables (e.g., an upper limit” and “a lower limit“ for the vehicle’s velocity) through the use of mathematical constructs, like Lagrange multipliers. *See also* TESLA-1005, ¶¶[0136]-[0160], [0088], [0092]-[0104], [0126], claims 8-9, 12.

52. A POSITA would have also recognized that Neiss discloses in the equation below factors that play a role in the computation of the total cost:

$$J = \Phi + J_{time} + J_{fuel} + J_{velocity} + J_{lateral \text{---} accel} + J_{penalty}$$

*Id.*, ¶¶[0088], [0126]. For example, this formulation uses an overall “cost function” (represented as  $J$ ) that is defined mathematically (as shown in the equation above) to reflect one or more objectives, including “consum[ing] less fuel” and “activat[ing] the brakes as little as possible” (represented as  $J_{fuel}$ ), “maintain[ing] a specified velocity (Set Speed)” (represented as  $J_{velocity}$ ), maintaining the “vehicle’s velocity”

between “an upper limit” and “a lower limit” (represented as  $J_{penalty}$ ), minimizing “travel time” (represented as  $J_{time}$ ), and/or reducing “lateral acceleration while driving in curves” so as not “to impair driving comfort and safety” (represented as  $J_{lateral\_accel}$ ). *See* TESLA-1005, ¶¶[0082]-[0126]. Combining all these individual costs to a total cost that depends on all these variables is an extensively researched process which has as an objective total cost that captures all the variables that may play a role for the underlying optimization task.  $J$  is a total cost which is expressed as a function of variables that the user can optimize through an optimization technique. *See id.*

53. A POSITA would have known how to select and apply a technique like gradient descent, linear programming, etc. to find the values of the variables that minimize the overall cost function while the constraints are satisfied. Furthermore, reducing one cost through changes in the value of the corresponding variable may impact another cost. Optimization is often described as a balancing act where different tradeoffs need to be considered. It should be noted that the optimization techniques often include iterative steps to continuously adjust the values of the variables so lower individual costs can be achieved. A POSITA would have known that neural networks, simulated annealing and other techniques could have been used to find the optimal values that minimize the aforementioned total cost.

54. Additionally, as mentioned above and in one example, Neiss describes this “vehicle operating cost function” ( $J$ ) including an objective defined to minimize fuel consumption (represented as  $J_{fuel}$ ), which Neiss mathematically expresses as follows:

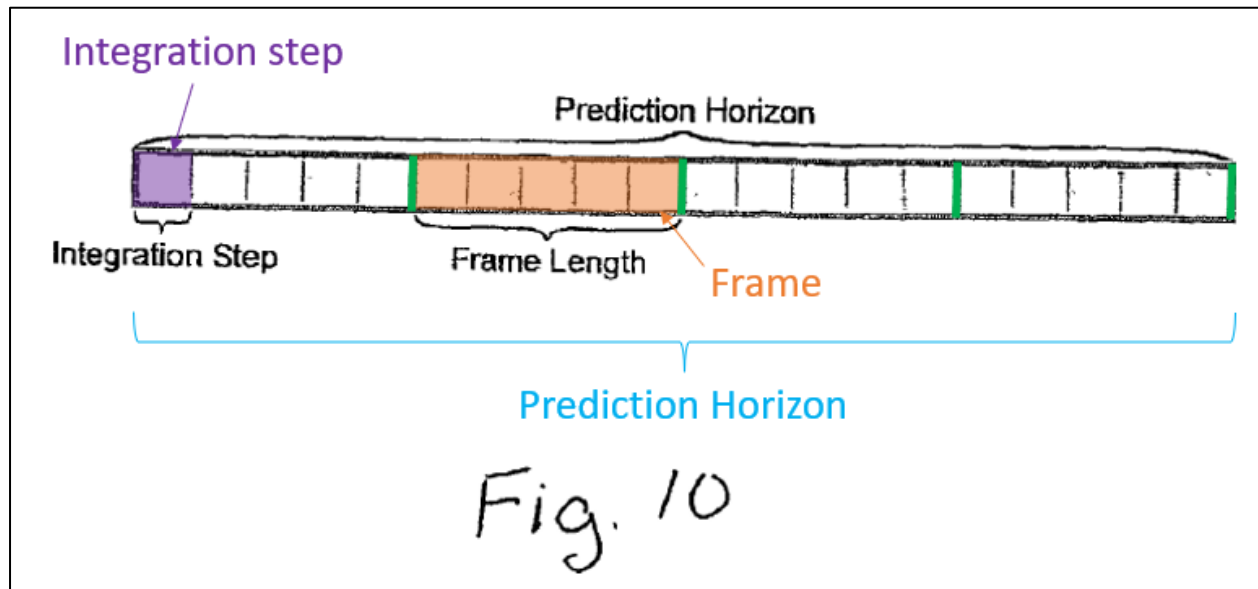
$$J_{fuel} = \frac{1}{2} \cdot h \cdot \sum_{k=0}^{N-1} S \psi v_k t h_k^2, \quad (\text{Eq. 18})$$

*Id.*, ¶¶[0081]-[0082], [0097]-[0098]. This equation is represented as “a function of [at least] vehicle speed, road gradient[, ] and throttle position” at points along the route ahead of the vehicle. *Id.*, ¶¶[0097]-[0098], Claims 10-11; *see id.*, ¶¶[0162], [0180], Abstract. Neiss thus describes an optimization problem with a cost function that depends on multiple subfunctions including fuel consumption ( $J_{fuel}$ ).

55. A POSITA would have known that neural networks, simulated annealing and other techniques could have been used to find the optimal values that minimize the aforementioned total cost.

56. The “Prediction Horizon” is “a predetermined distance ... along the route ahead of the vehicle.” *Id.*, ¶[0015]. As illustrated in Figure 10, shown below, the “Prediction Horizon” along the route ahead of the vehicle “is separated into Frames.” *Id.*, ¶[0170]. Each frame corresponds to “a specified distance” to be covered by the vehicle and is composed of a number of integration steps  $h$ . *Id.*,

¶¶[0169]-[0170] (“the Prediction Horizon must be a multiple of the Frame Length (also measured in number of integration steps  $h$ )”).



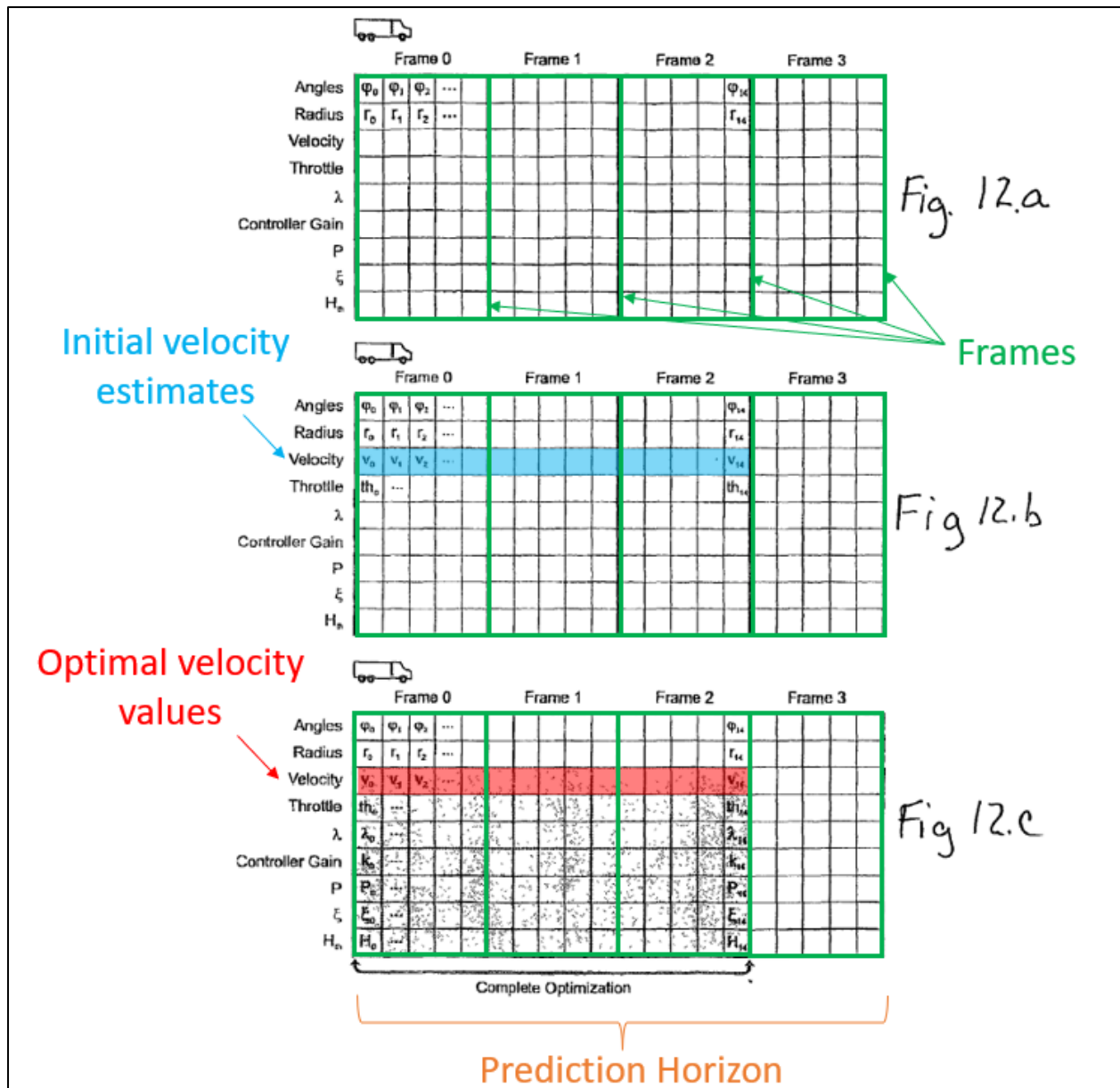
*TESLA-1005, FIG. 10 (annotated)*

57. To perform the minimization of the above-described “vehicle operating cost function,” an iterative optimization algorithm is executed to calculate vehicle control parameters including “optimal velocity, throttle pedal positions, and controller gain series” “[f]or an area around the current vehicle position[] (“Prediction Horizon”).” *Id.*, ¶¶[0015], [0162].

58. The optimization of the “vehicle operating cost function” along the Prediction Horizon is performed, with a lookup table being used to store the calculated “values for vehicle velocity, throttle position and controller gain” for each integration step of the Frames, as well as the values for “variables needed for the

calculation of the optimization algorithm.” *Id.*, ¶[0171]; *see also id.*, ¶[0162] (“For an area around the current vehicle position[] (“Prediction Horizon”), the optimal velocity, throttle pedal positions, and controller gain series are stored in a look-up table”).

59. As part of the optimization, Neiss’s cruise control system uses an initial set of estimated vehicle control parameters to compute a first set of optimized vehicle control parameters, including vehicle speed, for frames along the prediction horizon. TESLA-1005, ¶¶[0171]-[0181], [0186], [0048], [0149]-[0160], FIGS. 12a-c. The initial estimates and the computed, optimized vehicle parameters are stored in a lookup table corresponding to the frames of the prediction horizon, as shown below:



TESLA-1005, FIGS. 12a-c (annotated)

60. These vehicle control parameters are “continuously updated and replaced by new data as the vehicle moves along, thereby adjusting [these parameters] to reflect actual vehicle historical operating experience during the

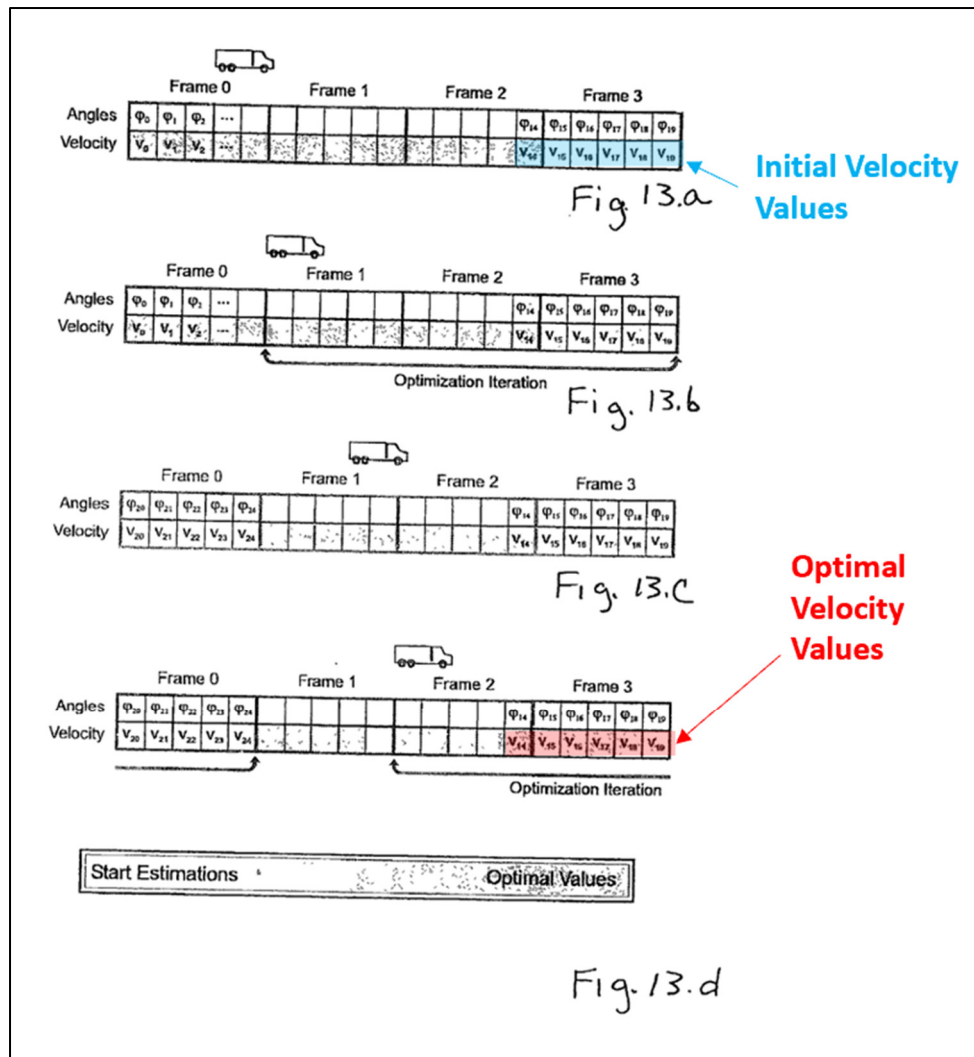
journey.” TESLA-1005, Abstract; *see id.*, ¶[0180] (explaining that the calculation is an iterative process that is carried out “until the algorithm has converged and thus the values are optimal”).

61. Specifically, in the initialization stage, “angle [road grade] and radius [radius of curvature of the road] values” are populated in the lookup table “from cell zero up to the last cell of the next to last frame,” that is, for each integration step (distance increment). *Id.*, ¶[0179]; *see id.*, FIG. 12a. Also, as part of the initialization stage, “starting values are estimated for the velocity trajectory, and the appropriate throttle pedal positions are computed” for each distance increment. *Id.*, *see id.*, FIG. 12b (velocity values highlighted in blue). Notably, in Figure 12b, “[t]he white background of the velocity memory cells ... indicates that these values are starting estimates ... and thus are not optimal.” TESLA-1005, ¶[0181].

62. In the second step of the initialization stage, before the “vehicle reaches the start point on the road where the predictive cruise control (PCC) system is triggered,” “several optimization iterations are applied on the memory range” (i.e., through the next to last frame of the Prediction Horizon) “until the algorithm has converged and thus the values are optimal.” TESLA-1005, ¶¶[0178], [0180]; *see id.*, FIG. 12c (optimal velocity values highlighted in red). In contrast to Figure 12b, Figure 12c’s velocity values and throttle pedal positions are highlighted with a “dark

background,” which “indicates that the velocity values are optimal.” *Id.*, ¶[0181]. Each iteration of the optimization calculation involves the calculation of “velocity, throttle[,] and controller gain values.” *Id.*, ¶[0186]; *see also id.*, ¶¶[0048], [0149]-[0160].

63. After initialization, optimization of the cost function continues “while the vehicle is driving,” as illustrated in FIGS. 13a-d, *infra*. TESLA-1005, ¶[0177]. Specifically, with reference to FIGS. 13a-d (reproduced below), Neiss explains that a “new calculation is triggered “[w]hen the vehicle reaches the end of the first frame.” *Id.*, ¶[0188]. The new calculation is initialized, assigning estimated “starting value[s]” of *v* and *th* for frame 3. *Id.*, ¶[0189]; *see* FIG. 13a (velocity values highlighted in blue). For this optimization iteration, “the current vehicle velocity is used as [the] start[ing] velocity in order to take the actual vehicle state into account. *Id.*, ¶[0190]. “The current vehicle velocity is used as start velocity” and “a whole optimization iteration is applied” to calculate the “more optimal” values for vehicle control parameters for frames 1, 2, and 3. *Id.* Later, “when the vehicle is driving in frame 2,” another “optimization iteration is applied on frames 2, 3, and 0,” yielding velocity values for frame 3 that “approach closer to the optimal values” (highlighted in red in FIG. 13d). *Id.*, ¶[0191].



TESLA-1005, FIGS. 13a-d (annotated)

64. Neiss further notes that when the “vehicle velocity drifts significantly from the velocity stored in the look-up table for the current position (as might be caused, for example, by a slower vehicle driving ahead), the velocity values for the following positions in the look-up table then can no longer be considered to be

optimal.” *Id.*, ¶[0167]. When this occurs, “an online version of the algorithm can adapt these values by taking the current velocity into account.” *Id.*

65. Notably, although FIGS. 12a-c and 13a-d show a Prediction Horizon spanning only four frames “to make the basic operation of the ring buffer clear,” Neiss explains that “[r]ealistic values for the Prediction Horizon would be 4000 assuming an Integration Step Size of 1 m, and 200 for the Number of Frames.” TESLA-1005, ¶[0192]. Thus, “nearly 200 optimization iteration steps are performed with regard to the values in one frame before they are actually read out.” *Id.*

66. When the calculated optimized parameters are read out of the lookup table (e.g., for the corresponding location where the vehicle is positioned/located), they are provided as inputs to a controller “which controls the vehicle to follow the calculated velocity trajectory.” *Id.*, ¶[0041]; *see id.*, ¶[0188]. Generally, “the desired vehicle speed” (i.e., the calculated optimized speed stored in the lookup table) is “converted to a desired engine speed, which is sent to the engine and the engine achieves it via its own controller or using a separate adaptive cruise controller.” *Id.*, ¶[0042]. For example, “the engine determines how much fuel to inject” or determines “a throttle angle.” *Id.*

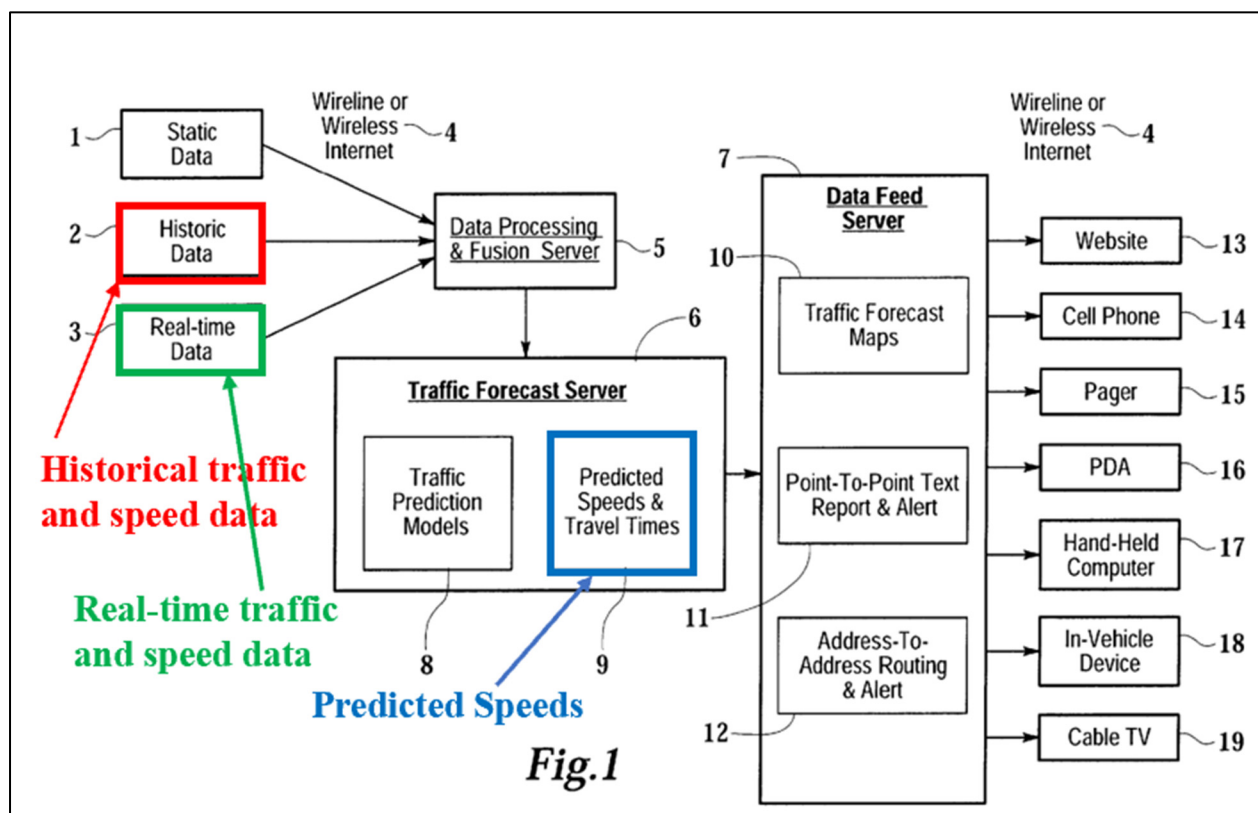
67. Neiss is analogous art to the ’426 Patent: it is from the same field of endeavor (e.g., vehicle speed determinations) and is reasonably pertinent to the

problem the '426 Patent was trying to solve (e.g., problem of how to determine speeds at which a vehicle should travel based, e.g., on road or operating conditions).

## 2. Overview of Ran (TESLA-1006)

68. Ran describes a system and method of predicting a travel time and speed for a route. TESLA-1006, 1:12-13.

69. An example of Ran's system is shown below:

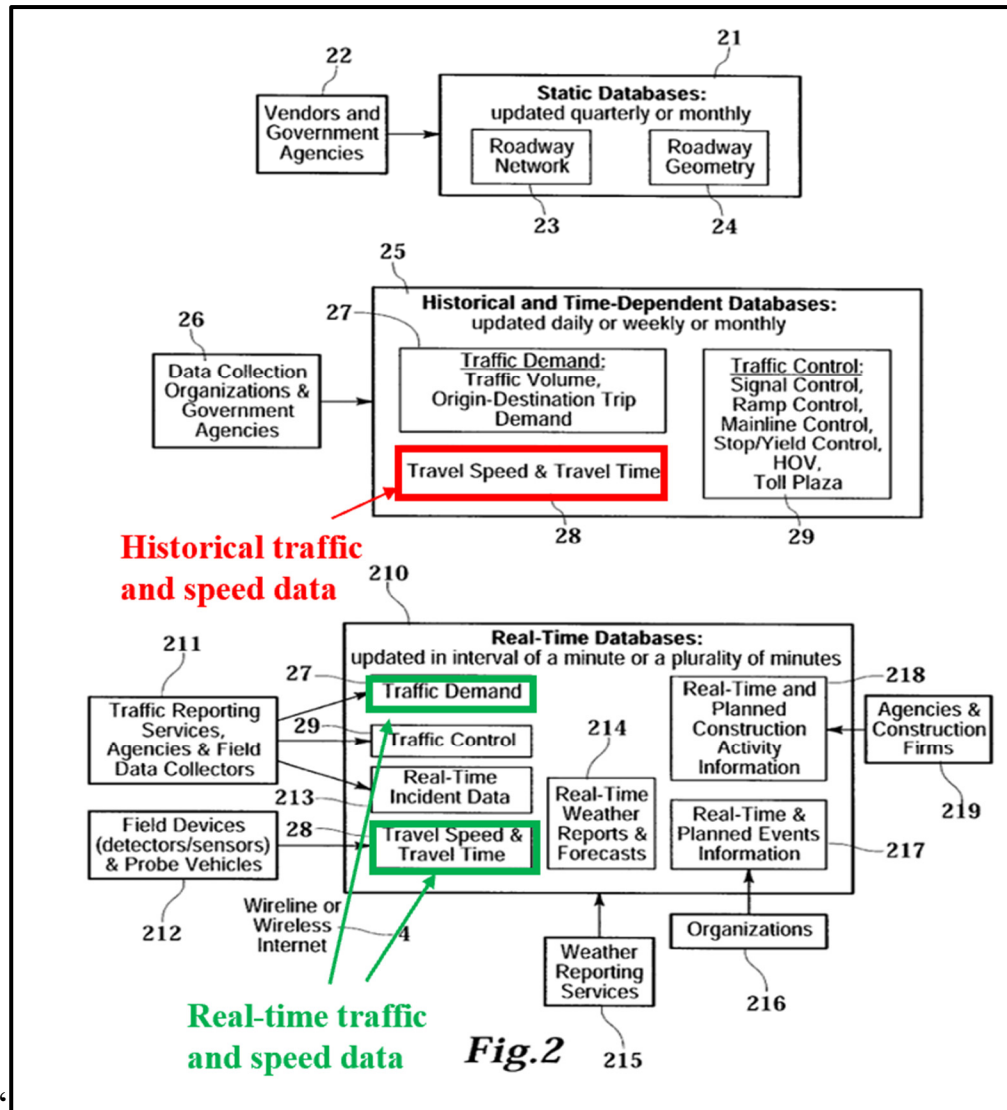


TESLA-1006, FIG. 1 (annotated)

70. As shown above, Ran's "Internet-based traffic prediction system has a data processing and fusion server 5, a traffic forecast server 6, and a data feed server

7. The data processing and fusion server 5 uses wireline or wireless internet 4 and communications network to retrieve static data 1, historical data 2, and real-time data 3.” TESLA-1006, 17:7-12.

71. Ran describes collecting this “static data 1, historical data 2, and real-time data 3” from various sources, including static databases, historical databases, and real-time databases to increase the accuracy in a predicted travel time and speed for traversing a route. *See id.*, 17:11-12, 18:18-31, FIG. 2.



TESLA-1006, FIG. 2 (annotated)

72. The “static databases 21 define the route segment network 23,” which includes “node[s], link[s], node coordinate[s], link length[s], address[es], route segment classification[s], speed limit[s], driving restriction[s], facilit[ies], landmark[s], etc.” and “route segment geometry 24,” which includes the “number of lanes, intersection layout[s], interchange layout[s], road curvature[s],

vertical/horizontal alignment[s], etc.” TESLA-1006, 17:38-43. Such databases “are typically available from map database vendors and government agencies.” *Id.*, 17:44-45.

73. The “historical and time-dependent databases 25” include “traffic demand 27, travel speed and travel time 28, and traffic control 29” databases. *Id.*, 17:48-50. The “traffic demand 27 database typically includes traffic volumes on route segments, trip generation at origin, trip attraction at destination, and origin-destination trip demand.” *Id.*, 17:50-53. The “travel speed and travel time 28 database provides the historical information for different time of day and day of week.” *Id.*, 17:53-55. The “traffic control 29 database stores the information on signal control, ramp control, freeway mainline control, stop/yield control, HOV control and toll plaza control.” *Id.*, 17:55-58.

74. The “real-time databases 210” include “seven major databases: traffic demand 27, traffic control 29, real-time incident 213, travel speed and travel time 28, real-time weather reports and forecasts 214, real-time and planned events information 217, and real-time and planned construction activity information 218,” which are “typically updated in intervals of a minute or a plurality of minutes by retrieving data from various sources via wireline or wireless Internet 4.” *Id.*, 17:64-18:5. “The real-time data of traffic demand 27, traffic control 29, and incident 213

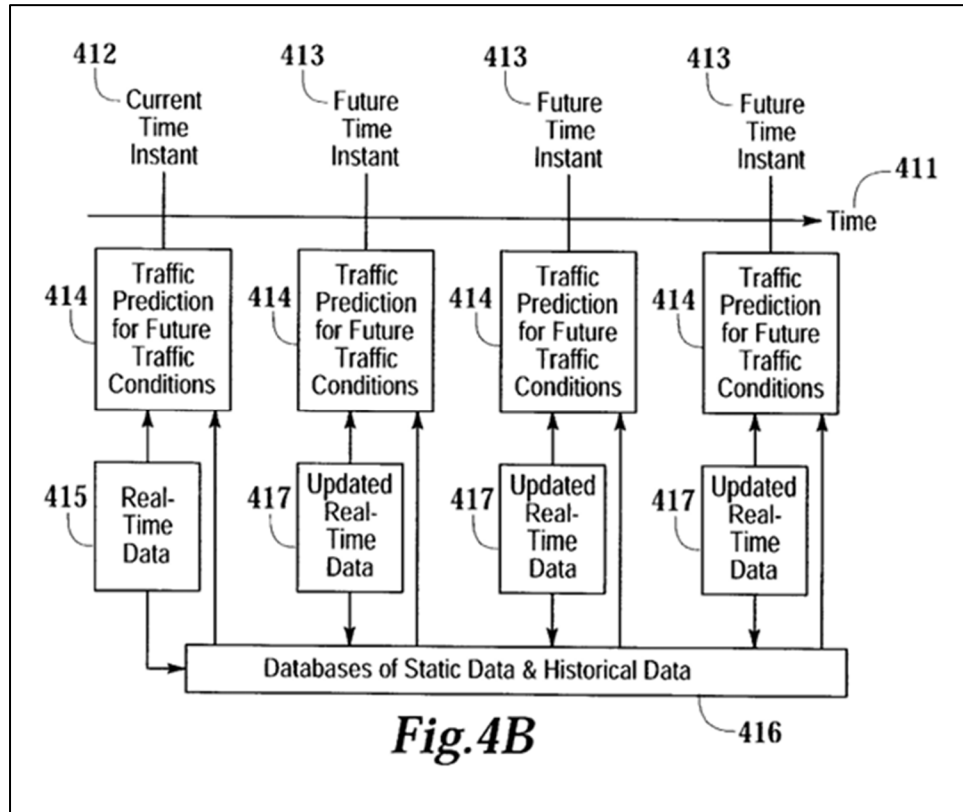
are typically provided by traffic reporting services, agencies[,] and field data collectors,” the “real-time travel speed and travel time 28 data are directly measured using field devices 212, such as detectors and sensors, or provided by probe vehicles 212,” the “real-time weather reports and forecasts 214 are available from weather reporting services 215,” the “real-time and planned events information 217 is available from various public and private organization 216, and the real-time and planned construction activity information 218 is typically available from transportation agencies and construction firms 219.” *Id.*, 18:5-17.

75. The data from these sources is fused together using a data fusion process, which can use “various types of models, including weighted average and fuzzy logic.” *Id.*, 18:29-31.

76. Ran describes using this combination/fusion of historical traffic data and updated real-time traffic data to iteratively predict future traffic conditions for segments of a route. *See, e.g., id.*, FIG. 4B below, 19:19-22 (“traffic prediction models” have the roles of “generat[ing] estimates of route segment speeds and travel times for the current time instant” and “produc[ing] prediction of route segment speeds and travel times for future time instants”), 20:1-17 (when “refin[ing] the short-term traffic predictions, all real-time data can be applied, including real-time travel time and speed data from field devices and probe vehicles 52, real-time

incident data 213, real-time and planned construction activity information 218, real-time and planned event information 217, and real-time weather reports and forecasts 214” where the “real-time data [is] retrieved” via the “wireline and wireless Internet”).

77. In particular, “the traffic forecast server 6, uses one or a plurality of traffic prediction models 8 to generate predicted speeds and travel times 9 for a route segment network.” *Id.*, 17:12-16.



TESLA-1006, FIG. 4B

78. “The traffic prediction models 8 have two roles: 1) generate estimates of route segment speeds and travel times for the current time instant, and 2) produce prediction of route segment speeds and travel times for future time instants.” *Id.*, 19:19-22.

79. As an initial matter, Ran clarifies that the “route segment[s]” is “representative of a particular path between destination points,” where the “destination points” are along the route. *Id.*, 1:66-2:3. “As indicated on the time 411 horizon, the traffic prediction procedure is executed for the current time instant 412 in order to provide the traffic prediction for future traffic conditions 414. In this prediction, the real-time data 415 are used to modify the historical traffic data in order to provide a better estimate and forecast for the short-term traffic conditions. In this forecast process, the databases of static data and historical data 416 are used as the basis and are updated using the real-time data 415. As time moves on to a future time instant 413, the real-time data 415 for the current time instant 412 become outdated and the updated real-time data 417 have to be used to modify the historical traffic data in order to provide a better estimate and forecast for the short-term traffic conditions.” *Id.*, 19:43-49.

80. Ran discloses at least four different types of prediction models: “1) simulation models 42, 2) dynamic traffic assignment models 43, 3) statistical models

44, and 4) heuristic models 45.” *Id.*, 18:53-56. One example of a model provided by Ran is the Greendshields model, which proposed a “fundamental speed-density relationship” where speed is a function of traffic density. *See id.*, 11:23-42. The Greendshields formula is expressed as follows:

$$V=V_o(1-D/D_j) \quad (1)$$

where V: speed (vph)

$V_o$ : freeflow speed based on the route segment classification (vph)

D: density or number of vehicles per mile (vehicles/mile)

$D_j$ : jam density or maximum number of vehicles per mile (vehicles/mile)

81. The Greendshields formula states that the traffic speed on a route segment can be determined in part by the number of vehicles or the vehicle density, and the route segment capacity (*i.e.*, the “maximum number of vehicles per mile”). *Id.* A vehicle speed can thus be predicted in light of traffic conditions by applying a function of the existing traffic conditions to a free flow speed for the route segment. *Id.*

82. In some instances, the predicted vehicle speed can be sent to various devices, including an “in-vehicle device 18.” *Id.*, 17:25-29. In other instances, Ran further teaches that the entire “multi-modal travel prediction and support system” can be located in an “in-vehicle computer.” *Id.*, 23:28-33.

83. Ran is analogous art to the '426 Patent: it is from the same field of endeavor (*e.g.*, vehicle speed prediction) and is reasonably pertinent to the problem the '426 Patent was trying to solve (*e.g.*, problem of how to determine speeds at which a vehicle should travel based, *e.g.*, on road or operating conditions).

### **3. Combination of Neiss and Ran**

84. As described in §VII(A)(1) *supra*, Neiss describes a predictive cruise control (PCC) system implemented on a vehicle's onboard computer that iteratively calculates the vehicle's optimized velocity for route segments (which Neiss refers to as frames) along the route ahead of the vehicle. Neiss further seeks to "adapt the velocity control to changing driving situations which result from intervention of external constraints on driving behavior as the vehicle transits a particular route," as "might be caused... for example, by a slower vehicle driving ahead." TESLA-1005, ¶¶[0010], [0167].

85. Ran similarly discloses that a vehicle's speed on segments of a particular route may be constrained by traffic on those route segments, and provides systems and methods for computing a vehicle's predicted or probable speed that accounts for real-time traffic conditions. *See, e.g.*, TESLA-1006, 11:23-42, 7:17-29, FIG. 3, 18:18-49; §VIII(A)(2) above.

86. As described below, a POSITA would have been motivated to combine the teachings of Neiss and Ran and would have had a reasonable expectation of success in doing so, to arrive at a combined system (referred here as Neiss-Ran) that improves already-optimized speeds for upcoming segments of the route (as Neiss discloses) using predicted speeds (*probable speed*) that account for real-time traffic conditions (as disclosed in Ran) at those upcoming route segments. Specifically, Neiss-Ran leverages Ran's teachings of prediction model(s) using, at least in part, real-time traffic information (e.g., as determined using field devices and external databases), to compute/calculate predicted speeds for upcoming route segments. TESLA-1006, 11:23-42, 21:42-49. In Neiss-Ran, Neiss's optimization algorithm would be modified by leveraging Ran's real-time traffic-based predicted speeds (*calculat[ed] probable speed*) for upcoming road segments/frames—instead of using current velocity of an existing frame, as Neiss teaches—to adapt the computed optimized values for the upcoming segments/frames to obtain improved speeds (*efficient speed*) that can be used to control the vehicle.

87. A POSITA would have been motivated to implement Ran's teachings in combination with Neiss for the following reasons.

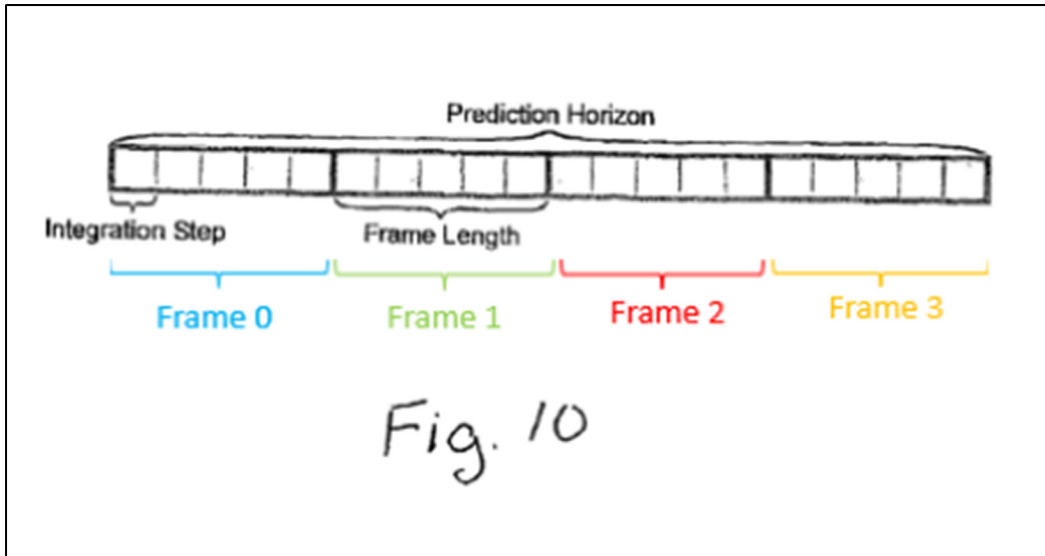
88. ***First***, Neiss seeks to use real-time driving conditions to adapt its optimization algorithm (*see, e.g.,* TESLA-1005, ¶[0163], Abstract, ¶¶[0009]-

[0012]), which provides an explicit motivation to look to Ran’s teachings of computing predicted speeds in upcoming road segments using real-time traffic information in those segments. Indeed, a POSITA would have been motivated to perform Neiss’s optimization for upcoming frames using Ran’s predicted speed values based on real-time traffic conditions in those frames—instead of using a current speed for an existing frame, as Neiss discloses—to account for more accurate, prevailing speeds in those frames, which in turn would enable a more accurate and realistic speed optimization (e.g., by accounting for real-time traffic conditions).

89. In more detail, Neiss’s predictive cruise control (PCC) system implements an optimization algorithm that seeks to reduce fuel consumption. TESLA-1005, ¶¶[0092]-[0098], [0178]-[0191]; §VIII(A)(1) *supra*. Using this algorithm, Neiss discloses computation of optimal vehicle control parameters, including optimal velocity, for segments or frames of the upcoming route, and stores these computed parameters in a look-up table. TESLA-1005, ¶¶[0092]-[0098], [0178]-[0191]; §VIII(A)(1) *supra*. As part of its iterative optimization process, Neiss explains that it adapts the control values for the upcoming frames in the lookup table using the “current velocity” of the vehicle. *See, e.g.*, TESLA-1005, ¶¶[0167], [0190]. In one example, Neiss discloses that the current velocity of the vehicle for

a particular road segment/frame is used during the optimization as a starting velocity for the upcoming frames/segments of the route. *Id.*

90. A POSITA, however, would have recognized that the current velocity for the frame/segment in which the vehicle is currently being driven does not necessarily reflect the vehicle's likely velocity for upcoming frames/segments because, e.g., traffic conditions might be different in the upcoming frames/segments may be different from those in the current frame/segment. Indeed, using the following figure from Neiss for illustrative purposes, a POSITA would have recognized that traffic in the segments ahead of the vehicle (e.g., in frame 3) could be significantly different from that in frame 1, such that the speed with which vehicles are driving in frame 1 may be higher or faster relative to the speed with which vehicles are driving in frame 3.



*TESLA-1005, FIG. 10 (annotated)*

91. Before the Critical Date, it was well known to use real-time traffic conditions to predict a vehicle's expected speed as it traverses a particular route. This is corroborated by, e.g., Kuroda TESLA-1011; 6:59-7:34 (teaching that the "information of the route to the destination," including "traffic jam information" and "traffic regulation information" is used in the prediction of vehicle speeds). Indeed, as described above, Ran provides one such solution in which a traffic prediction models are used to calculate the predicted speeds for segments of the route using real-time traffic information and subsequently, these predicted speed values are provided to an in-vehicle computer of the vehicle. *See* §VIII(A)(2) *supra*; TESLA-1006, 11:23-64, 17:25-18:61, 21:43-49, 1:12-36, 4:9-29, 7:1-5, Tables 1-3, 14:8-15, 18:59-61, 20:9-21:49.

92. In particular, Ran discloses that historical and real-time traffic and speed data are provided as inputs to one or more traffic prediction models, to generate “predicted speeds” for route segments of a route. *Id.* Ran further discloses communicating the predicted speed information to an in-vehicle computer via wireless or wireline communication mechanisms. *Id.*, 17:25-27.

93. In view of Ran’s well-known use of vehicle speed prediction for upcoming road segments based on real-time traffic conditions, a POSITA would have been motivated to modify Neiss’s use of a current velocity of an existing frame when optimizing vehicle control parameters (including vehicle speed) for upcoming frames, to instead perform the optimization using predicted speeds that account for real-time traffic conditions (as taught by Ran) in those upcoming frames (instead of using current velocity of an existing frame). That is because doing so would yield a more accurate and realistic speed optimization given that the optimization would account for the more accurate, prevailing speeds for upcoming frames/segments of the route for which optimizations are being performed.

94. A POSITA would have thus modified Neiss’s optimization algorithm by leveraging the real-time traffic-based predicted speeds (as disclosed in Ran) for upcoming road segments/frames—instead of Neiss’s existing use of the current

velocity of an existing frame—to adapt Neiss’s computed optimized values for these segments/frames.

95. As a result, and using the above figure as an example, when optimizing the vehicle control parameter values (including the optimal vehicle velocity) for frames 1, 2, and 3, Neiss’s modified algorithm would adapt these computed optimal values using the Ran-based predicted velocity based on real-time traffic information for these frames/segments, respectively. In one example, adapting the optimized velocity computations using the predicted speeds for the upcoming frames/segments would involve using the predicted speeds as start velocities for those segments/frames.

96. **Second**, a POSITA would have been motivated to combine these references’ teachings to enable more resource efficient optimization calculations. Specifically, Neiss acknowledges that its optimization iterations are computing resource intensive (TESLA-1005, ¶¶[0169]) and a POSITA would have sought to avoid or reduce scenarios where the actual “vehicle velocity drifts significantly from the velocity stored in the look-up table for the current position,” in which case an updated resource-intensive optimization iteration may need to be performed or the system would switch to a failure mode and stops all computations. See TESLA-1005, ¶¶[0167], [0190]. Using real-time traffic and speed information at upcoming

segments (as disclosed in Ran) to adapt the optimized speeds for those segments in Neiss's algorithm, therefore avoids or reduces the likelihood of scenarios where the resource-intensive optimization cycles fail or are repeated.

97. ***Third***, a POSITA would have been motivated to pursue this implementation because doing so is merely the application of known techniques (*e.g.*, using estimated/predicted velocities for an upcoming segment for a route (*see, e.g.*, Ran (TESLA-1006) and TESLA-1007, 1:52-56 (generating a speed plan for an upcoming road section)) to a known system (*e.g.*, Neiss's system that uses current velocity of an existing frame to adapt the computed optimal velocity computations for upcoming frames/segments) to yield predictable results (*e.g.*, computation and application of optimal control values that are adapted using real-time traffic-based speeds, thereby accomplishing Neiss's goal of online adaptation of optimal control parameters in view of real-time conditions).

98. A POSITA would have also had a reasonable expectation in combining Neiss's and Ran's teachings in the above-described manner for the following reasons. As an initial matter, each of Neiss and Ran's systems/components would continue to operate in the same manner they did prior to combination. Specifically, Neiss's optimization algorithm would continue to be implemented by its onboard computer and would operate iteratively on road segments/frames ahead of the

vehicle. TESLA-1005, Abstract, ¶¶[0015], [0033]-[0049], FIGS. 4-6, ¶¶[0167]-[0169], [0177]-[0191], FIGS. 12a-c, FIGS. 13a-d. Ran's predicted speed computation would continue to be performed by Ran's traffic prediction models (as easily implemented at another computing device, like an in-vehicle computer, such as Neiss's onboard computer) using, at least, real-time traffic data from real-time data sources. *See* TESLA-1006, 17:7-16, 18:18-31. A POSITA would have recognized that it would be straightforward and well-within the skillset of a POSITA to implement "traffic prediction models 8" at an in-vehicle computer that would be configured in a manner similar to the computing device in Ran on which such model(s) operate and that generate predicted speeds using data from "input databases" and data fusion algorithms.

99. Moreover, Neiss already discloses iterative calculations of optimized velocities for route segments ahead of the vehicle, and further acknowledges the need to adjust or adapt the calculated velocities to account for real-time conditions. *See* §VIII(A)(1) *supra*. And, Ran teaches calculating expected or predicted velocities for upcoming segments of the route using real-time traffic conditions. *See* §VIII(A)(2) *supra*. As such, it would have been straightforward and well-within the skillset of a POSITA to configure Neiss's onboard computer and the associated optimization algorithm so as to substitute its use of current velocity value for an

existing segment/frame with the real-time traffic-informed, more accurate velocity for the frame (as Ran teaches) for which the optimization is being performed. Beyond this modification, a POSITA would have recognized that Neiss’s algorithm would continue to operate in the manner already disclosed in Neiss.

#### 4. Claim Analysis

##### (a) Claim 1

*[1pre] A method of managing the power consumption of a vehicle, comprising:*

100. If the preamble is determined to be limiting, Neiss-Ran teaches the preamble for the reasons described below.

101. As explained below, Neiss-Ran leverages Neiss’s teachings and algorithm for minimizing fuel consumption of a vehicle, which, in the context of the ’426 Patent and the relevant technical field, would be understood by a POSITA to correspond to *managing the power consumption of the vehicle*.<sup>2</sup> For example, the ’426 Patent states that “optimizing the power supplied to the engine typically means minimizing the energy loss from the vehicle, thereby increasing fuel efficiency.” TESLA-1001, 6:14-17. Thus, in the case of fuel-powered vehicles, *managing the power consumption* refers to minimizing fuel consumption at the vehicle. *See id.*,

---

<sup>2</sup> Claim language is denoted in bold italics throughout this declaration.

6:14-30 (also noting that the claimed invention can apply to all appropriate vehicles, including fuel-powered vehicles, hybrid vehicles, and electric vehicles). In the relevant field, a POSITA would have recognized that fuel is a power source and managing power consumption in a vehicle would encompass managing the consumption of the power source(s) for the vehicle, such as fuel in the case of fuel-powered vehicles. This is further corroborated by, e.g., Hosoda (TESLA-1009, ¶¶[0015], [0025], FIG. 10 (explaining that hybrid vehicles are powered by both power supplied from a battery and the force of liquid fuel, such as gasoline, which collectively constitute a “power unit for generating driving force”; recognizing the need to monitor both via a “fuel and power storage remaining amount display 210”)).

102. In particular, Neiss describes a “predictive cruise control system [that] utilizes information about the current vehicle position, as well as upcoming terrain in order to save fuel and increase driving comfort.” TESLA-1005, Abstract. Specifically, Neiss’s predictive cruise control system minimizes a vehicle operating cost function, “based on an analysis of vehicle operating parameters and route information regarding a route section ahead of the vehicle.” TESLA-1005, ¶¶[0001], [0015], [0081]-[0082]. Indeed, Neiss explains that in optimal control theory, a cost function is used to define a desired operating behavior of the system and minimizing

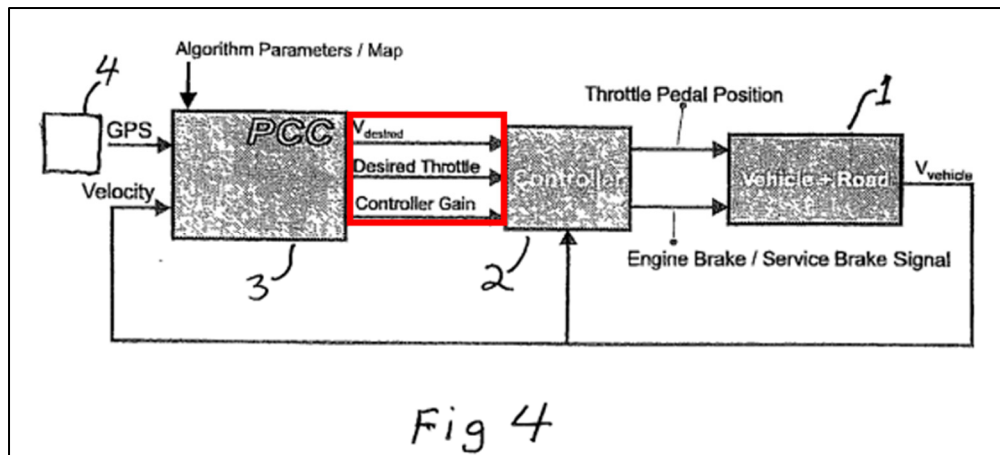
the cost function achieves the optimal or desired system operation. TESLA-1005, ¶[0081].

103. In one embodiment, Neiss’s predictive cruise control system identifies “vehicle control parameters,” such as speed and throttle position, that “minimize[] vehicle operating costs,” *e.g.*, optimize vehicle operation, “based on an analysis of vehicle operating parameters and route information regarding a route section ahead of the vehicle.” TESLA-1005, ¶¶[0001], [0015]. For instance, one goal of Neiss’s optimization is to reduce fuel consumption. *See id.*, ¶¶[0081]-[0082] (“the following goals were established to define ‘optimal operation’: The vehicle should consume less fuel and should activate the brakes as little as possible”), Claims 1 and 10 (“iteratively calculating ... vehicle control parameters that minimize said vehicle operating cost function ... wherein said cost function includes vehicle fuel consumption as a function of vehicle speed, road gradient, and throttle position”), ¶¶[0014] (“the predictive cruise control system according to the invention ... utilizes information about the current vehicle position, as well as upcoming terrain in order to save fuel and increase driving comfort.”), [0004].

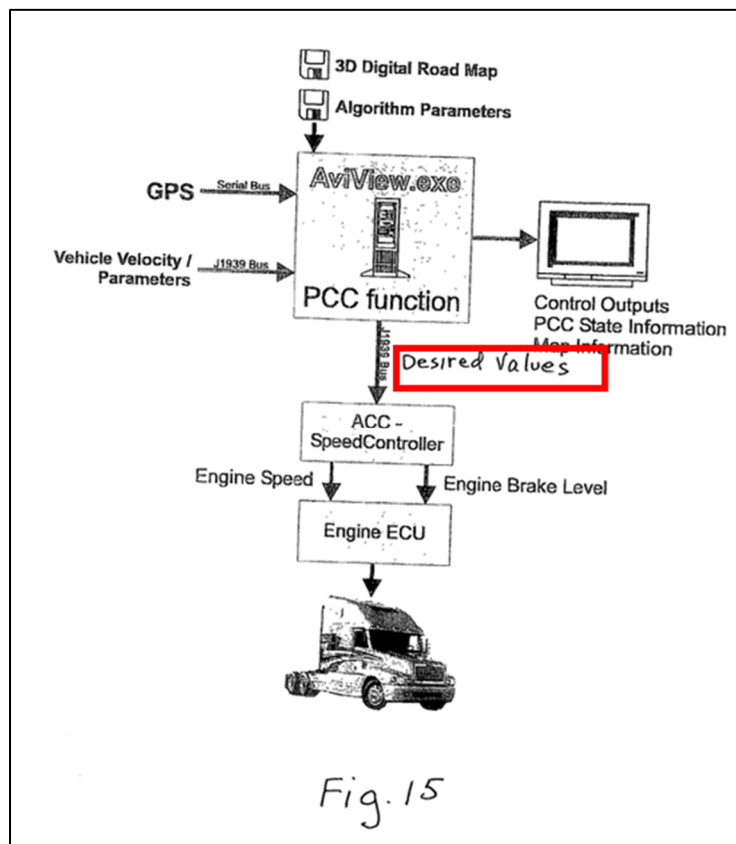
104. Moreover, Neiss describes controlling the vehicle according to the vehicle parameters determined by the optimization of the vehicle operating cost function. Specifically, the calculated optimized vehicle control parameters are

stored in memory and subsequently used by a controller to control the vehicle in accordance with these vehicle control parameters. *See, e.g.*, TESLA-1005, ¶¶[0015] (“The vehicle is then controlled by reading the optimized vehicle control parameters from the memory, corresponding to the current position of the vehicle”), [0014] (“The vehicle velocity is then controlled to follow an analytically computed optimal driving strategy”), Claim 1 (“controlling current vehicle operation based on said optimized vehicle control parameters”), ¶[0042].

105. Neiss achieves this control by “convert[ing]” “the desired vehicle speed” (as computed by its optimization algorithm) to “a desired engine speed, which is sent to the engine and the engine achieves it via its own controller or using a separate adaptive cruise controller,” *e.g.*, by injecting fuel or specifying a throttle angle. TESLA-1005, ¶¶[0042]-[0044]. As shown in Figures 4 and 15 below, the “desired values” (i.e., the calculated optimized velocity, throttle, and controller gain values) are applied to a speed controller, which applies the corresponding “engine speed” and “engine brake level” to the engine ECU to cause the vehicle to implement the output of Neiss’s predictive cruise algorithm at the vehicle.



TESLA-1005, FIG. 4 (annotated)



TESLA-1005, FIG. 15 (annotated)

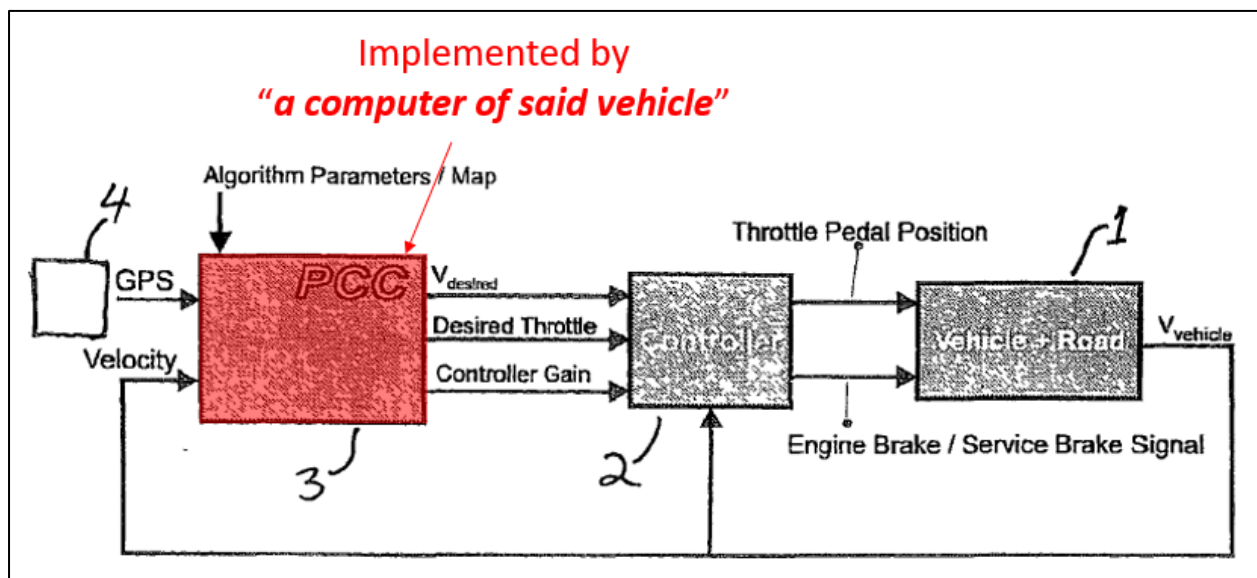
106. Thus, the computation of optimal vehicle control parameters (including vehicle velocity) that allow the vehicle to “consume less fuel,” and the subsequent control of the vehicle according to those parameters, corresponds to *managing the power consumption of the vehicle*, e.g., in that control parameters identified to minimize fuel consumption are used to control the operation of the vehicle engine.

***[1a] (a) receiving a route at a computer of said vehicle;***

107. Neiss-Ran teaches limitation [1a] for the reasons described below.

108. Neiss teaches a *computer of said vehicle*. Specifically, Neiss discloses an onboard computer that includes memory and a data processor. *See, e.g.*, TESLA-1005, Abstract (“an onboard computer iteratively calculates and stores in a memory vehicle control parameters that optimize the vehicle operating cost function for a predetermined Prediction Horizon along the route ahead of the vehicle”), ¶¶[0015], [0034], [0037]-[0041], [0044] (describing the predictive cruise control (PCC) block as computing the optimal vehicle control parameters, such as desired speed, throttle, and controller gain), FIGS. 4, 15 (showing and describing PCC block implemented at the vehicle), Claim 15 (describing a data processor to calculate optimum vehicle control parameters and memory for storing road map and a lookup table including the calculated optimal vehicle control parameters).

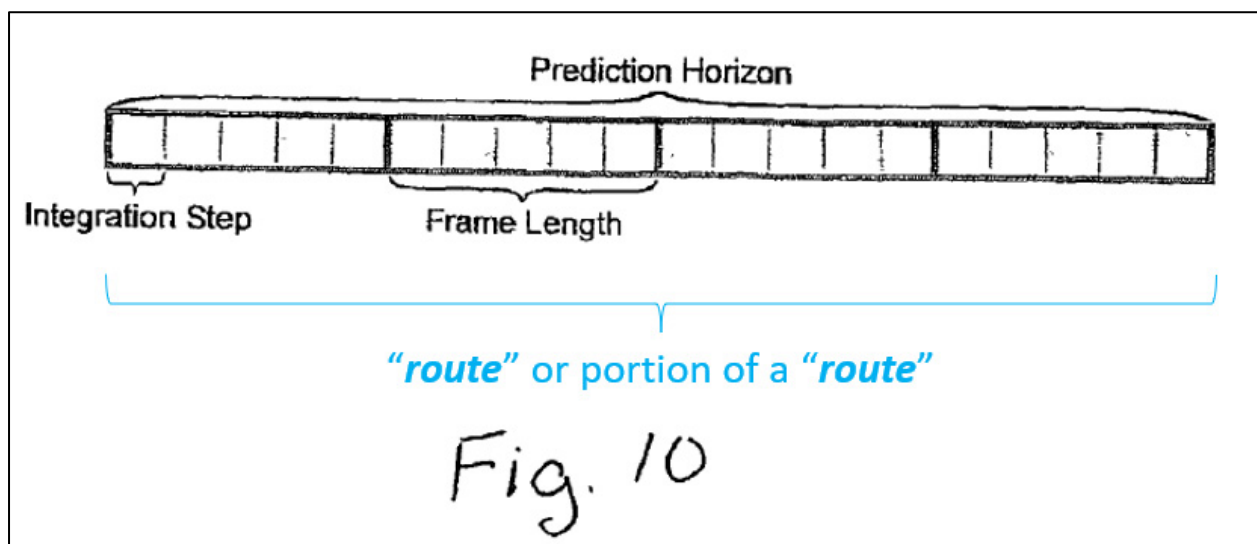
109. Neiss's PCC block executes the optimization algorithm, and, because Neiss's onboard computer is also described as executing the optimization algorithm, the PCC block would be understood as being part of the onboard computer. Neiss's "PCC vehicle control system," as implemented by Neiss's onboard computer, includes "a PCC block 3" that "generates as outputs" the optimized velocity, throttle, and controller gain values, and provides these values as "inputs for the controller 2 which controls the vehicle to follow the calculated velocity trajectory." TESLA-1005, ¶¶[0034], [0037]-[0041]; see FIG. 4, below.



TESLA-1005, FIG. 4 (annotated)

110. Neiss's optimization algorithm "iteratively calculat[es] ... vehicle control parameters that optimize the vehicle operating cost function for a predetermined distance (referred to as a Prediction Horizon) along the route ahead

of the vehicle.” TESLA-1005, ¶[0015]; *see id.*, ¶¶[0001] (“a cruise control unit for a motor vehicle, which minimizes vehicle operating costs, based on an analysis of vehicle operating parameters and route information regarding a route section ahead of the vehicle”), [0049], Abstract, Claim 1 (“iteratively calculating and storing in a memory, vehicle control parameters that minimize said vehicle operating cost function for a predetermined distance along said route ahead of said vehicle”).



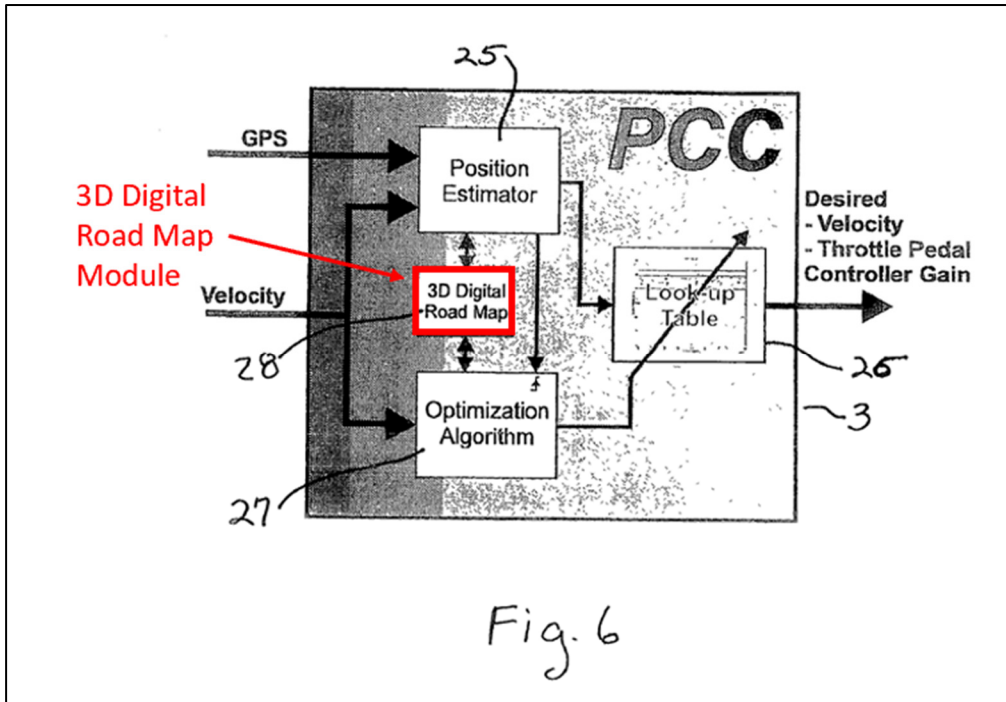
*TESLA-1005, FIG. 10 (annotated)*

111. The Prediction Horizon (shown above in blue; *see* FIG. 10, above) corresponds to *a route*, *e.g.*, because it constitutes a path being traveled by the vehicle. Additionally, or alternatively, Neiss discloses that the vehicle traverses an overall route (*e.g.*, one with a starting point and an ending point or final destination where the vehicle stops) and that its optimization operations are performed

iteratively for portions of the overall route (which Neiss refers to as the Prediction Horizon) ahead of the vehicle as the vehicle continues to travel the route to its final destination. TESLA-1005, Abstract, ¶¶[0001], [0015]. As such, a POSITA would have understood that Neiss discloses a vehicle traversing an overall route, with optimizations performed on each iteratively updated Prediction Horizon (i.e., predetermined distance) ahead of the vehicle as it travels this route. *See id.*

112. Moreover, and as explained below, Neiss-Ran teaches ***receiving a route*** (e.g., receiving either the immediate Prediction Horizon ahead of the vehicle or the route represented by a starting position and an ending position or final destination) ***at a computer of said vehicle*** (e.g., onboard computer in Neiss’s vehicle).

113. In particular, Neiss’s PCC block that is implemented on the onboard computer includes a “3D Digital Road Map.” TESLA-1005, Abstract, ¶¶[0004]-[0005], [0014]-[0015], [0049], [0179], FIG. 6 (reproduced below).



*TESLA-1005, FIG. 6 (annotated)*

114. Neiss further discloses that, as the vehicle travels a “particular route,” the PCC block accesses information about the route ahead of the vehicle using this 3D Digital Road Map. TESLA-1005, ¶¶[0049] (“3D Digital Road Map 28: The Optimization Algorithm [of the PCC] ... uses information read from the digital map to determine the gradient angles and curve radii of the upcoming road”), [0004] (“the current vehicle position may be determined, for example, by means of a Global Positioning System (GPS), and information about the upcoming terrain can be taken from a three-dimensional digital road map for the purpose of controlling vehicle operation”), [0014]-[0015] (same), [0049], [0179], Claim 15 (“wherein said predictive cruise control module comprises: a first memory having stored therein a

road map including route parameters that characterize routes within a vehicle operating area; [and] a data processor programmed to calculate said optimum vehicle control parameters, based on ... said route parameters”).

115. In particular, Neiss describes that the current position of the vehicle is used to query the 3D Digital Road Map to identify the area “along the route” ahead of the vehicle (which Neiss refers to as the Prediction Horizon) and access information about the identified Prediction Horizon (e.g., “gradient angles and curve radii for the upcoming road”) therefrom. *See id.* Neiss’s optimization algorithm thus uses this “[f]orward-looking terrain information” “for a predetermined distance (referred to as a Prediction Horizon) along the route ahead of the vehicle” to “determine, as a function of position, a desired speed,” e.g., the onboard computer uses information about upcoming terrain (of the Prediction Horizon) to calculate vehicle control parameters. *Id.*, ¶¶[0031], [0015]; *see id.*, Claim 15 (“wherein said predictive cruise control module comprises ... a data processor programmed to calculate said optimum vehicle control parameters”), ¶[0162].

116. Further still, Neiss teaches that its optimization algorithm operates “iteratively” to identify subsequent Prediction Horizons ahead of the vehicle (as the vehicle continues driving) and accesses each such Prediction Horizon and associated road information from the 3D Digital Road Map, which are then used by the

optimization algorithm to iteratively compute the optimal vehicle control parameters. TESLA-1005, ¶¶[0012], [0014], [0031], [0045]-[0049], [0163], [0167], [0181], [0186], [0190], FIGS. 12a-c, FIGS. 13a-d.

117. From these disclosures, a POSITA would have understood that the onboard computer (which implements Neiss's PCC block) *receiv[es] a route* (e.g., receives a Prediction Horizon, or receives the set of Prediction Horizons that constitute a complete, origin-to-destination route) because Neiss's optimization algorithm (1) uses the 3D Digital Road Map to identify (*receive*) each Prediction Horizon ahead of the vehicle (where the Prediction Horizon is itself a *route* or a portion of the *route* formed of successive Predictions Horizons, each of which is received until the vehicle reaches its final destination) and (2) performs optimization calculations using the information about this route (e.g., road curvature and gradient information) as received from the 3D Digital Road Map.

118. If it is determined that the present limitation requires receiving an *entire* route (i.e., an end-to-end route with a starting point to a final destination) *at once*, a POSITA would have found that to be obvious in view of Neiss-Ran. As described above, Neiss-Ran's onboard computer iteratively receives predetermined distances of the route ahead of the vehicle, as the vehicle drives from a starting point and until it reaches its final destination. A POSITA would have found it obvious to modify

Neiss-Ran so that Neiss-Ran's onboard computer receives the entire end-to-end route all at once, instead of receiving it iteratively in the manner described by Neiss.

119. A POSITA would have been motivated to modify Neiss-Ran in this manner for at least the following reasons.

120. **First**, receiving the entire route at once would be computationally efficient because it would enable quicker computations on upcoming route segments/frames by avoiding or reducing use of network and computational resources that would otherwise be needed to obtain this information as the vehicle drives and thereafter perform the computations on such received information.

121. **Second**, a vehicle onboard computer receiving an entire end-to-end route (e.g., from external map applications) was well-known in the context of vehicle control systems and enabled receipt of accurate routes that, e.g., account for real-time events, such as road and/or traffic conditions. This is corroborated by, e.g., Kuroda (TESLA-1011, 1:42-43) (teaching that a vehicle can include a "route retrieving unit [that] retriev[es] a route to a destination"). Thus, modifying Neiss-Ran's onboard computer to receive the route would have been an application of known techniques (e.g., an onboard computer of a cruise control system receiving a route) to a known system (e.g., a vehicle onboard computer implementing a predictive cruise control system that processes information about the route) to yield

predictable results (e.g., an onboard route computer for predictive cruise control that receives an entire route with respect to which optimizations and/or analytic operations can be performed).

122. A POSITA would have also had a reasonable expectation of success in configuring Neiss-Ran in this manner. That is because (1) it was well-known and conventional for vehicle onboard computer to receive routes, e.g., from map applications (e.g., using user input specifying a start position and a final destination, and a map application including routing logic that identifies and provides to an onboard vehicle computer the identified route (*see, e.g.*, Hafner (TESLA-1027, 7:52-54 (teaching that the driver “informs the system of her[] destination,” and the “system access all available prior information about this destination and various routes to the destination”))), (2) Neiss already includes a digital map that stores information about routes that the vehicle drives (e.g., TESLA-1005, ¶¶[0049], [0031], [0015], [0162], Abstract), and (3) Neiss’s optimization algorithm already obtains/receives information about the route ahead of the vehicle for its optimization calculations (*see id.*). Thus, it would have been straightforward and well within a POSITA’s skillset to configure Neiss’s map application (e.g., the 3D Digital Road Map) with such conventional routing logic that enables identification of the entire route and enables Neiss’s onboard computer (and its associated optimization

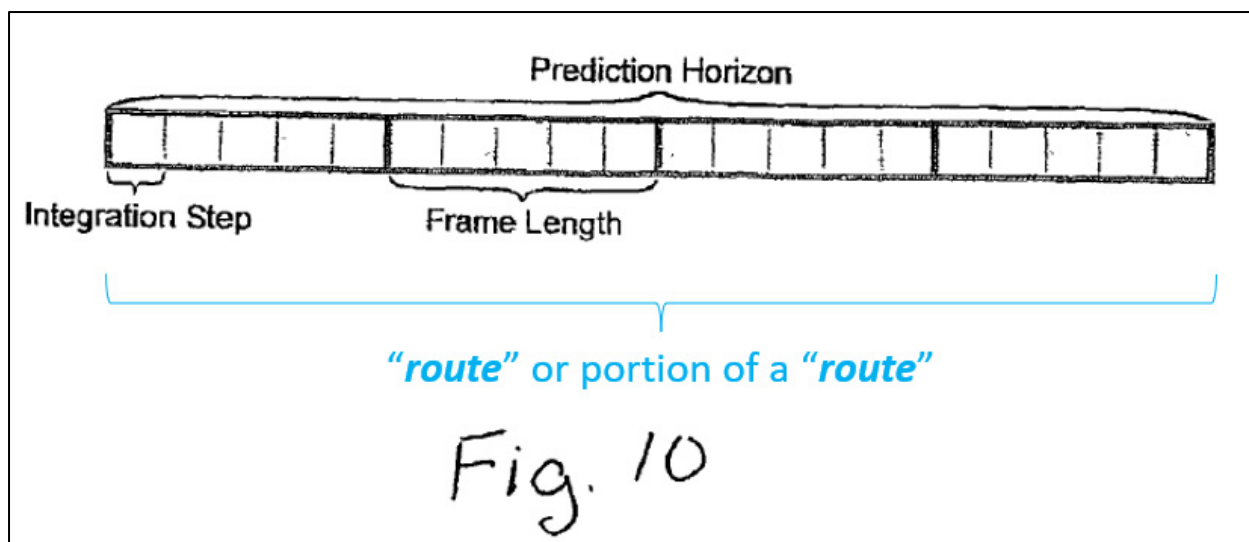
algorithm) to receive the entire route from the map application (e.g., using the same mechanism by which the optimization algorithm already communicates with the 3D Digital Road Map).

***[1b] (b) segmenting said route into a plurality of segment destinations;***

123. Neiss-Ran teaches limitation [1b] for the reasons described below.

124. As an initial matter, the '426 Patent's specification does not use the phrase "segment destination." *See generally* TESLA-1001. Moreover, while the specification describes segmenting a route, it describes segmenting the route into segments (and not segment destinations, as claimed). *See, e.g., id.*, 13:38-48 (stating that a "route may be broken up into segments," where a "segment may comprise any distance to be traveled, including the entire route, or small portions of the route"). Thus, in the context of the '426 Patent's specification, a POSITA would have understood the present limitation to encompass segmenting a route into a plurality of segments, where each segment has an endpoint/destination.

125. Similarly, Neiss discloses segmenting the route into segments, each segment having an endpoint or destination. In particular, and as explained below, Neiss discloses segmenting the route into frames, where each frame has an endpoint. *See, e.g.*, TESLA-1005, ¶[0170], FIG. 10.



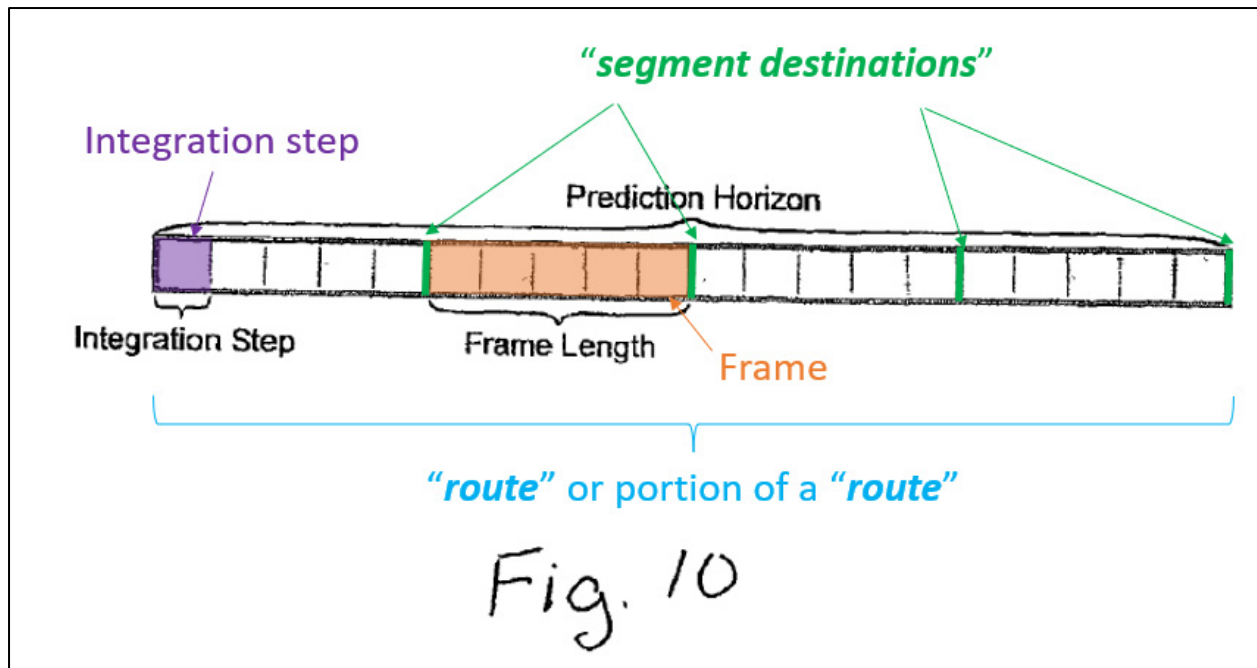
*TESLA-1005, FIG. 10 (annotated)*

126. As described for limitation [1a], Neiss “iteratively calculates and stores in a memory vehicle control parameters that optimize the vehicle operating cost function” as the vehicle proceeds along a route. *Id.*, Abstract; *see* §VIII(A)(4)[1a] above. Each optimization iteration is performed for a “predetermined distance ... along the route ahead of the vehicle,” which Neiss refers to as the “Prediction Horizon.” *Id.*, ¶[0015].

127. Also, as explained for limitation [1a], the “Prediction Horizon” immediately ahead of the vehicle corresponds to the claimed **route** or alternatively, the claimed **route** is made up of successive Prediction Horizons from the vehicle’s start location to its final destination. As a vehicle travels along the route, the Prediction Horizon is iteratively updated to reflect a new pre-determined distance of

the route ahead of the vehicle. TESLA-1005, ¶[0047] (clarifying that the “Prediction Horizon” is “an area ... around the current vehicle position,” which changes as the vehicle continues to traverse the route). In this manner, Neiss’s iterative algorithm would be understood as operating on portions of the route ahead of the vehicle until the vehicle’s final destination is reached.

128. Neiss further describes that each Prediction Horizon (i.e., the route or a subset of the route) “is separated into Frames.” TESLA-1005, ¶[0170]; *see id.*, ¶[0192], FIG. 10, below. As shown below, each frame (highlighting one frame in orange in FIG. 10) is “a specified distance,” i.e., a portion (segment) of the route. TESLA-1005, ¶[0169]. Each frame in turn is composed of multiple “integration steps” (one of which is highlighted in purple below), with each integration step corresponding to a “small distance[]” along its frame. *Id.*, ¶¶[0162], [0170]. The endpoint of each frame in the Prediction Horizon corresponds to a ***segment destination*** (noted in green) given that it is the end (destination) of a frame (segment) of the Prediction Horizon.

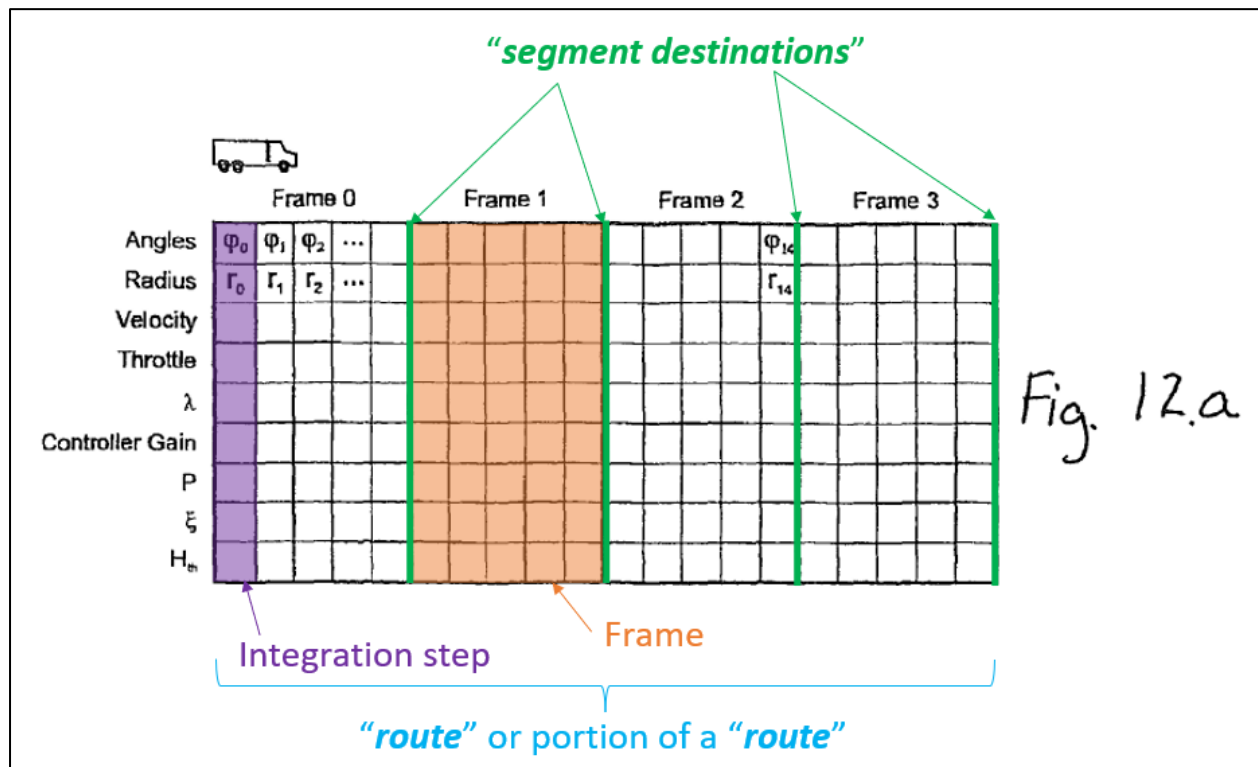


TESLA-1005, FIG. 10 (annotated)

129. Each Prediction Horizon ahead of the vehicle (*said route*) is thus *segment[ed] ... into segment destinations* at least because each Prediction Horizon is composed of a plurality of frames (*segment[s]*), each having a respective endpoint (*segment destinations*). In the above example of FIG. 10, “the Number of Frames is 4,” but Neiss acknowledges that in real-world implementations, the number of frames in a Prediction Horizon can be substantially more, e.g., 4000). TESLA-1005, ¶¶[0169]-[0170], [0192].

130. In more detail, Neiss-Ran’s optimization algorithm is implemented using a look-up table “that corresponds to the length (in frames and integration steps)

of the Prediction Horizon.” TESLA-1005, ¶[0171]; *see id.*, Claim 18 (“said ring buffer memory is divided into a plurality of frames which collectively correspond to said prediction horizon, each frame including a predetermined number of cells corresponding to said integration interval”), FIGS. 12a(below)-c, FIGS. 13a-13d. The locations in the lookup table that correspond to the endpoints of the frames of the Prediction Horizon likewise correspond to the *segment destinations* of said *route*.



TESLA-1005, FIG. 12a (annotated)

***[1c] (c) said computer device for each of said segment destinations: [performing operations recited in steps (i)-(iv)]***

131. Neiss-Ran teaches limitation [1c] for the reasons described below.

132. As explained below, in Neiss-Ran, the optimization algorithm is performed by the PCC block of the onboard computer (i.e., ***said computer device***) for each frame in the route ahead of the vehicle (i.e., ***for each*** segment and its corresponding ***segment destination***). Also, as explained above for limitation [1a], the PCC block is implemented on the vehicle onboard computer. See §VII(A)(4)[1a] above.

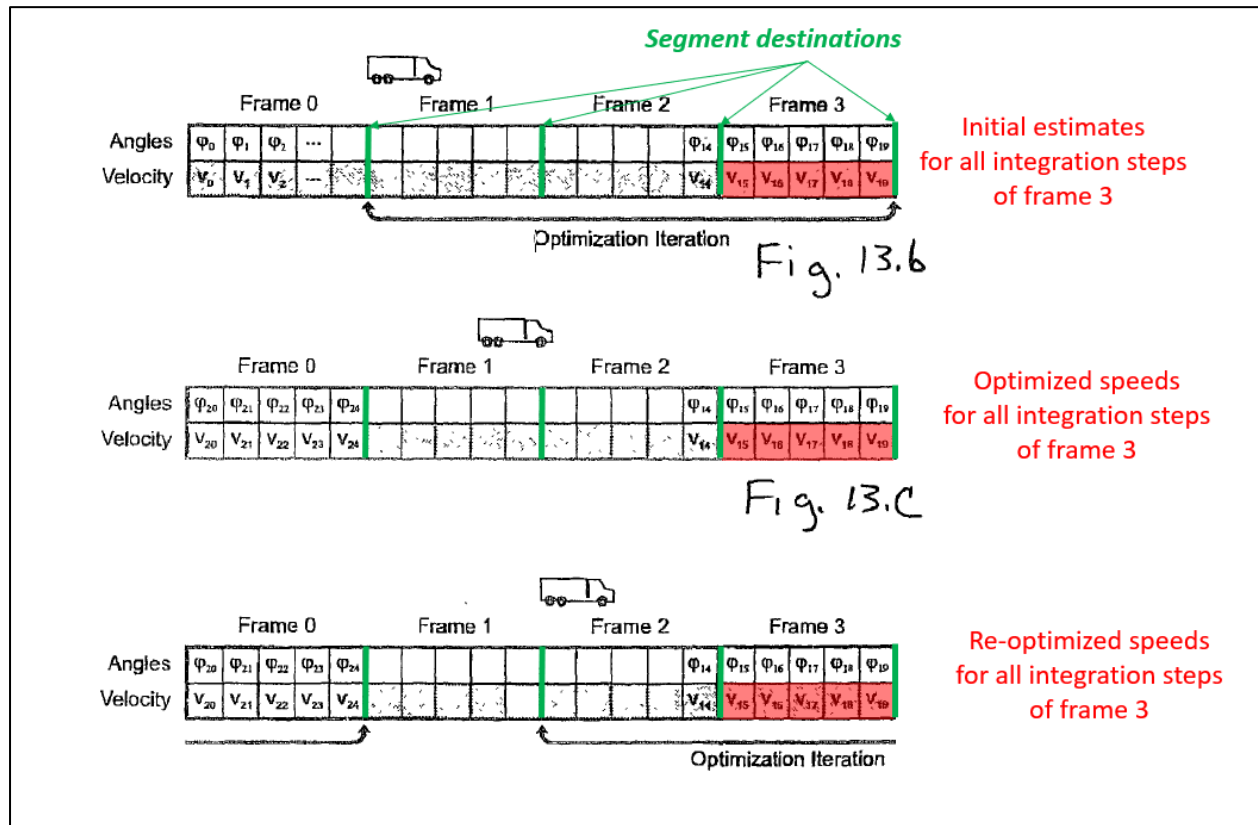
133. In more detail, Neiss discloses that the optimization algorithm is iteratively performed for each frame in a Prediction Horizon ahead of the vehicle. See, e.g., TESLA-1005, Abstract (“As the vehicle proceeds, an onboard computer iteratively calculates and stores in a memory vehicle control parameters that optimize the vehicle operating cost function for a predetermined Prediction Horizon along the route ahead of the vehicle. The optimal vehicle control parameters for the Prediction Horizon are then stored in a memory and continuously updated and replaced by new data as the vehicle moves along....”), ¶¶[0015] (same), [0169]-[0170] (“Rather, a new optimization calculation only occurs once the vehicle has covered a specified distance, which is referred to hereinafter as a ‘frame’.... The whole Prediction Horizon is separated into Frames, so that the Prediction Horizon

must be a multiple of the Frame Length (also measured in number of integration steps h)"). For example, Neiss explains that, at initialization, optimal vehicle control parameters are computed for first set of frames (e.g., frames 0, 1, and 2), and then once vehicle gets to frame 1, an optimization iteration is performed for another set of frames (e.g., frames 1, 2, and 3). *See, e.g., id.*, ¶¶[0178]-[0180], [0188]-[0192].

134. Moreover, and as explained above, Neiss's optimization algorithm is implemented using a look-up table that "is divided into a plurality of frames which collectively correspond to said prediction horizon, each frame including a predetermined number of cells corresponding to" the integration steps of frames ahead. TESLA-1005, Claim 18; *see also id.*, FIGS. 12a-c, FIGS. 13a-d. Additionally, each frame in the look-up table corresponds to a route segment and has an endpoint that likewise correspond to the ***segment destinations*** (as explained for limitation [1b]). *See* §VIII(A)(4)[1b] above.

135. As explained below in relation to limitations [1d1]-[1f], below, Neiss's system calculates "values for vehicle velocity, throttle position and controller gain" iteratively for each frame (and the integration steps making up the frame) and enters the values in the lookup table, until the calculation is complete for each frame (the endpoint of which is the ***segment destination***). TESLA-1005, ¶¶[0171], [0188], [0015]. Once the vehicle reaches the end of a particular frame, "a new calculation

is triggered” for each integration step in the upcoming frames along the route. *Id.*, ¶[0188]. As illustrated in FIGS. 13b-d below, with each optimization, Neiss’s algorithm computes and stores in memory, the optimal control parameters for traveling the entirety of each such frame (i.e., optimal values are computed for each integration step of the frame, e.g., from the first integration step of the frame to the last integration step of the frame that terminates at the endpoint of the frame, e.g., the *segment destinations*).



*TESLA-1005, FIGS. 13b-d (annotated) (showing optimal vehicle control parameters being generated for each integration step making up frame 3)*

136. In this manner, Neiss's PCC block implemented by its onboard computer (i.e., *said computer device*) iteratively executes its optimization algorithm for each frame in the Prediction Horizon, and computes optimized/re-optimized speeds for the entirety of each frame (i.e., from the first integration step of the frame to the last integration step that terminates at the endpoint/destination of the frame/segment).

137. And because each frame has an end point (i.e., the *segment destinations*)—as explained for limitation [1b]—Neiss's optimization algorithm is performed by its *computer device for each of said segment destinations* along the vehicle's travelled route. Additional details and analysis for how Neiss-Ran's computer device performs the claimed operations recited in limitations [1d1] to [1f] for each segment destination are described below with reference to those limitations.

***[1d1] (i) calculating an optimized speed by minimizing power consumption over said segment destination;***

138. Neiss-Ran teaches limitation [1d1] for the reasons described below and for the additional reasons described in [1d2] below.

139. As explained for the preamble, a POSITA would have understood that *minimizing power consumption*, as used in the claim, encompasses minimizing fuel consumption. As I explained above in [1pre], fuel is an example of a power source

and minimizing fuel consumption amounts to minimizing power consumption. Also, the '426 Patent relates fuel efficiency to power consumption in the context of fuel-powered vehicles, as explained above in [1pre]. *See* TESLA-1001, 6:14-30.

140. The combination leverages Neiss's teachings, which a POSITA would have understood as teaching *calculating an optimized speed by minimizing power consumption over said segment destination*.

141. As described below, Neiss-Ran's Optimization Algorithm Module of Neiss' PCC block (which is implemented by Neiss's onboard computer, i.e., *said computer device*) calculates a speed/velocity for each frame (i.e., a segment that has an endpoint or *segment destination*) that is optimized to minimize fuel consumption of the vehicle. *See, e.g.*, TESLA-1005, ¶¶[0179]-[0181], [0048], [0081]-[0082], [0092]-[0098], FIGS. 12a-c.

142. As explained above in relation to claim elements [1a] and [1b], the Prediction Horizon is made up of multiple Frames, each of which includes multiple Integration Steps. *See* §VIII(A)(4)[1a]-[1b] above. Neiss's predictive cruise control (PCC) system iteratively calculates an optimal velocity for each of the Integration Steps in a Frame until calculations for all the Integration Steps in a Frame have been performed. *See, e.g.*, TESLA-1005, FIGS. 12a-c, FIGS. 13a-d, [0009]-[0014], [0177]-[0191]. A POSITA would thus have understood that the *optimized speed*

was calculated *over said segment destination* because an “optimal velocity” is iteratively calculated by Neiss’s PCC system for each Integration Step in a Frame, which has an endpoint.

143. Also, as further discussed above in relation to claim element [1b], the PCC block is implemented as part of Neiss’s “onboard computer” and is responsible for “iteratively calculate[ing] and stor[ing] in a memory vehicle control parameters” (which include optimal vehicle velocity) that optimize the vehicle operating cost function for” frames of “the route ahead of the vehicle.” TESLA-1005, ¶[0015]. Thus, the calculation of the *optimized speed* is performed at the *computer device* of Neiss’s vehicle.

144. Moreover, the “optimal velocity” for the vehicle is calculated for frames/segments of the route by minimizing an operating cost function, with one objective of the cost function being to “maximize fuel savings” and, thus, minimizing fuel consumption (which, as explained above for [1pre] corresponds to *managing the power consumption*). TESLA-1005, ¶¶[0031] (“The basic idea of Predictive Cruise Control (PCC) is to replace or supplement the set speed of a conventional cruise control with a speed band. Forward-looking terrain information is used to determine, as a function of position, a desired speed inside the speed band ... in order to maximize fuel savings”), [0045]-[0048] (the “Optimization Algorithm

Module 27” is the “main module” of the PCC block” that calculates the “optimal velocity trajectory, the optimal throttle pedal positions, and the controller gains” for “the approaching road within the Prediction Horizon, based on optimization of a cost function”), [0081]-[0082] (describing that one objective of the cost function is that the “vehicle should consume less fuel and should activate brakes as little as possible”), [0014] (the PCC system “utilizes information about the current vehicle position, as well as upcoming terrain in order to save fuel,” where the “vehicle velocity is ... controlled in order to follow an analytically computed optimal driving strategy, based on this information”).

145. As explained in §VIII.A.1 (incorporated here), Neiss describes an optimization problem with a cost function ( $J$ ) that depends on multiple subfunctions including fuel consumption ( $J_{fuel}$ ). *See also* TESLA-1005, ¶¶[0136]-[0159], [0088], [0092]-[0104], [0126], claims 8-9, 12. A POSITA would have also recognized that Neiss discloses in the equation below factors that play a role in the computation of the total cost:

$$J = \Phi + J_{time} + J_{fuel} + J_{velocity} + J_{lateral} - accel + J_{penalty}$$

*Id.*, ¶¶[0088], [0126]. For example, this formulation uses an overall “vehicle operating cost function” (represented as  $J$ ) that is defined mathematically (as shown

in the equation above) to reflect one or more objectives, including “consum[ing] less fuel” and “activat[ing] the brakes as little as possible” (represented as  $J_{fuel}$ ), “maintain[ing] a specified velocity (Set Speed)” (represented as  $J_{velocity}$ ), maintaining the “vehicle’s velocity” between “an upper limit” and “a lower limit” (represented as  $J_{penalty}$ ), minimizing “travel time” (represented as  $J_{time}$ ), and/or reducing “lateral acceleration while driving in curves” so as not “to impair driving comfort and safety” (represented as  $J_{lateral\_accel}$ ). See TESLA-1005, ¶¶[0082]-[0126]. Combining all these individual costs to a total cost that depends on all these variables is an extensively researched process which has as an objective total cost that captures all the variables that may play a role for the underlying optimization task.  $J$  is a total cost which is expressed as a function of variables that the user can compute through an optimization technique.

146. Additionally, as mentioned above and in one example, Neiss describes this “vehicle operating cost function” ( $J$ ) including an objective defined to minimize fuel consumption (represented as  $J_{fuel}$ ), which Neiss mathematically expresses as follows:

$$J_{fuel} = \frac{1}{2} \cdot h \cdot \sum_{k=0}^{N-1} S \psi v_k t h_k^2, \quad (\text{Eq. 18})$$

*Id.*, ¶¶[0081]-[0082], [0097]. This equation is represented as “a function of [at least] vehicle speed, road gradient[,] and throttle position” at points along the route ahead of the vehicle. *Id.*, ¶¶[0097]-[0098], Claims 10-11; *see id.*, ¶[0162]. Neiss thus describes an optimization problem with a cost function that depends on multiple subfunctions including fuel consumption ( $J_{fuel}$ ). *Id.*, ¶¶[0097]-[0098] (in this equation, “ $N$  is the number of base points in the considered prediction horizon and  $h=S_{k+1} - S_k$  is the Integration Step Size,” and  $v_k$  is the vehicle velocity for the  $k$ th integration step”), ¶¶[0126]-[0128], [0180] (Neiss’s optimization algorithm applies “several optimization iterations ... until the algorithm has converged and thus the values are optimal”), [0149]-[0160] (describing the algorithmic steps executed in each optimization iteration).

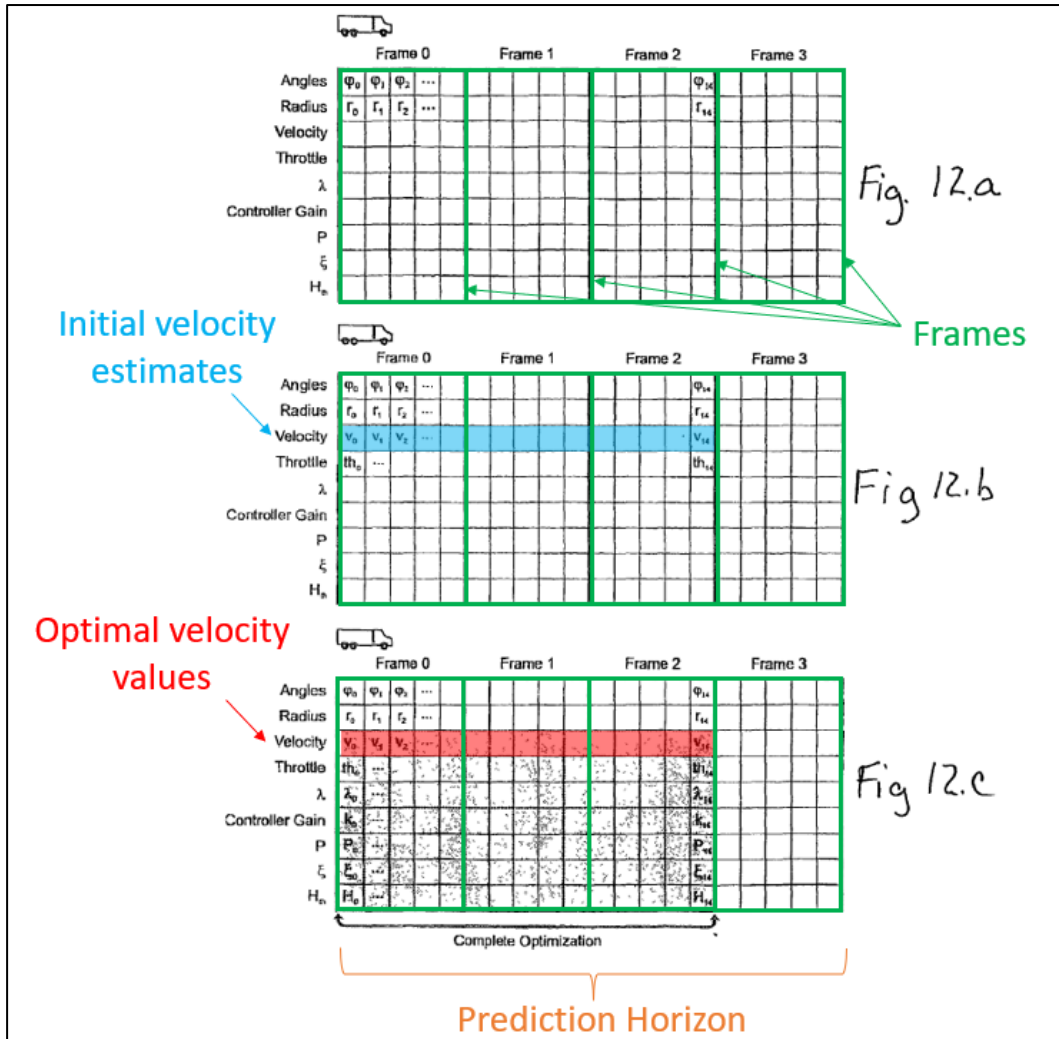
147. Neiss provides the following formula for the cost function with the objective of minimizing fuel consumption (represented as  $J_{fuel}$ ):

$$J_{fuel} = \frac{1}{2} \cdot h \cdot \sum_{k=0}^{N-1} S \psi v_k t h_k^2, \quad (\text{Eq. 18})$$

TESLA-1005, ¶¶[0097]-[0098] (in this equation, “ $N$  is the number of base points in the considered prediction horizon and  $h=S_{k+1} - S_k$  is the Integration Step Size,” and  $v_k$  is the vehicle velocity for the  $k$ th integration step”); *see also id.*, ¶¶[0126]-[0128] (including cost function for fuel minimization as part of overall system cost function

that is to be minimized), [0180] (Neiss’s optimization algorithm applies “several optimization iterations ... until the algorithm has converged and thus the values are optimal”), [0149]-[0160] (describing the algorithmic steps executed in each optimization iteration).

148. Neiss’s optimization algorithm thus iteratively calculates and outputs for each frame, optimized “velocity values” (which are then stored in a look-up table, as depicted in FIGS. 12a-c below). TESLA-1005, ¶[0181], FIGS. 12a-c; *see id.*, ¶[0186] (“velocity, throttle[,] and controller gain values are adapted, according to step 4 of the algorithm”).



TESLA-1005, FIGS. 12a-c (annotated) ; see also id., FIGS. 13a-d

149. In this manner, Neiss's PCC block implemented by its onboard computer (i.e., the **computer device**) iteratively executes its optimization algorithm to calculate an optimal velocity (i.e., **calculating an optimized speed**) by minimizing fuel consumption (i.e., **managing the power consumption**) for an upcoming frame with an endpoint/destination (i.e., **over said segment destination**). Additionally,

Neiss’s calculation uses as inputs one or more pieces of information, as described with reference to limitation [1d2] below.

***[1d2] wherein information is (j) information about the external environment of a vehicle, (jj) information about the operational status of the vehicle, (jjj) one or more user command inputs, or (jv) one or more operational parameters of the vehicle are received as input to said calculating and analyzed by one or more processors,***

150. Neiss-Ran teaches limitation [1d2] for the reasons described below.

151. As a preliminary matter, this limitation identifies four types of information that limit the ***calculating*** in limitation [1d1] and provides a disjunctive listing of these four items (“wherein information is (j) ..., (jj) ..., (jjj) ..., ***or*** (jv)...”). TESLA-1001, Claim 1. Given the disjunctive form used in reciting this limitation, I understand that this limitation is satisfied by the showing of a single type of information recited herein. Nevertheless, even if the claim is interpreted to require a showing of more than one type of claimed information (or even all four types of claim information), a POSITA would have found that to be obvious in view of Neiss’s disclosures as leveraged in the Neiss-Ran combination.

152. In particular, and as described below, Neiss-Ran discloses or renders obvious the calculation of optimal “velocity values” (i.e., ***optimized speed***) that uses as ***input*** the four types of information recited in this limitation. See, e.g., TESLA-1005, ¶¶[0049], [0190], [0031], [0092]-[0098]; see also analysis for [1d1].

153. ***First***, Neiss-Ran discloses using ***information about the external environment of the vehicle*** as an ***input*** to calculating the optimal “velocity values” (i.e., ***calculating the optimized speed***). The ’426 Patent lists “the current location of the vehicle,” “the current slope/grade of the route,” and “the predicted slope/grade of the next segments (or upcoming segments) of the route” as examples of “information about the external environment of the vehicle.” TESLA-1001, 3:30-35. Similarly, and as explained above for limitation [1d1], Neiss’s optimization algorithm uses ***information about the external environment*** of the vehicle in the form of the current location of the vehicle and the road segment ahead of the vehicle, including “gradient angles and curve radii of the upcoming road,” to calculate the optimal vehicle control parameters, including the optimal velocity/speed (i.e., the calculated ***optimized speed***). TESLA-1005, ¶¶[0049], Abstract, [0035], [0046]-[0049], [0167]-[0169]; *see also id.*, Claims 1-2 (“iteratively calculating ... vehicle control parameters,” “including at least vehicle speed,” “that minimize said vehicle operating cost function for a predetermined distance along said route ahead of said vehicle, as a function of at least ... vehicle position and route parameters,” where “said route parameters include at least road gradient and radius of curvature”).

154. ***Second***, Neiss-Ran also discloses using ***information about the operational status of the vehicle*** as input to calculating the optimal “velocity values”

(i.e., *calculating* the *optimized speed*). The '426 Patent explains that “the vehicle’s current speed” is an example of “information about the operational status of the vehicle.” TESLA-1001, 3:56-58. Similarly, Neiss’s optimization algorithm uses information about the current vehicle velocity for its optimization. TESLA-1005, ¶[0190] (“The current vehicle velocity is used as start velocity in order to take the actual vehicle state into account.”), [0036], [0162]-[0164].

155. ***Third***, Neiss-Ran discloses or renders obvious using *information about one or more user command inputs* as input to calculating the optimal “velocity values” (*calculating* the *optimized speed*). The '426 Patent states that the “preferred speed” and “maximum and minimum range over which speed should adjust” are examples of “user command inputs.” TESLA-1001, 4:7-11. Neiss similarly discloses using a preferred speed (which Neiss refers to as a “set speed”) and a maximum and minimum speed range (which Neiss refers to as a “speed band” formed between a lower and upper speed limit) as inputs for its optimal speed calculations. TESLA-1005, ¶[0031]; *see id.*, ¶¶[0084], [0102]-[0104], Claims 8-9 (specifying upper and lower velocity limits around the set speed). Neiss’s optimization algorithm then uses “[f]orward-looking information ... to determine ... a desired speed inside the speed band ... to maximize fuel savings.” *Id.*

156. A POSITA would have found obvious that the “set speed” and/or “speed band” are user-specified inputs (i.e., *one or more user command inputs*). A POSITA would have been motivated to provide a user the ability to input these values to enable the calculation of optimal vehicle control parameters that are finetuned/customized to reflect the user’s preferences and/or desired driving speeds. It was well known for user to input set speeds and speed bands as corroborated, e.g., by Petrie (TESLA-1016, Abstract (teaching that an “operator can define a relatively wide allowable speed range” or “a relatively narrow allowable speed range” in the context of a “cruise control system”)) and Michi (TESLA-1017, ¶[0007] (teaching that “the driver [can] increase or decrease the desired speed as commands”)).

157. A POSITA would have also been motivated to pursue this implementation because doing so is merely the application of known techniques (e.g., user specifying set speeds and speed bands to control operation of a cruise control) to a known system (e.g., Neiss’s system that uses set speeds and speed bands to calculate optimal vehicle control parameters) to yield predictable results (e.g., user-specified command inputs of specifying set speeds and speed bands, which Neiss already discloses as being used as part of its calculation of optimal vehicle control parameters).

158. Moreover, a POSITA would have expected a reasonable expectation of success in so implementing the Neiss-Ran system. That's because (1) cruise controllers routinely and conventionally included input mechanisms (e.g., hardware and control logic) by which a user can specify set speeds and speed bands at which to control the vehicle and (2) including such conventional input mechanism(s) as part of Neiss's cruise control system would enable user specification of set speeds and speed bands that Neiss already teaches as being used for the vehicle control parameter optimization. This is corroborated, e.g., by Petrie (TESLA-1016, ¶[0016] (teaching that a "driver interface adapted to receive driver input," including at least one of a driver determined such upper set speed, and a driver determined such lower set speed," in the context of a "cruise control system")) and Michi (TESLA-1017, Claim 1 (teaching an "input device to receive input of a desired speed by a driver")).

159. ***Fourth***, Neiss's optimization algorithm uses ***one or more operational parameters of the vehicle*** as an ***input***, in calculating the optimal velocity (i.e., the ***optimized speed***). Consistent with the '426 Patent's specification, which provides "maximum fuel volume" as one example of the "one or more operational parameters of the vehicle," Neiss discloses using maximum fuel volume as an operational parameter/input for its optimization algorithm in calculating the optimal "velocity values." TESLA-1001, 25:33-34. For example, Neiss discloses the use of *fuel<sub>max</sub>*

as an input for the optimization algorithm in multiple ways. *See, e.g.*, TESLA-1005, ¶¶[0070] (explaining that “ $fuel_{max}$  is the maximum value of ‘fuel’ in the look-up table”), [0075], [0093]-[0098] (expressing equations and functions (reproduced below) in relation to  $fuel_{max}$ ), [0069] (expressing “relation between the throttle pedal position  $th$  and the fuel value” with respect to  $fuel_{max}$ ).

$$J_{fuel} = \frac{1}{2} \cdot h \cdot \sum_{k=0}^{N-1} S\psi v_k th_k^2. \quad (\text{Eq. 18})$$

$$Q(t) = \frac{fuel_{max} i_D i_T}{\underbrace{2 \cdot 10^6 \pi r_W \rho_{fuel}}_{= \cdot \sqrt{\psi}}} \cdot v(t) \cdot th(t) \quad (\text{Eq. 15})$$

$$J_{fuel} = \frac{1}{2} \int_0^t S Q^2 dt. \quad (\text{Eq. 16})$$

160. Given these disclosures in Neiss, a POSITA would have understood that the calculation of optimal “velocity values” in Neiss uses as input maximum fuel volume, which is represented as  $fuel_{max}$  (i.e., *one or more operational parameters of the vehicle*).

161. Further, as described above for limitations [1a] and [1d1], Neiss includes an “onboard computer” (within which the PCC block is implemented) for performing the iterative calculations of the “vehicle control parameters that optimize the vehicle operating cost function for a predetermined Prediction Horizon along the route ahead of the vehicle.” TESLA-1005, Abstract, ¶¶[0001], [0015], [0037]-[0040], [0045], [0048]. Neiss further states that its system includes “a data processor programmed to calculate said optimum vehicle control parameters, based on said cost function, said vehicle operating parameters, said route parameters[,] and an optimization algorithm.” TESLA-1005, Claim 15. Given these disclosures of the components performing Neiss’s disclosed optimization algorithm, a POSITA would have understood that Neiss’s onboard computer (*said computer device*) includes at least a data processor (*one or more processors*) that receive one or more of the above-described information as part of executing the optimization algorithm to calculate optimal vehicle control parameters, including the optimal “velocity values” (i.e., *received as input to said calculating<sup>3</sup> ... by one or more processors*).

---

<sup>3</sup> The present limitation’s recitation of “said calculating” is an antecedent reference to the immediately preceding “calculating” limitation ([1d1]).

Moreover, a POSITA would have recognized the well-known use of processors in computers to execute algorithms. *See also* §VIII(A)(4)[1d1] *supra*.

162. Additionally, because the above-described information is used by Neiss’s optimization algorithm (as executed by the data processor) to perform the above-described optimal parameter calculations, a POSITA would have understood that this information would also be ***analyzed by*** the data processor (i.e., the ***one or more processors***) as part of such optimization calculations. Indeed, a POSITA would have understood that an algorithm would have to process and analyze received data before it can perform calculations with respect to such information.

***[1e] (ii) calculating a probable speed;***

163. Neiss-Ran combination limitation [1e] for the reasons described below.

164. The ’426 Patent’s specification does not use the phrase “probable speed” and the claim does not define this phrase nor recite any requirements for it. *See* TESLA-1001. The plain meaning of “probable speed” is a speed that is probable or likely for the vehicle would to be traveling at. *See, e.g.*, Heinle’s Dictionary (TESLA-1025 (defining “probable” as “likely to happen, having a good chance of occurring”)), Webster’s Dictionary (TESLA-1026 (defining “probable” as “likely to occur or prove true”)).

165. Neiss-Ran renders obvious calculating a probable speed. As described above (*see* §VIII(A)(3) *supra*; *see also* TESLA-1005, ¶¶[0188]-[0191], [0010], [0163], [0167]-[0169]), the combination would have leveraged Ran’s teachings of using traffic prediction models, as implemented at the computer device, to calculate predicted speeds for upcoming road segments based at least on real-time traffic conditions on those road segment(s). *See, e.g.*, TESLA-1006, 11:23-42 (describing use of historical and real-time data that is fused together and used by the prediction model to determine the predicted speed for an upcoming segment), 18:18-31 (same), 19:46-49 (“the real-time data 415 are used to modify the historical traffic data in order to provide a better estimate and forecast for the short-term traffic conditions”). Ran explains that the traffic prediction models “generate estimates of route segment speeds ... for the current time instant” and “produce prediction of route segment speeds ... for the future time instants.” TESLA-1006, 17:48-63, 19:19-20:14; *see id.*, 18:53-56 (providing four examples of statistical models, including simulation models, dynamic traffic assignment models, statistical models, and heuristic models), FIG. 4A, 7:14-27, 17:48-60, 19:19-20:17 (explaining that predicted travel speed can be computed using models, e.g., a weighted average, that combine historical speeds and a real-time speeds).

166. One example of such a traffic prediction model in Ran is a Greendshields model, which provides a “fundamental speed-density relationship” where speed is a function of traffic density. *See id.*, 11:23-42. The Greendshields formula is expressed as follows:

$$V=V_o(1-D/D_j) \quad (1)$$

where V: speed (vph)

$V_o$ : freeflow speed based on the route segment classification (vph)

D: density or number of vehicles per mile (vehicles/mile)

$D_j$ : jam density or maximum number of vehicles per mile (vehicles/mile)

167. This formula states that the traffic speed on a route segment can be determined in part by the number of vehicles per mile or the vehicle density, and the route segment capacity (*i.e.*, the “maximum number of vehicles per mile”). *Id.*

168. Thus, as explained above and for the reasons discussed with reference to §VIII(A)(3) *supra*, the combination would have leveraged Ran’s teachings of calculating a predicted speed for the route segment (*i.e.*, ***calculating a probable speed***) using at least real-time traffic conditions. Because the predicted speed for the segment is determined based on such real-time traffic conditions at the upcoming route segments, a POSITA would have understood that this speed is a likely or

*probable speed* that the vehicle is expected to travel at the respective road segment (i.e., the frame in the context of Neiss-Ran).

169. Moreover, as described above in §VIII(A)(3) *supra*, Neiss's onboard computer that executes the PCC block and the associated optimization algorithm (i.e., *said computer device*) would calculate the predicted speed (i.e., *calculating a probable speed*). Neiss already discloses performing optimization calculations using its onboard computer, and, thus, it would have been a straightforward design choice to similarly implement the probable speed calculation at the same computing device (using Ran's disclosed traffic prediction models and the associated data used by those models). Indeed, because the predicted speed is used to adapt the optimized speed values for the upcoming route segments, it would be more resource efficient to perform the Ran-based predicted speed computations at the vehicle instead of a separate and remote server device. *See* §VIII(A)(3) *supra*.

170. In this manner, a POSITA would have found it obvious that the onboard computer of the Neiss-Ran combination (i.e., *said computer device*) calculates a predicted speed based on real-time traffic conditions (i.e., *calculating a probable speed*) for each frame along a route (where each frame corresponds to a segment that has an end point or a *segment destination*).

***[1f] (iii) determining an efficient speed based on said optimized speed and said probable speed; and***

171. Neiss-Ran teaches ***determining an efficient speed based on said optimized speed and said probable speed*** (the calculations of which are described above with reference to limitations [1d1], [1d2] and [1e]) for the reasons described below.

172. Specifically, and as described further below, Neiss-Ran teaches this limitation because it discloses calculating a vehicle's optimal speed for an upcoming segment (i.e., an ***optimized speed***), and improving this calculated speed (***optimized speed***) iteratively based on the real-time predicted speed (i.e., ***probable speed***) to thereby obtain an ***efficient speed***. See, e.g., TESLA-1005, ¶¶[0188]-[0191], [0010], [0163], [0167]-[0169]; TESLA-1006, 11:23-42.

173. As described above, Neiss teaches the calculation of the ***optimized speed*** for an upcoming road segment/frame. See §VIII(A)(4)[1d1] *supra*. Also, as described above, Neiss in view of Ran teaches the calculation of a predicted speed for a route segment/frame based at least on real-time traffic conditions (***probable speed***) for the particular route segment/frame. See §VIII(A)(4)[1e] *supra*.

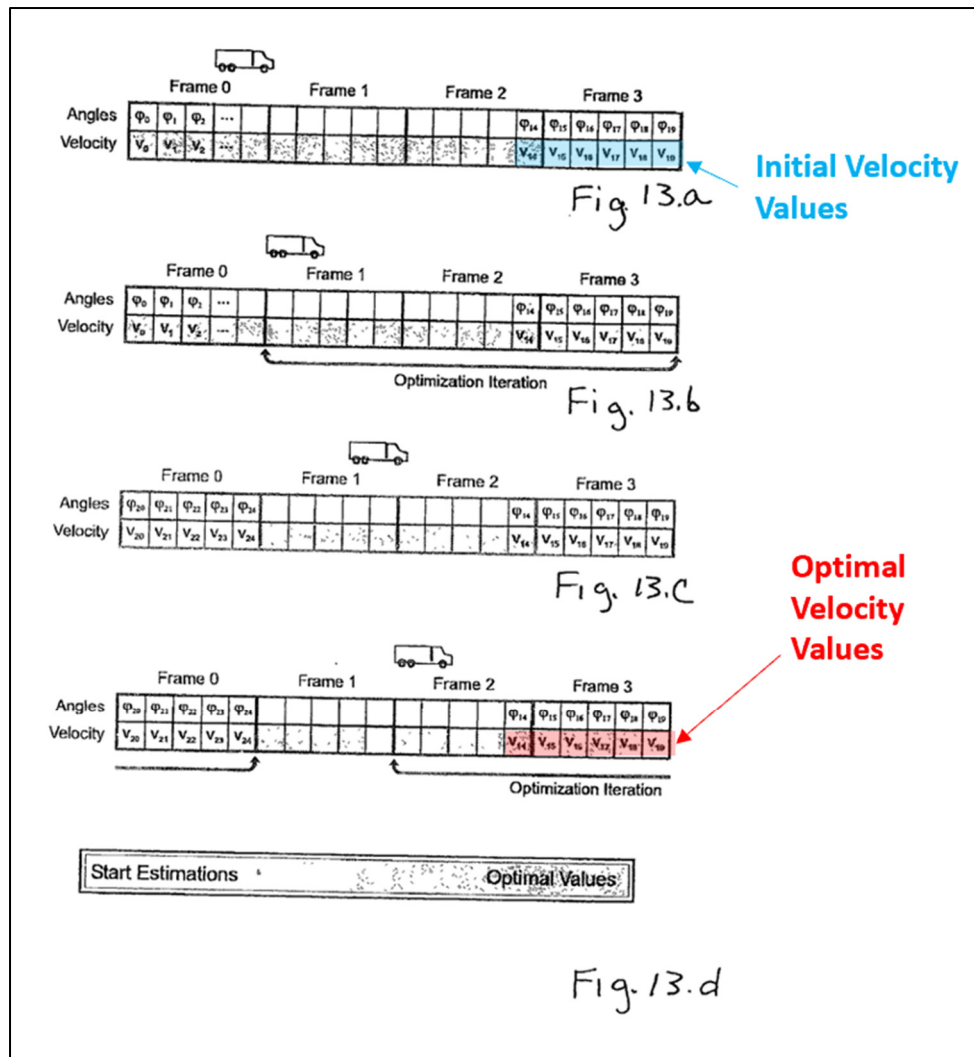
174. Neiss further states that its “online, real time calculation adapts the look-up table during the drive by taking the current vehicle velocity into account.” TESLA-1005, ¶[0163]; see *id.*, FIGS. 13a-d *infra*. This can occur when “vehicle

velocity drifts significantly from the velocity stored in the look-up table for the current position (as might be caused, for example, by a slower vehicle driving ahead),” in which case, “the velocity values for the following positions in the look-up table then can no longer be considered to be optimal.” *Id.*, ¶[0167]. In such situations, “an online version of the algorithm can adapt these values by taking the current velocity into account.” *Id.* In one example, Neiss explains that “taking the current velocity into account” includes, e.g., using the vehicle’s current velocity as a start velocity for upcoming frames in the Prediction Horizon. *Id.*, ¶[0190].

175. However, as explained in §VIII(A)(3) *supra*, a POSITA would have recognized that the “current velocity” that is used for such adaptation is the vehicle velocity for the frame in which the vehicle is currently driving. *See* TESLA-1005, ¶[0190]. A POSITA would have also recognized that there may be situations where there is a difference between the vehicle’s speed as it enters a first frame (e.g., Frame 1) and the vehicle’s velocity in a later frame (where, e.g., the traffic conditions might be different and the relative velocities of vehicles may therefore be different from the vehicle’s current velocity). *See* §VIII(A)(3) *supra*. To further increase the accuracy and efficiency of Neiss’s algorithm, and as explained in §VIII(A)(3) above, a POSITA would have found it obvious to modify Neiss’s algorithm so that—instead of using a current velocity for an existing frame where the vehicle is driving—the

optimization algorithm uses the real-time traffic-based predicted speed for the route segment for which the optimization is being performed, where the predicted speed is based at least on the prevailing and real-time traffic conditions in that segment (as taught by Ran).

176. Thus, Neiss-Ran performs a first optimization iteration that yields optimized vehicle control parameters (including optimal velocities) for one or more upcoming frames. *See* TESLA-1005, ¶¶[0182]-[0187]. Subsequently, as the vehicle drives along the route, additional optimization iterations are performed, and each optimization iteration results in the values in the look-up table (for the upcoming frames/segments) being updated. TESLA-1005, ¶¶[0188]-[0191].



TESLA-1005, FIGS. 13a-d (annotated)

177. For example, as depicted in Figures 13a-d above, the initial velocity estimates are optimized into “optimal velocity values” (highlighted in red) in an initial set of optimization iterations. *Id.*, FIGS. 13a-d. In the combination, each additional optimization iteration (given Neiss contemplates multiple optimization iterations being performed on the same frame) would use the predicted vehicle

velocity for the upcoming segment/frame—instead of using the “current vehicle velocity” for the segment/frame currently being driven (as Neiss teaches)—to adapt and further refine the previously computed optimal “velocity values.” TESLA-1005, ¶¶[0012], [0163], [0167], [0190]. In this manner, Neiss-Ran’s system thus accounts for real-time conditions and uses a more accurate, traffic-informed predicted speed to optimize and adapt the computed optimal velocities in the lookup table and thereby obtains the updated optimal values (including optimal velocity values) for the frame/segment. *See id.*, ¶[0190]. In this manner, and consistent with Neiss’s stated goals, the optimal “velocity values” continue to be adapted in real time as the vehicle continues to travel along the route.

178. Because Neiss’s optimal “velocity values” are iteratively adapted/refined with each iteration using the predicted speed computed using real-time traffic information (*probable speed*), Neiss-Ran teaches determining an updated or re-optimized velocity value (*determining an efficient speed*) for an upcoming segment/frame *based on* the initially optimized velocity values (*optimized speed*) for the segment and the predicted speed based on real-time traffic conditions for that segment (*probable speed*). *See* TESLA-1005, ¶¶[0167] (“... the velocity values for the following positions in the look-up table then can no longer be considered to be optimal. In this case an online version of the algorithm can adapt

these values by taking the current velocity into account...”), [0169] (it is “important” that “the optimization algorithm” “permits online adaptation of the look-up table”), [0010] (the “vehicle control system” is able “to adapt the velocity control to changing driving situations which result from the intervention of external constraints on driving behavior as the vehicle transmits a particular route”), Abstract (the “optimal vehicle control parameters for the Prediction Horizon are ... continuously updated and replaced by new data as the vehicle moves along”), [0015] (“as the optimization iterations are performed, the “optimal vehicle control parameters are continuously updated [from the previously-calculated optimal vehicle control parameters stored in the memory] and replaced by new data as the vehicle ... moves along”), [0190], [0156]-[0159], [0168], [0015], [0010], Abstract, FIGS. 13a-d *supra*. Thus, Neiss-Ran’s updated optimal value for speed corresponds to an ***efficient speed*** because it achieves the objective that “[t]he vehicle should consume less fuel” while accounting for real-time operating conditions. *See* TESLA-1005, [0082], [0031] (“determine...a desired speed...to maximize fuel savings”); *see also* TESLA-1001, 19:18-20 (’426 specification describing its “method[.]...may be used to calculate a speed ... for running the vehicle at an optimal fuel efficiency”). A POSITA would have understood that the data stored in the memory would have been updated with the claimed ***efficient speed*** as optimization iterations are performed.

179. Moreover, as explained above for limitation [1a] and [1d1], Neiss's onboard computer that implements the PCC block (*said computer device*) performs the optimization algorithm. Because the PCC block is also responsible for the calculations performed in each subsequent iteration of the algorithm, a POSITA would have recognized that the same computer device would also be used to compute the refined/adapted optimal velocity values (*efficient speed*) for upcoming route segments/frames that account for the real-time traffic conditions in those frames.

180. Additionally, or alternatively, if it is determined that Neiss does not explicitly teach how to “adapt[] the look-up table during the drive” by taking the real-time traffic-based predicted speed “into account,” a POSITA would have found it obvious to perform adaptation of the already-computed optimized speed for a route segment/frame and the expected/probable speed for that segment/frame, using conventional and well-known techniques for combining two speed values. TESLA-1005, ¶[0163]. As an example, Ran states that a single speed for a vehicle can be calculated as a weighted average of two speeds. TESLA-1006, 7:14-27 (“Predictive\_Speed = Historical\_Speed\*b + Current\_Speed\*(1-b) + Random Error,” where “Predictive\_Speed is the predictive value in future time instant t; Historical\_Speed is the historical average at the same time instant t for the same type

of day; Current\_Speed is the real-time speed measured using field devices; b is a weighting parameter; Random Error is an error term for the speed prediction.”). Ran also describes additional techniques for combining speed values, e.g., “various types of models, including weighted average and fuzzy logic.” TESLA-1006, 18:29-31, FIG. 3; *see also id.*, 18:18-49.

181. Thus, given Neiss’s recognition of adapting optimized velocities using real-time conditions that account for “current velocity” and the well-known and conventional approaches to combining speed values (as Ran teaches), a POSITA would have been motivated to combine the *optimized speed* value (as described in limitation [1d1]-[1d2]) and *probable speed* value (as described in limitation [1e]) to obtain a combined/adapted speed, which corresponds to the *efficient speed*. *See* TESLA-1005, ¶¶[0008] (recognizing that “unforeseeable driving situations” can occur “during driving with automatic throttle control,” such as “when the vehicle approaches a slower vehicle ahead”), [0167]; *see* TESLA-1001, 19:3-12 (’426 specification describing that the efficient speed is a weighted combination of the optimized speed and the probable optimized speed).

182. A POSITA would have also been motivated to pursue this implementation because doing so is merely the application of known techniques (e.g., combining speed values using known techniques, e.g., modeling, weighted

averaging, application of fuzzy logic) to a known system (e.g., Neiss's system that uses a velocity value to adapt previously-computed optimal velocities in the look-up table) to yield predictable results (e.g., computation of an updated and adapted speed value, as Neiss teaches and Ran contemplates).

183. Further, a POSITA would have had a reasonable expectation of success in combining the teachings in this manner because Neiss already contemplates adapting optimized speeds with another speed quantity, and Ran provides a conventional and well-known modeling/weighted average/fuzzy logic technique(s) to combine these speed quantities to achieve a more realistic optimal speed value that accounts for real-time traffic conditions (the *efficient speed*), which is consistent with Neiss's goals and Ran's teachings.

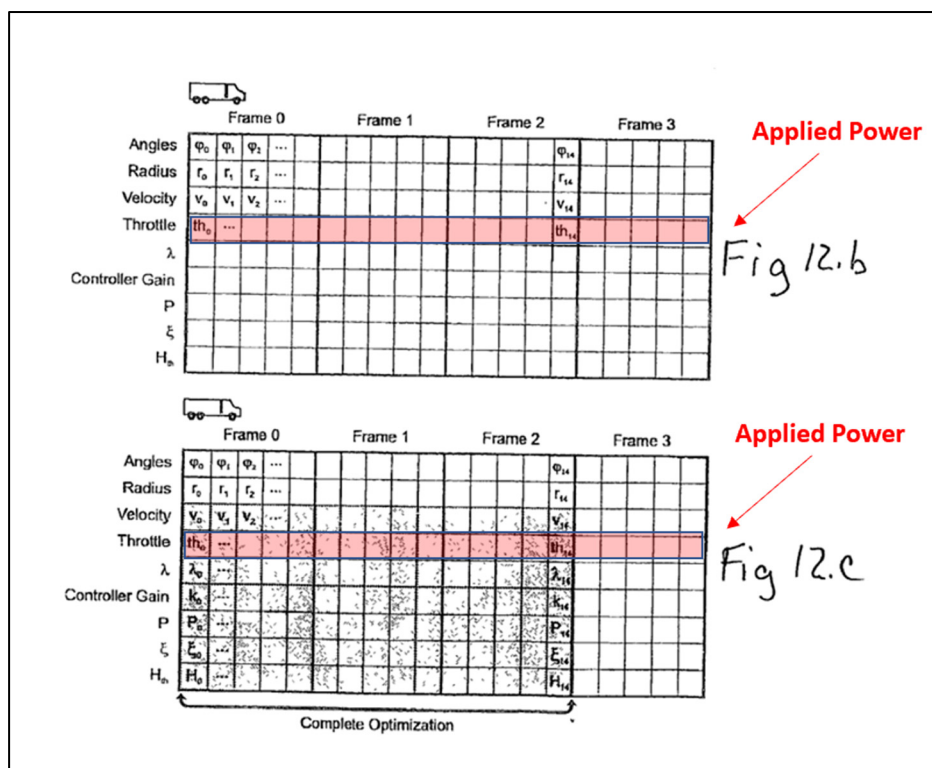
***[1g] (iv) applying the calculated power corresponding to said determined efficient speed to the vehicle engine to travel said segment destination.***

184. Neiss-Ran teaches limitation [1g] for the reasons described below.

185. ***First***, Neiss-Ran teaches *calculating power* (e.g., throttle pedal positions that maps to the amount of power/fuel to inject) *corresponding to the determined efficient speed* (the updated/re-optimized speed values in Neiss-Ran's lookup table). As described above for limitations [1d1]-[1f], Neiss's optimization algorithm iteratively calculates vehicle control parameters, which includes the

optimal velocity values (as adjusted/adapted based on the expected/predicted speed informed by real-time traffic conditions), the throttle pedal positions, and the controller gain series, for the upcoming frames along a vehicle's traveled route. TESLA-1005, ¶¶[0182]-[0191]; *see* §VIII(A)(4)[1d1]-[1f] *supra*.

186. Indeed, the optimal velocity values, the throttle pedal positions, and the controller gain series are stored in the look-up table in Neiss (which is “contained in a ring buffer memory”). TESLA-1005, ¶[0162]. As the vehicle traverses the route, Neiss's system determines the estimated vehicle position, and the optimal velocity values and throttle pedal positions stored in the look-up table are “read[] out.” *Id.*; *see also id.*, ¶[0004] (“the current vehicle position may be determined, for example, by means of a Global Positioning System (GPS)”). A POSITA would therefore have understood that for each *efficient speed* calculated by the Neiss-Ran system, corresponding throttle pedal position and controller gain series values are also calculated. *See also* TESLA-1005, FIGS. 12b-c *infra*.



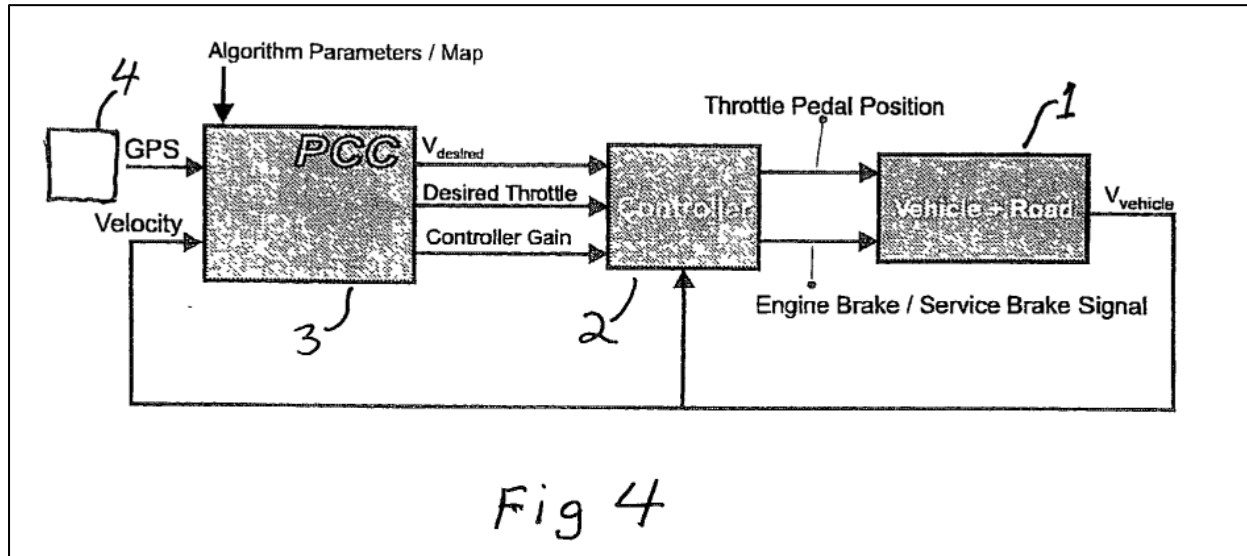
TESLA-1005, FIGS. 12b-c (annotated)

187. A POSITA would have understood that Neiss's disclosure of applying a pedal or throttle position corresponding to the determined optimal velocity (i.e., *efficient speed*) teaches the recited *apply[ing] a calculated power corresponding to said determined efficient speed*. That is because, as Neiss explains, the pedal or throttle position corresponds or maps to the amount of fuel/power applied to the vehicle, and when the vehicle is controlled according to the calculated throttle (that corresponds to an optimal velocity), the system injects an amount of fuel (corresponding to this throttle position value) to the engine. TESLA-1005, ¶¶[0034]

(“A block diagram of the PCC vehicle control system according to the invention is shown in FIG. 4. It includes a conventional controller 2, which may (but need not) be a closed loop cruise control system in which the throttle of the vehicle 1 is controlled based on an error signal between actual and desired vehicle speed.”), [0039] (the throttle value computed using Neiss’s optimization algorithm represents “the Throttle Pedal Position” that “controls the fuel injection to the motor”), [0042] (“The amount of fuel injected can be mapped to a pedal or throttle position, which is why it is referred to in this way”); see §VIII(A)(4)[1pre], [1d1], where I explain that power corresponds to fuel in the context of fuel-powered engines.

188. **Second**, Neiss-Ran teaches *applying the calculated power ... to the vehicle engine to travel said segment destination*. Indeed, Neiss recognizes that “there are many ways in which the desired vehicle velocity can be realized by the controller (TESLA-1005, ¶[0042]), which generally require a command specifying a “throttle position” that corresponds to a speed to be achieved. *Id.* This results in “the desired vehicle speed [being] converted to a desired engine speed, which is sent to the engine and the engine achieves it via its own controller or using a separate adaptive cruise controller.” *Id.* As one example, Neiss explains that, for a diesel engine, the “amount of fuel injected” that powers the vehicle “can be mapped” to “a pedal or throttle position” (such as the computed throttle values associated with

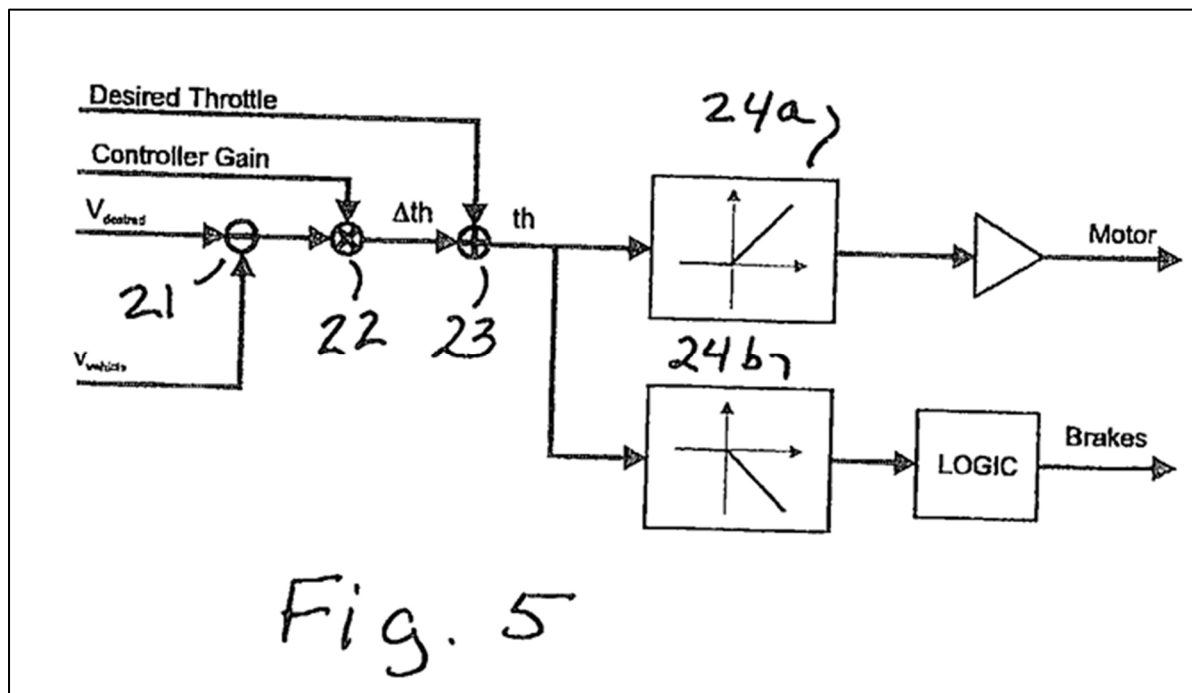
*efficient speed*). *Id.*; *see, e.g., id.*, ¶[0039], FIG. 4 (*infra*) (depicting the applicability of the throttle pedal position to the vehicle); FIG. 5 (*infra*) (depicting a “linear controller of a type which may be included in the system in FIG. 4,” where the “desired speed” and “desired throttle input” as computed using Neiss’s optimization algorithm are both used as inputs to “control the vehicle throttle and brakes”); *see also* TESLA-1001, 19:15-17 (’426 Patent explaining that “efficient speeds are expressed as power or energy to be applied to the vehicle engine”).



TESLA-1005, FIG. 4

189. FIG. 5 (reproduced below) “shows a linear controller of a type which may be included in the system of FIG. 4. In the subtraction unit 21, the actual vehicle speed ... is compared with the desired speed  $v_{des}$ , and multiplied by the input controller gain in a multiplier unit 22, to generate a throttle error signal  $\Delta th$ . The

latter is then added to a desired throttle input in an adder 23 to generate a throttle signal  $th$ , that is applied to vehicle throttle and braking characteristics 24 a, b, which are used to control the vehicle throttle and brakes.” TESLA-1005, ¶[0043].



TESLA-1005, FIG. 5

190. Additionally, the *calculated power* is applied to cause the vehicle *to travel said segment destination*. As explained above, *efficient speed[s]* are calculated by the Neiss-Ran optimization algorithm for each Integration Step in each Frame. See §VIII(A)(4)[1f] *supra*. The application of throttle positions that correspond to each *efficient speed* thus causes the application of *calculated power* for each Integration Step in each Frame, which causes the vehicle to travel through the Frame until it reaches the endpoint of the Frame (*segment destination*). See

TESLA-1005, Claim 1 (“reading from said memory optimized vehicle control parameters corresponding to a current position of said vehicle; and controlling current vehicle operation based on said optimized vehicle control parameters”), [0162] (“The output values of the PCC system are then determined by reading out the values corresponding to the estimated vehicle position.”), [0188]-[0190], [0192] (similar).

**(b) Claim 2**

***[2] The method of claim 1, further comprising continuously calculating said efficient speed until a final destination is reached.***

191. Neiss-Ran teaches Claim 2 for the reasons described below. As described above for Claim 1, Neiss’s system iteratively calculates the claimed “*efficient speed*” for a “route section ahead of the vehicle” as the vehicle “travels over a particular route.” TESLA-1005, ¶¶[0001], [0015]; *see id.*, Abstract (“As the vehicle proceeds, an onboard computer iteratively calculates and stores in a memory vehicle control parameters that optimize the vehicle operating cost function for a predetermined Prediction Horizon along the route ahead of the vehicle. The optimal vehicle control parameters for the Prediction Horizon are then stored in a memory and continuously updated and replaced by new data as the vehicle moves along, thereby adjusting the “optimal” control parameters to reflect actual vehicle historical operating experience during the journey.”). Indeed, as the vehicle travels, “a whole

iteration is applied,” with iterations being continuously performed on various frames along the route ahead of the vehicle. *See id.*, ¶¶[0188]-[0192]. Because the optimization algorithm in Neiss-Ran operates iteratively and computes optimal vehicle control parameters as the vehicle continues driving along a route (i.e., for each additional portion of the route ahead of the vehicle as the vehicle travels the route), a POSITA would have understood that the optimization calculations would continue to be iteratively performed until there are no more “route sections ahead of the vehicle,” for example, when a final destination has been reached.

192. Alternatively, a POSITA would have found obvious that the calculation would continue *until a final destination is reached*, e.g., to allow (1) reaping benefits of continued optimizations, as disclosed in Neiss, for the entire trip route (TESLA-1005, ¶[0012]), (2) accounting for changes in driving or operating conditions as the vehicle travels the route (TESLA-1005, ¶¶[0165], [0167]), and (3) achieving a predictably operating system that predictably extends Neiss’s existing techniques to the entire route, thus merely applying known techniques (calculation of operating parameters until a final destination is reached) to a known system (Neiss’s system for ongoing parameter calculation) to yield predictable results (a system for ongoing calculation of operating parameters until a final destination is reached) (TESLA-1005, ¶¶[0165], [0167], [0012]). A POSITA would have

expected success because this involves common techniques and devices—namely, merely extending Neiss’s functionality for iterative optimization as the vehicle is being driven for the entire trip—that are conventional, predictable, and present no unexpected results given their consistency with Neiss’s teachings.

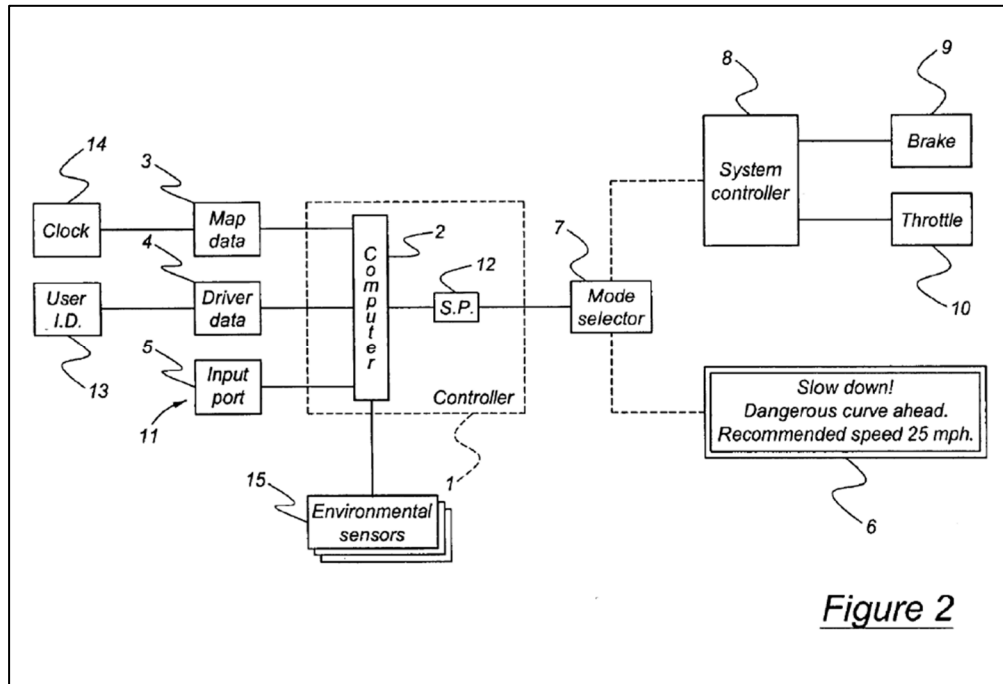
**B. [GROUND 2] – Claim 3 is Rendered Obvious by Neiss-Ran-Andersson**

**1. Overview of Andersson (TESLA-1007)**

193. Andersson describes methods for controlling a vehicle to address drawbacks in conventional cruise control systems that are unable to modify user-set speeds to account for changing road conditions. *See* TESLA-1007, 1:6-43; *see id.*, 1:44-3:18, 3:34-4:60, 5:62-6:14.

194. Andersson extracts vehicle position information to predict “a current and an upcoming road section,” and generates a “speed plan” for these road sections. *Id.*, 1:46-57. To generate the speed plans, Andersson’s system “extract[s] ... one or more road attributes” for each road section and “relat[es] the road attributes to [a] driver behavior model.” *Id.*, 2:4-15. The driver behavior model may incorporate “a neural network ... to be used for predicting a driver[’]s intended speed on a road section never driven before.” *Id.*, 4:53-60. This is done by “studying a chosen set of road section attributes, ... such as curvature, road surface[,] and so on, and

studying driver [] behavior at previously driven road sections showing similar attributes, as stored in the driver information database.” *Id.*



*TESLA-1007, FIG. 2*

195. As shown in Figure 2 above, Andersson’s system includes a “computing device 2,” which includes “a neural network unit” for deriving the driver behavior model. *Id.*, 4:47-49. “The driver behavior model is thereafter inputted to a speed plan generator 12,” which generates speed plans for the vehicle. *Id.*, 3:54-56. The computing device 2 also receives as inputs data from “the map database 3,” data from “the driver information database 4,” and a current position of the vehicle from the “input connection 5” (which receives a “positioning signal 11”). *Id.*, 3:36-

41. The speed plans can either be displayed to the driver via “a driver display unit” or used to “autonomously control” the vehicle. *Id.*, 2:32-39.

196. Andersson is analogous art to the '426 Patent: it is from the same field of endeavor (*e.g.*, vehicle speed prediction) and is reasonably pertinent to the problem the '426 Patent was trying to solve (*e.g.*, problem of how to determine speeds at which a vehicle should travel based, *e.g.*, on road or operating conditions).

## **2. Combination of Neiss, Ran, and Andersson**

197. As described above in §§VIII(A)(3)-(4), Neiss-Ran’s predictive cruise control system is configured to determine an *efficient speed* for upcoming route segments based on optimized velocity values that are further refined in subsequent iterations, where each iteration adapts the optimized speeds using the expected speeds in the upcoming frames/segments based on real-time traffic conditions in those frames/segments (*probable speed*). *See* §VIII(A) *supra*. Moreover, Ran teaches that this predicted/probable speed is calculated by fusing/combining historical driving data with real-time traffic data, *e.g.*, using “various types of models, including weighted average and fuzzy logic.” TESLA-1006, 18:18-49, FIG. 3, 7:17-27.

198. Andersson describes a method for predicting vehicle speeds along a route segment that a driver has not driven on before by looking to historical data

from similar route segments. *See* §VIII(B)(1) *supra*. Specifically, and as explained in §VIII(B)(1) *supra*, Andersson teaches generating a “speed plan” for a particular route segment based on historical data for similar route segments. *See* §VIII(B)(1) *supra*. The “speed plan” in Andersson “contain[s] information regarding preferred vehicle speed, accelerations[,] and retardations for an up-coming road section.” TESLA-1007, 3:58-60.

199. As described below, a POSITA would have been motivated to combine the teachings of Andersson and Neiss-Ran and would have had a reasonable expectation of success in doing so, to arrive at a combined system (referred to herein as Neiss-Ran-Andersson) that augments the *efficient speed* determination in Neiss-Ran to further account for historical speed data for a similar route segment, as disclosed by Andersson. Thus, in view of Ran and Andersson, the combination would be implemented by computing the predicted speed (*probable speed*) based on prediction models that, e.g., compute weighted averages of the following speed values—the historical driving speeds (which further include historical driving data for similar segments, as Andersson contemplates) and the speeds based on real-time traffic information (in view of Neiss-Ran, *see* §VIII(A)(4)[1e] *supra*). TESLA-1006, 11:23-42, 18:18-49, 7:17-27; TESLA-1007, 4:47-60. And, the efficient speed would then be determined in the same manner as described above in §VIII(A)(4),

e.g., based on optimized speed and the predicted/probable speed (the latter being augmented in view of Andersson’s teachings, as described above).

200. A POSITA would have been motivated to combine the teachings of Neiss, Ran, and Andersson for at least the following reasons.

201. **First**, a POSITA would have recognized the benefit of using historical data in predicting a vehicle’s speed at an upcoming route segment. Specifically, the use of historical data provides increased accuracy, as historical data reflects previous real-world data samples over a long time horizon. This is corroborated by, e.g., Entin (TESLA-1018, ¶[0006]), which teaches that typical monitoring systems use historical data to analyze and detect normal usage patterns, which are used to generate predictive functions for normal operation, and where the monitoring system detects abnormal behavior based on thresholds that track the expected normal operational pattern. This is further corroborated by Ran (TESLA-1006, 15:33-37), which recognizes the significant impact of historical data for long-term prediction purposes.

202. Both Neiss and Ran recognize that historical speed data for a particular route/route segment can be used to predict speed values for that route/route segment. *See, e.g.*, TESLA-1006, 7:17-27 (vehicle speed can be predicted as a weighted average of a “historical speed” and a “current speed” for each route segment);

TESLA-1005, Abstract (“The optimal vehicle control parameters for the Prediction Horizon are then stored in a memory and continuously updated and replaced by new data as the vehicle moves along, thereby adjusting the ‘optimal’ control parameters to reflect actual vehicle historical operating experience during the journey”). Indeed, it was also well known and commonplace to use historical driving data on particular road segments to more effectively predict speed and other control parameters on a current route. This is further corroborated by, e.g., Kuroda (TESLA-1011, 6:65-7:34 (teaching use of historical driving speeds to predict vehicle speeds) and Ichikawa (TESLA-1019, ¶[0006] (recognizing that traveling history can be used as a factor in predicting a traveling speed pattern for each route segment)).

203. ***Second***, a POSITA would have recognized that using historical data along similar route segments enhances systems that rely *solely* on historical data along a particular route segment to predict vehicle speeds, as discussed above. Specifically, while a POSITA would have recognized that historical driving data can be beneficial when predicting speeds for upcoming routes (*see, e.g.*, TESLA-1006, 15:33:37 (recognizing the significant impact of historical data for long-term prediction purposes)), historical data for a particular route is not always available (*see, e.g.*, TESLA-1005, ¶[0007] (teaching that known prior art systems can be “disadvantage[ous]” because of the requirement of “at least one ‘record drive’ to be

performed prior to operation of the system’’)). A POSITA would have therefore been motivated to seek out solutions, such as those described in Andersson, that allowed leveraging of historical data for a similar route segment even if historical data for a particular route segment is unavailable—and thereby enabling vehicle control systems to benefit from the use of historical data.

204. ***Third***, before the Critical Date, it was well known to use historical driving data for similar road segment to predict speed for a current road segment, as Andersson expressly teaches. *See, e.g.*, TESLA-1011, 6:65-7:34 (using driving history to predict a “vehicle speed pattern”) TESLA-1012, ¶[0028] (recording vehicle speed for use “at a later date on a trip by another driver over the same or ***substantially similar route***”). Similarly, and as explained in §VIII(B)(1) *supra*, Andersson teaches generating a “speed plan” for a particular route segment based on historical data for similar route segments. *See* §VIII(B)(1) *supra*. The “speed plan” in Andersson “contain[s] information regarding preferred vehicle speed, accelerations[,], and retardations for an up-coming road section.” TESLA-1007, 3:58-60.

205. A POSITA would have thus been motivated to combine Andersson’s teachings with those of Neiss-Ran because doing so is merely the application of known techniques (*e.g.*, determining a speed along a route segment based on

historical speeds on route segments having similar characteristics, as disclosed in Andersson) to a known system (e.g., Neiss-Ran’s predictive cruise control system that adapts an optimization algorithm to predict vehicle speeds using real-time and historical speeds for road segments) to yield predictable results (e.g., vehicle speed predictions that additionally account for augmented historical data, such as data for similar segments).

206. A POSITA would have had a reasonable expectation of success in pursuing this combination—e.g., augmenting the *efficient speed* determination to further account for historical speed data for a similar route segment—because Neiss and Ran already appreciated the value and/or conventionality of factoring in historical speeds when predicting vehicle speeds. *See, e.g.*, TESLA-1005, Abstract; TESLA-1006, 7:17-27. For example, the Neiss-Ran combination leverages Ran’s teachings of computing a predicted speed for a route segment using at least real-time traffic information. Ran further discloses that its prediction models also use historical driving data for route segments to compute the predicted speed. TESLA-1006, 17:7-16. Andersson also discloses prediction of driving speeds using historical data (including data for similar road segments).

207. Thus, given Andersson’s teaching of using historical driving data for similar travel segments in predicting speed and Ran’s recognition of predicting

speed using real-time traffic data as well as historical driving data for such segments, it would have been straightforward and well within a POSITA's skill level to compute the predicted speed based on both real-time traffic information as well as historical driving data (which further includes historical data for similar road segments, as Andersson teaches). TESLA-1006, 7:28-29. Indeed, Ran recognizes that historical data can be readily fused/combined with real-time traffic data, e.g., using "various types of models, including weighted average and fuzzy logic"). TESLA-1006, 18:29-31. And, as one example, Ran provides a "simple statistical model" for "predicting travel speed on a route segment" using the weighted average of a "historical speed" and a "current speed." *Id.*, 7:17-27; *see also* TESLA-1001, 19:3-11 ('426 Patent similarly teaches use of a weighted average to combine two different speed quantities). In this context, although Ran states that the current speed "is the real-time speed measured using field devices," a POSITA would have recognized that the real-time speed could alternatively be predicted using traffic information as well (as Ran further contemplates for its disclosed system). TESLA-1006, 7:24-25.

208. Thus, in view of Neiss-Ran and Andersson, a POSITA would have known of conventional ways (e.g., known models, weighted averaging, neural networks, etc.) and would have readily configured the Neiss-Ran-Andersson system

to, compute the predicted speed (*probable speed*) by, e.g., combining (using, e.g., models, weighted averages, etc.) the following speed values—the historical driving speeds (which further include historical driving data for similar segments, as Andersson contemplates) and the speeds based on real-time traffic information (in view of Neiss-Ran, *see* §VIII(A)(4)[1e] *supra*). TESLA-1006, 7:14-27.

209. Accordingly, because the determination of *efficient speed* (as described above for limitation [1f]) would be based at least in part on the *probable speed* and, as described in this section, the *probable speed* is further calculated using historical data for a similar route segment, the resulting Neiss-Ran-Andersson combination renders obvious determining the *efficient speed* based at least in part on the historical data for a similar route segment (as further explained in the following section).

### 3. Claim 3 Analysis

***[3] The method of claim 1, wherein the step of determining said efficient speed for the vehicle further comprises determining a historical speed for a similar destination.***

210. The Neiss-Ran-Andersson combination teaches claim 3 for the reasons described below.

211. As an initial matter, other than the verbatim recitation of this limitation in the '426 Patent's specification, there is no other specification disclosure for this limitation. *See generally* TESLA-1001. Thus, the '426 Patent does not explain what

a “similar destination” is, or how the determination of the efficient speed would include or use determining a historical speed for such a “similar destination,” as required by the claim. *See generally id.*

212. The only guidance of using a “similar” route comes in the context of the specification describing “an efficient speed” as “a function of ... any historical behavior of the vehicle taking the same (or a similar) route.” *Id.*, 18:65-19:2. Thus, in the context of the specification and the claims (which recite “segment destinations”), a POSITA would have understood that the claimed “similar destination” encompasses route segments that have endpoints (i.e., ***destinations***) and have “similar” characteristics as the route segment for which an “efficient speed” is computed.

213. As explained above in §VIII(B)(2), the Neiss-Ran-Andersson combination would leverage Andersson’s teachings of computing predicted speeds for road segments based on historical speeds driven on similar road segments (with respective endpoints or destinations). TESLA-1007, 4:47-60 (correlating historical driver behavior (*i.e.*, historical speeds associated with a driver) with road attributes to predict a driver’s speed on a route segment); *see id.*, 1:44-3:18, 3:34-4:60, 5:62-6:14. For example, Andersson teaches that it receives “vehicle position information via a position input connection” and “extract[s] ... map information relating to ... a

current and an upcoming road section,” and then generates a “speed plan” for the current and the upcoming road section. *Id.*, 1:46-57. To generate the speed plans, Andersson’s system “extract[s] ... one or more road attributes” for each road section and “relat[es] the road attributes to [a] driver behavior model.” *Id.*, 2:4-15. The driver behavior model “predict[s] a driver[’]s intended speed on a road section never driven before.” *Id.*, 4:53-60. This is done by “studying a chosen set of road section attributes, ... such as curvature, road surface[,], and so on, and studying driver behavior at previously driven road sections showing similar attributes, as stored in the driver information database.” *Id.* A POSITA would have understood that the driver behavior at previous driven road sections with similar attributes includes previously-driven speeds on such similar road sections (given the separate recitation of road attributes and driver attributes/behavior, the latter being commonly associated with driving speed and other preferences of the driver).

214. Given Andersson’s teaching of using historical driving data/speed for similar travel segments in predicting speed and Ran’s recognition of predicting speed using real-time traffic data as well as historically driven speeds for such segments, it would have been straightforward and well within a POSITA’s skill level to compute the predicted speed based on both real-time traffic information as well

as historical driving data (which further includes historical data for similar road segments, as Andersson teaches). TESLA-1006, 7:28-29; *see* §VIII(B)(2) *supra*.

215. Indeed, Ran recognizes that historical data can be readily fused/combined with real-time traffic data, e.g., using “various types of models, including weighted average and fuzzy logic.” TESLA-1006, 18:29-31. And, as one example, Ran provides a “simple statistical model” for “predicting travel speed on a route segment” using weighted average of historical speeds (including those driven on similar segments) and speeds based on real-time traffic data. *Id.*, 7:17-27; *see also* TESLA-1001, 19:3-11 (similarly teaching use of a weighted average to combined two different speed quantities).

216. Thus, in view of Ran and Andersson and as explained above with reference to §VIII(B)(2) *supra*, the combination would be implemented by computing the predicted speed (i.e., the ***probable speed***) based on prediction models that, e.g., compute weighted averages of the following speed values—the historical driving speeds (which further include historical driving data for similar segments, as Andersson contemplates) and the speeds based on real-time traffic information (in view of Neiss-Ran, *see* §VIII(A)(4)[1e] *supra*). TESLA-1006, 11:23-42; TESLA-1007, 4:47-60.

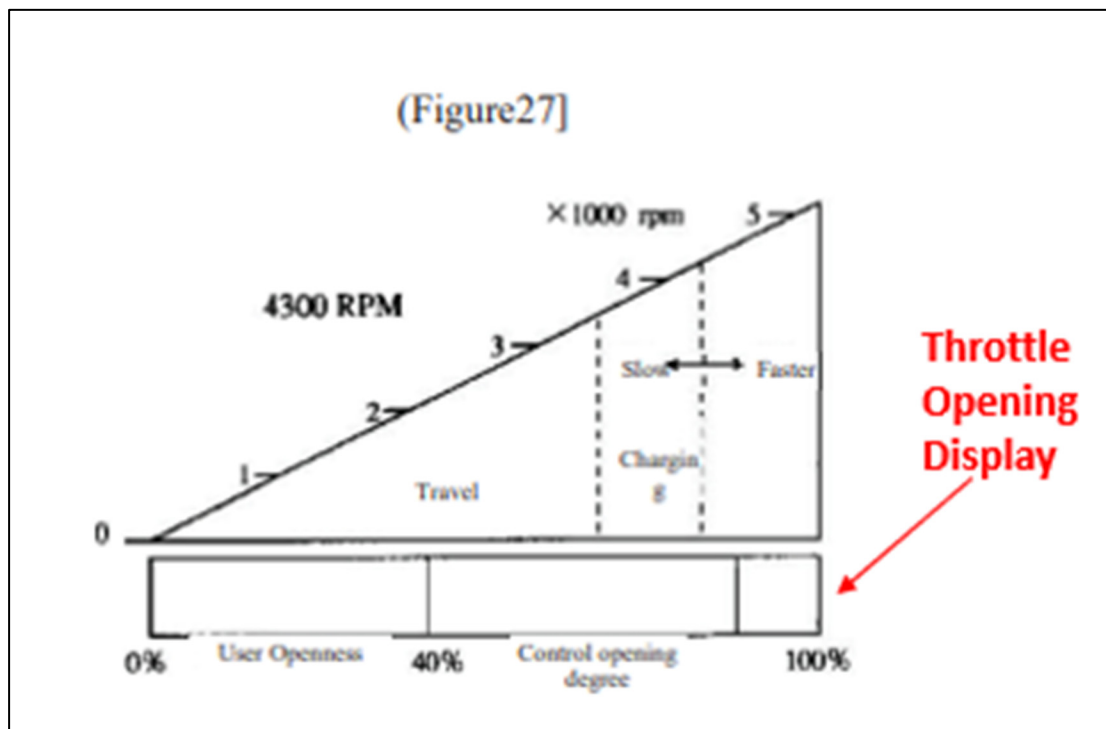
217. Moreover, because (1) the determination of *efficient speed* (as described above for limitation [1f]) would be based at least in part on the *probable speed* and (2) as described above (and in §VIII(B)(2) *supra*), the *probable speed* is further calculated using historical data for a similar route segment, the resulting Neiss-Ran-Andersson combination renders obvious determining the *efficient speed* based at least in part on the *historical speed for a similar [segment] destination* (e.g., a route segment that has an endpoint/destination and has similar characteristics).

**C. [GROUND 3] – Claim 4 is Rendered Obvious by Neiss-Ran-Hosoda**

**1. Overview of Hosoda**

218. Hosoda describes methods to display various types of information “so that changes in fuel efficiency can be easily understood.” TESLA-1009, Abstract; *see id.*, ¶¶[0001] (“The present invention relates to travel display devices”); [0002] (prior art application “discloses a hybrid electric vehicle that displays the state of efficiency of the engine and the state of energy transfer when driving”); [0003] (“it is necessary to inform the driver that fuel efficiency will decrease when driving at low fuel efficiency”); [0004] (the invention’s “purpose is to provide a driving display device in a hybrid electric vehicle that can easily understand changes in fuel efficiency based on the driving condition of the vehicle and the driving operation of

the driver and does not cause discomfort to the driver”); [0005]. Hosoda discloses a “[d]isplay control ECU 200” that “performs display control on the display 201,” which is “a liquid crystal display for displaying an instrument panel or a navigation screen, etc. opposite the driver’s seat.” *Id.*, ¶¶[0023]-[0024]. The display 201 is configured to display a variety of data, including the fuel and power storage remaining amount, the vehicle speed, the engine speed, the vehicle position, and also the throttle opening. *Id.*, ¶¶[0027], [0061], [0028]-[0064] (displaying information related to fuel and energy storage levels, fuel efficiency improvement rates, drive force transmissions, engine RPM, and change modes) .



TESLA-1009, FIG. 27 (annotated)

219. As shown above in Figure 27, in addition to the engine speed, the display provides the “ratio of each throttle opening” using “a message and an image.” *Id.*, ¶[0061]. Specifically, Hosoda teaches displaying “at the bottom of the allocation ratio display” the “throttle opening (user opening) by the operator’s accelerator pedal operation according to the engine speed” and the “throttle opening (control opening) by the engine and motor control ECU control.” *Id.*

## 2. Combination of Neiss, Ran, and Hosoda

220. As described in §VIII(A)(3) and with reference to limitation [1g] *supra*, the Neiss-Ran combination generally describes a predictive cruise control system configured to determine an *efficient speed* for upcoming route segments, where a *calculated power* corresponding to the *efficient speed* is applied *to the vehicle engine to travel said segment destination*. See §VIII(A) *supra*. Hosoda describes a method for displaying the power applied to a vehicle to an operator of the vehicle via an in-vehicle display. See §VIII(C)(1) *supra*.

221. A POSITA would have been motivated to combine the teachings of Hosoda and Neiss-Ran and would have had a reasonable expectation of success in doing so, to arrive at a combined system (referred to herein as Neiss-Ran-Hosoda) that implements a display in the vehicle and notifies the vehicle operator, via this display, of the calculated applied power that is used to control the engine.

222. In particular, in Neiss-Ran, the optimization algorithm is used to compute various vehicle control parameters, including a *calculated power* corresponding to the *efficient speed* (e.g., throttle pedal positions that maps to the amount of power/fuel to inject) and stores the values for these computed optimal parameters in a look-up table in memory.

223. A POSITA would have found it obvious, in view of Hosoda, to implement a mechanism to notify the vehicle's operator of the calculated applied power. In the relevant time period, it was well known to notify the operator of the vehicle of various parameters associated with the control of the vehicle. This is further corroborated by, e.g., Suzuki (TESLA-1020, 4:14-28) and Kojima (TESLA-1013, ¶¶[0025]-[0029]) (both teaching vehicle displays that display information about the vehicle, including vehicle speed, engine speed, remaining fuel, and other information). And, as explained above, Hosoda provides one such system in which a display unit within the vehicle is configured to display throttle-related information. *See* §VIII(C)(1) *supra*.

224. When implementing Hosoda's teachings in combination with those of Neiss and Ran, a POSITA would have implemented a display in the vehicle and such display would be configured to notify the operator of the applied power (e.g., the throttle pedal position output by the Neiss-Ran optimization algorithm).

225. A POSITA would have been motivated to implement a display and use the same to notify the operator of the vehicle of the calculated applied power for at least the following reasons.

226. ***First***, a POSITA would have recognized the benefit of notifying a vehicle operator of the calculated applied power to the vehicle. As Hosoda explains, displaying such information on a vehicle display enables the vehicle's operator to "easily understand a change in fuel efficiency" so that the vehicle operator is aware of the vehicle's performance. TESLA-1009, ¶¶[0001]-[0004].

227. ***Second***, a POSITA would have been motivated to pursue this implementation because doing so is merely the application of known techniques (*e.g.*, displaying statistics related to a vehicle at a display in the vehicle) to a known system (*e.g.*, Neiss-Ran's predictive cruise control system that computes and stores optimal velocities and corresponding throttle values to control the vehicle on upcoming route segments/frames) to yield predictable results (*e.g.*, displaying outputs of the optimal control parameters at a display in a vehicle).

228. A POSITA would have had a reasonable expectation of success in implemented the combination in the above manner for at least the following reasons. First, implementing displays in vehicles was well-known and conventional in the relevant time period. *See, e.g.*, TESLA-1009, Abstract; *see also* Suzuki (TESLA-

1020, 4:14-28) and Kojima (TESLA-1013, ¶¶[0025]-[0029]) (both teaching vehicle displays that display information about the vehicle, including vehicle speed, engine speed, remaining fuel, and other information). Also, in Neiss-Ran, the optimization algorithm already computes various vehicle control parameters, including a *calculated power* corresponding to the *efficient speed* (e.g., throttle pedal positions that maps to the amount of power/fuel to inject) and stores the values for these computed optimal parameters in a look-up table in memory. And, before the Critical Date, displays in vehicles were well-known and conventional means by which the vehicle notified an operator of the vehicle's operations, including the control parameters that are used to control the vehicle's engine, as Hosoda also teaches. See §VIII(C)(1) *supra*.

229. Thus, given (1) the conventional and well-known use of displays in vehicles to notify vehicle operators of vehicle control parameters and (2) Hosoda's teaching of displaying throttle-based control values on an in-vehicle display, it would have been straightforward and predictable to modify Neiss-Ran in view of Hosoda to implement an in-vehicle display and provide for display thereon the already-computed and stored vehicle control parameter values in Neiss-Ran, including the parameter related to throttle-based power being applied to the vehicle's engine.

### 3. Claim 4 Analysis

***[4] The method of claim 1, further comprising notifying the operator of the vehicle of the calculated applied power.***

230. Neiss-Ran-Hosoda teaches Claim 4 for the reasons described below.

231. As described above in §§VIII(C)(1)-(2), Neiss-Ran-Hosoda leverages Hosoda’s teachings relating to displaying throttle-related values on a vehicle display. TESLA-1009, ¶[0024] (disclosing a “display 201 [that] is a liquid crystal display for displaying an instrument panel or navigation screen, etc. opposite the driver’s seat”); *see id.*, ¶¶[0001]-[0005], [0023]-[0064] (providing examples of data that can be displayed). On this display, throttle related information (e.g., the “ratio of each throttle opening,” “throttle opening (user opening) by the operator’s accelerator pedal operation according to the engine speed,” and the “throttle opening (control opening) by the engine and motor control ECU control”) may be displayed. *Id.*, ¶[0061].

232. Neiss-Ran’s system already calculated and stored, in memory, throttle related information that corresponds to the calculated ***efficient speed*** (that is, the ***calculated applied power***). *See* §VIII(A)(4)[1g] *supra*. And, as described above in §VIII(C)(3) *supra*, a POSITA would have been motivated and would have had a reasonable expectation of success in displaying the already-calculated throttle related information at an in-vehicle display. *See* §VIII(C)(2) *supra*. In combination

then, a POSITA would have found obvious to implement Hosoda's display within the vehicle associated with the Neiss-Ran cruise control system to therefore display the *calculated applied power* output by the Neiss-Ran optimization algorithm—thereby *notifying the operator of the vehicle* of the same.

## IX. ADDITIONAL REMARKS

233. I currently hold the opinions set expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

234. I hereby declare that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of the Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Dated: February 5, 2024

By: \_\_\_\_\_