

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

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES

PATENT NO. 7,925,426 PURSUANT TO 35 U.S.C. §§ 311–319,

37 C.F.R. § 42

TABLE OF CONTENTS

I.	IPR REQUIREMENTS	1
A.	Grounds for Standing.....	1
B.	Challenge and Relief Requested.....	1
C.	Claim Construction	2
D.	Level of Ordinary Skill in the Art.....	2
II.	THE '426 PATENT.....	3
A.	Overview	3
B.	Prosecution History.....	4
III.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	5
A.	[GROUND 1] – Neiss-Ran Render Obvious Claims 1-2	5
1.	Neiss	5
2.	Ran.....	11
3.	Combination of Neiss and Ran.....	16
4.	Claim Analysis	22
(a)	Claim 1	22
(b)	Claim 2	65
B.	[GROUND 2] – Neiss-Ran-Andersson Render Obvious Claim 3	66
1.	Andersson.....	66
2.	Combination of Neiss, Ran, and Andersson	68
3.	Claim 3 Analysis	72
C.	[GROUND 3] – Neiss-Ran-Hosoda Render Obvious Claim 4	75
1.	Hosoda.....	75
2.	Combination of Neiss, Ran, and Hosoda	77
3.	Claim 4 Analysis	79
IV.	PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION.....	80
A.	35 U.S.C. §325(d) – Discretionary Denial Under 325(d) is Not Warranted.....	80
B.	The Fintiv Factors Favor Institution and No Denial Under §314(a).....	81
1.	Factor 1: Institution Will Enable a Stay	81
2.	Factor 2: Uncertain District Court Schedule.....	82
3.	Factor 3: Early Stage of Parallel Proceedings.....	82
4.	Factor 4: Petition Raises Unique Issues	83
5.	Factor 5: Tesla’s Involvement in the WDTX Litigation.....	83
6.	Factor 6: Other Considerations Support Institution	83
V.	CONCLUSION AND FEES	84

VI.	MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1).....	84
A.	Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....	84
B.	Related Matters Under 37 C.F.R. § 42.8(b)(2)	85
C.	Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3).....	85
D.	Service Information	85

LIST OF EXHIBITS

TESLA-1001	U.S. Patent No. 7,925,426 (“the ’426 Patent”)
TESLA-1002	Excerpts from the Prosecution History of the ’426 Patent
TESLA-1003	Declaration of Dr. Papanikolopoulos, Ph.D.
TESLA-1004	Curriculum Vitae of Dr. Papanikolopoulos, Ph.D.
TESLA-1005	U.S. Patent Publication No. 2004/0068359 (“Neiss”)
TESLA-1006	U.S. Patent No. 6,317,686 (“Ran”)
TESLA-1007	U.S. Patent No. 6,836,719 (“Andersson”)
TESLA-1008	Japanese Patent Pub. No. 11220807 (“Hosoda Original”)
TESLA-1009	Certified Translation of Japanese Patent Pub. No. 11220807 (“Hosoda”)
TESLA-1010	RESERVED
TESLA-1011	U.S. Patent No. 6,314,347 (“Kuroda”)
TESLA-1012	U.S. Patent Publication No. 2006/0195249 (“Fosseen”)
TESLA-1013	U.S. Patent Publication No. 2005/0228556 (“Kojima”)
TESLA-1014	U.S. Patent 6,843,337 (“Matsubara”)
TESLA-1015	U.S. Patent 6,895,320 (“Bauer”)
TESLA-1016	U.S. Patent Publication No. 2004/0084237 (“Petrie”)
TESLA-1017	U.S. Patent Publication No. 2005/0004742 (“Michi”)

TESLA-1018	U.S. Patent Publication No. 2005/0216793 (“Entin”)
TESLA-1019	U.S. Patent Publication No. 2004/0230376 (“Ichikawa”)
TESLA-1020	U.S. Patent No. 6,396,394 (“Suzuki”)
TESLA-1021	Complaint, <i>iQar Inc. v. Tesla, Inc.</i> , 1:23-cv-00694, W.D. Tex., June 23, 2023
TESLA-1022	Proof of Service of Complaint, <i>iQar Inc. v. Tesla, Inc.</i> , 1:23-cv- 00694, W.D. Tex., June 23, 2023
TESLA-1023	Memorandum, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation (USPTO June 21, 2022) (“Director’s Guidance”)
TESLA-1024	Motion to Transfer Venue
TESLA-1025	Excerpts from Heinle’s Newbury House Dictionary of American English (4 th Ed.)
TESLA-1026	Excerpts from Webster’s College Dictionary
TESLA-1027	U.S. Patent No. 7,013,205 (“Hafner”)

LISTING OF CHALLENGED CLAIMS

Claim 1	
[1pre]	A method of managing the power consumption of a vehicle, comprising:
[1a]	(a) a computer device receiving a route;
[1b]	(b) said computer device segmenting said route into segment destinations;
[1c]	(c) said computer device for each of said segment destinations:
[1d1]	(i) calculating an optimized speed by minimizing power consumption over said segment destination,
[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;
[1e]	(ii) calculating a probable speed;
[1f]	(iii) determining an efficient speed based on said optimized speed and said probable speed; and

[1g]	(iv) applying the calculated power corresponding to said determined efficient speed to the vehicle engine to travel said segment destination.
Claim 2	
[2]	The method of claim 1, further comprising continuously calculating said efficient speed until a final destination is reached.
Claim 3	
[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.
Claim 4	
[4]	The method of claim 1, further comprising notifying the operator of the vehicle of the calculated applied power.

Tesla, Inc. (“Petitioner” or “Tesla”) petitions for *Inter Partes* Review (“IPR”) of claims 1-4 (“Challenged Claims”) of U.S. Patent 7,925,426 (“’426 Patent”). This Petition demonstrates at least a reasonable likelihood that Tesla will prevail as to at least one of the Challenged Claims.

I. IPR REQUIREMENTS

A. Grounds for Standing

The ’426 Patent is available for IPR. This Petition is being filed within one-year of service of a complaint against Tesla. TESLA-1021; TESLA-1022. Tesla is not barred/estopped from requesting this review.

B. Challenge and Relief Requested

Tesla requests an IPR of the Challenged Claims on the below grounds. Dr. Papanikolopoulos provides supporting testimony in his Declaration. TESLA-1003, ¶¶1-234.

Ground	Claim(s)	35 U.S.C. §103
1	1-2	Neiss-Ran
2	3	Neiss-Ran-Andersson
3	4	Neiss-Ran-Hosoda

The ’426 Patent claims an earliest priority date of November 17, 2005. TESLA-1001, Cover. Tesla does not concede that the ’426 Patent is entitled to this

alleged priority date, but applies prior art predating this date (Critical Date).

Reference	Filed	Published	Basis
Neiss	10/4/2002	4/8/2004	§102(b)
Ran	07/21/2000	11/13/2001	
Hosoda ¹	1/30/1998	8/10/1999	
Andersson	10/15/2002	12/28/2004	§§102(a)/(e)

C. Claim Construction

No claim constructions are necessary because “claim terms need only be construed” if “necessary to resolve the controversy.” *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011); TESLA-1003, ¶27. Petitioner reserves the right to respond to any constructions offered/adopted by Patent Owner or the Board. Petitioner applies art herein consistent with Patent Owner’s allegations of infringement in the co-pending litigation.

D. Level of Ordinary Skill in the Art

A person of ordinary skill in the art (“POSITA”) relating to the subject matter of the ’426 Patent as of the Critical Date would have had a bachelor’s degree in electrical engineering, computer engineering, mechanical engineering (including

¹ Citations to Hosoda (TESLA-1008) herein are to its translation (TESLA-1009).

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. TESLA-1003, ¶¶1-19, 25-26; TESLA-1004. Additional graduate education could substitute for professional experience, and *vice versa. Id.*

II. THE '426 PATENT

A. Overview

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; TESLA-1003, ¶¶28-42. 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, hybrid, and electric vehicles. *See id.*, 6:20-30.

The claimed method includes “receiving a route at a computer device” and “segmenting said route into a plurality of segment destinations.” *Id.*, 26:27-29; *see id.*, 11:27-30, 11:46-47, 13:36-42, 13:45-47, 13:49-56; TESLA-1003, ¶¶30-31.

The claimed method further recites operations to be performed by the computing device for each segment destination. *See generally id.*; TESLA-1003,

¶¶32-42. Specifically, for each segment destination, the claimed method recites the computer device performing the following operations:

- (1) “calculating an optimized speed by minimizing power consumption over said segment destination,” where one or more types of information are “received as input to” for this calculation (TESLA-1001, 26:30-39; *see id.*, 7:21-51, 8:21-40, 8:61-9:6, 9:55-10:2, 25:33-34 (providing examples of information used for calculation); TESLA-1003, ¶¶34-39 (citing TESLA-1001, 14:30-49).
- (2) “calculating a probable speed” (TESLA-1001, 26:40; TESLA-1003, ¶40 (specification does not use the phrase “probable speed”)).
- (3) “determining an efficient speed based on the optimized and probable speeds” (TESLA-1001, 26:41-42; *see id.*, 19:3-6 (describing, in one example, weighted average of optimized speed and probable optimized speed); TESLA-1003, ¶41).
- (4) “applying calculated power corresponding to the determined efficient speed to the vehicle engine” (TESLA-1001, 26:43-45, 18:62-65; TESLA-1003, ¶42).

B. Prosecution History

During prosecution, the examiner initially rejected the pending independent claims in view of Matsubara (TESLA-1014), and rejected pending claim 19 (which

recited, in part, calculating an energy efficient speed to travel to the segment destination, ... or calculating the applied power from the target speed) based on Matsubara and Bauer (TESLA-1015). TESLA-1002, 62-76, 142-146, 48-54; TESLA-1003, ¶¶43-45.

Applicant subsequently cancelled claims 1-16 and retained then-pending method claim 7, which was amended to recite, among other claim elements, the “optimized speed,” “probable speed,” and “efficient speed” limitations present in issued claim 1. TESLA-1002, 41-44. The claims were then allowed, with the Reasons for Allowance stating that “these prior art do not make obvious” the entire amended method claim (which issued as claim 1). *Id.*, 24-25; TESLA-1003, ¶¶46-47.

III. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. [GROUND 1] – Neiss-Ran Render Obvious Claims 1-2

1. Neiss²

Neiss describes a “predictive cruise control system [that] utilizes information about the current vehicle position, as well as upcoming terrain” “to determine, as a

² Descriptions of references and combinations thereof are incorporated into each mapping that includes citations to these references. Emphasis added unless otherwise indicated.

function of position, a desired speed inside [a] speed band...to maximize fuel savings.” TESLA-1005, Abstract, ¶¶[0031], FIG. 2; TESLA-1003, ¶¶48-67.

Neiss’s cruise control unit “minimizes vehicle operating costs” by optimizing an overall “cost function” that is defined based on “environmental parameters, vehicle parameters, vehicle operating parameters[,] and route parameters.” *Id.*, ¶¶[0001], [0015]. As the vehicle drives, “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 vehicle’s current position. *Id.*; TESLA-1003, ¶50.

Neiss’s “cost function” (J) is defined mathematically (below) to reflect one or more objectives, including “consum[ing] less fuel” and “activat[ing] the brakes as little as possible”:

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

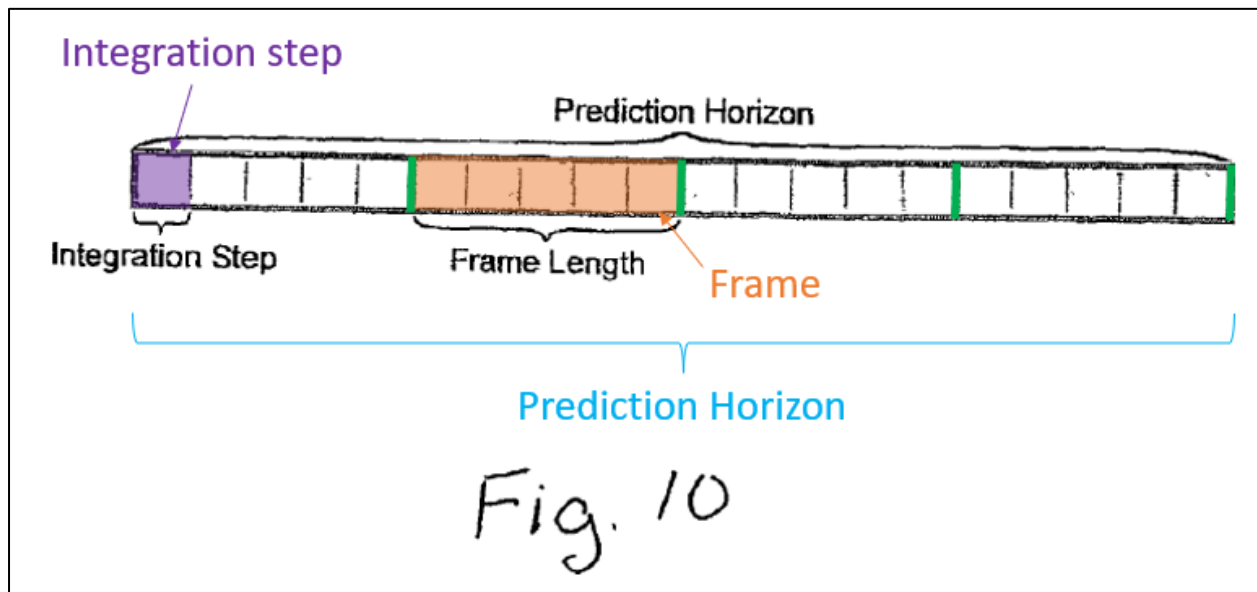
TESLA-1005, ¶¶[0082]-[0126].

As shown, this function includes an objective to minimize fuel consumption (represented as J_{fuel}), which is mathematically expressed below:

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

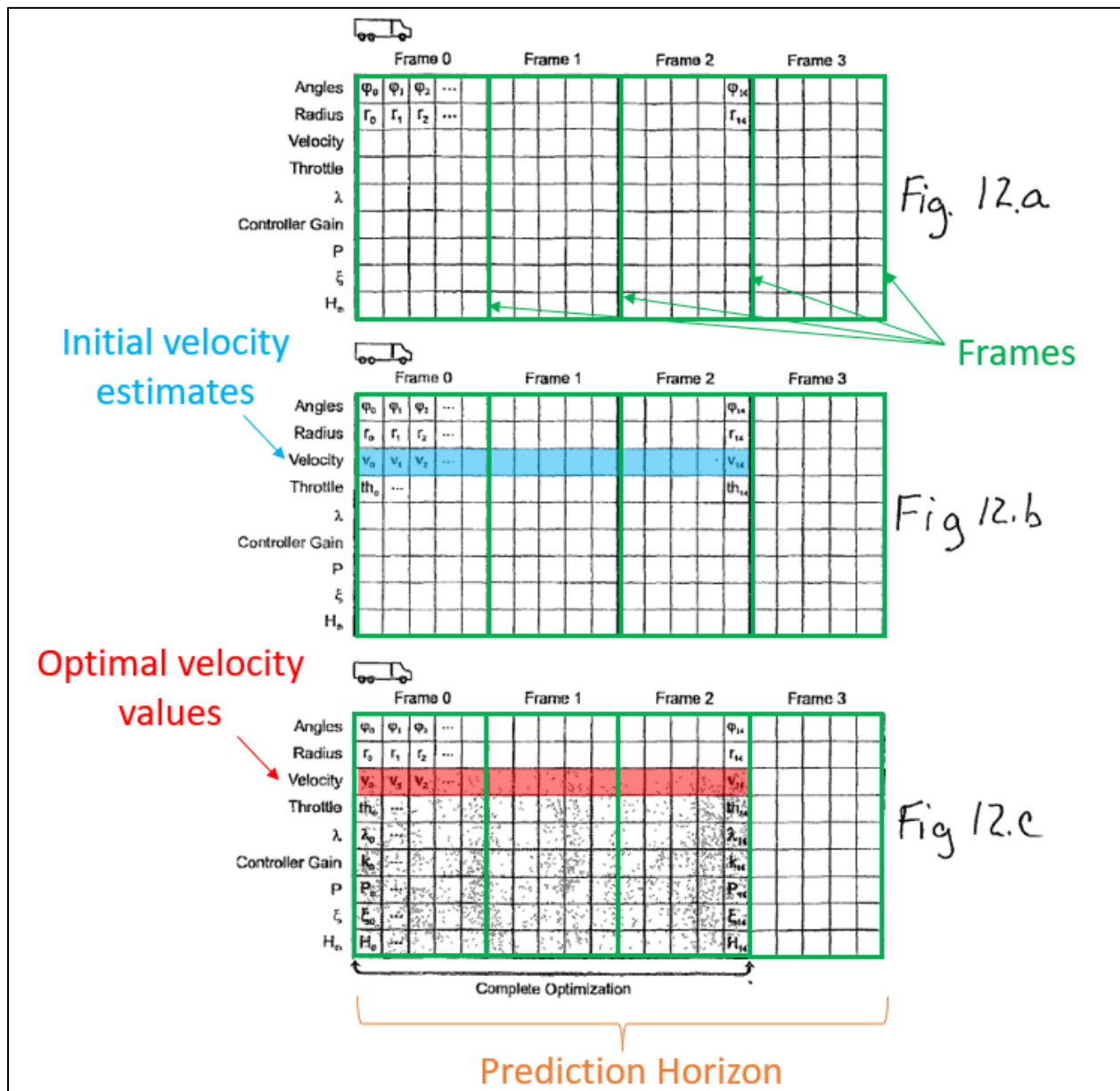
Id., ¶¶[0081]-[0082], [0097]; TESLA-1003, ¶¶51-55. This equation is represented as “a function of [at least] vehicle speed, road gradient[,] and throttle position” and specifically, represents a summation of the velocity v_k and control values th_k at N points in a “Prediction Horizon.” *Id.*, ¶¶[0097]-[0098], [0162], [0180], claims 10-11, Abstract.

The “Prediction Horizon” is “a predetermined distance...along the route ahead of the vehicle.” *Id.*, ¶[0015]. As illustrated in Figure 10, *infra*, the “Prediction Horizon” “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].



TESLA-1005, FIG. 10 (annotated)

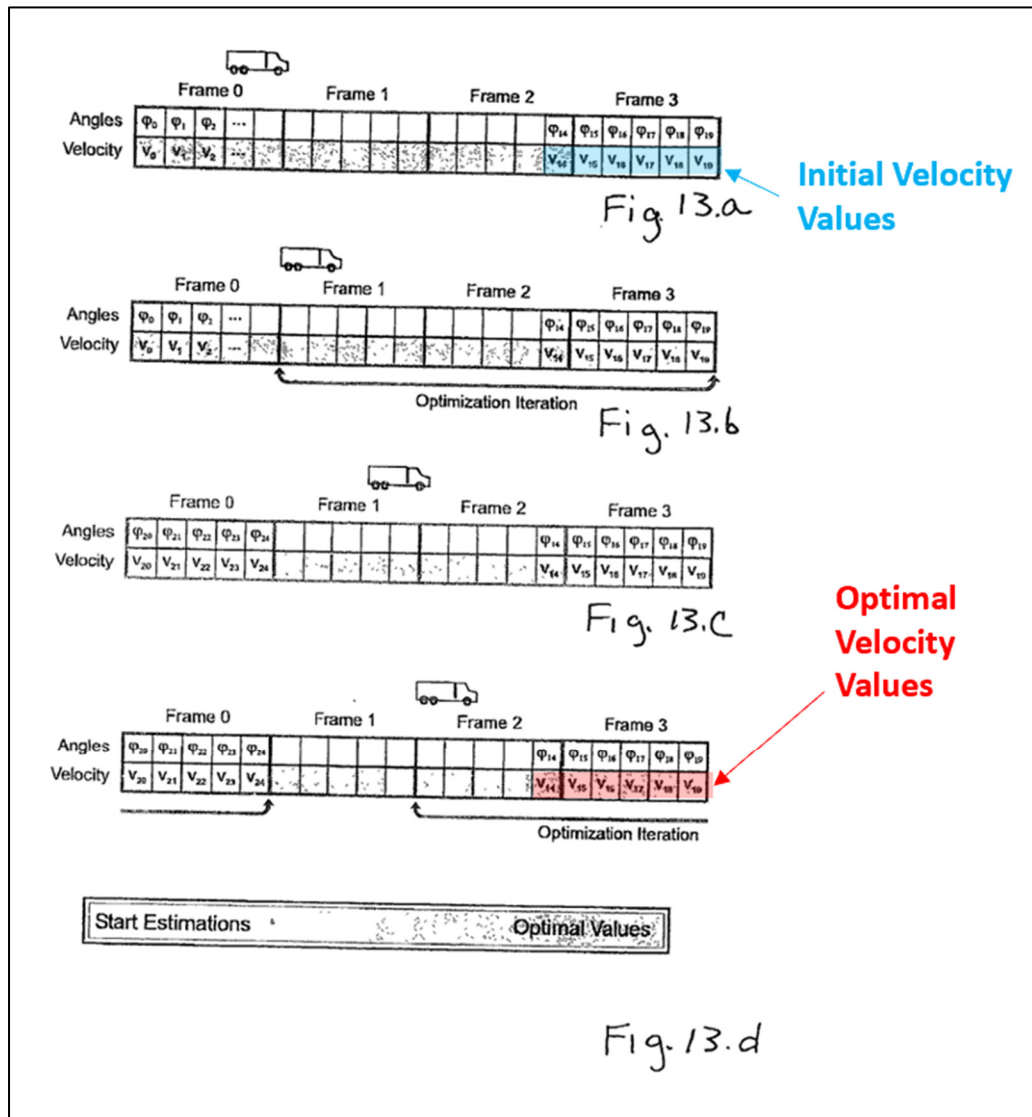
As part of the optimization (and corresponding minimization of the vehicle operating cost function), Neiss's cruise control system uses initial estimates of vehicle control parameters to compute a first set of optimized vehicle control parameters, including vehicle speed, for frames along the Prediction Horizon. TESLA-1005, ¶¶[0015], [0171]-[0181], [0186], [0048], [0149]-[0160], [0015], [0162], FIGS. 12a-c; TESLA-1003, ¶¶51-59. The estimates and the 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)

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.*, ¶¶[0178]-[0181], [0188]-[0191], FIGS. 13a-d (below); TESLA-

1003, ¶¶57-63. With each iteration then, the vehicle control parameters, including velocity, are re-optimized and “approach closer to the optimal values” (annotated red in FIG. 13d below). *Id.*



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

Per Neiss, 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.*; TESLA-1003, ¶¶62-66.

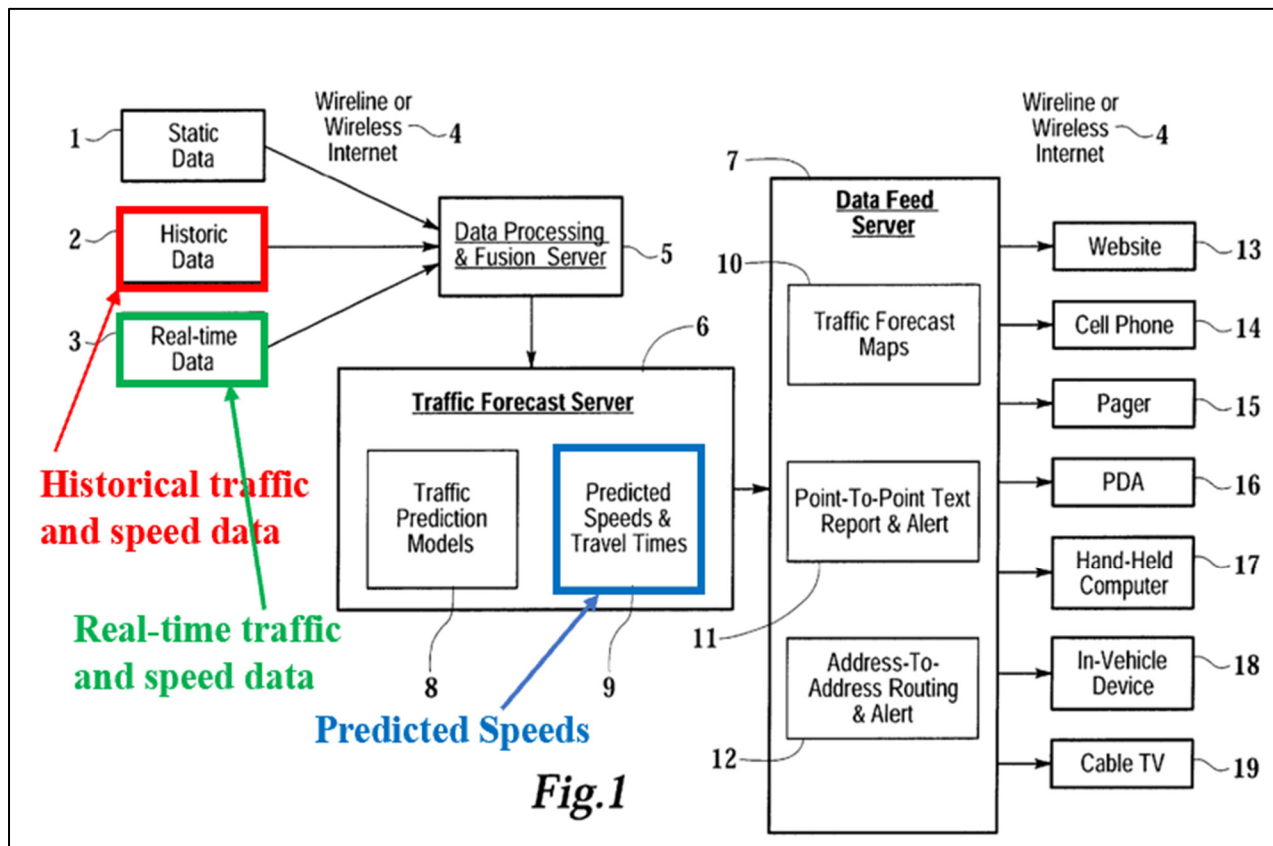
When the vehicle reaches a particular position, the calculated optimized parameters for the corresponding position are read from the lookup table and provided as inputs to a controller “which controls the vehicle to follow the calculated velocity trajectory.” *Id.*, ¶¶[0041]-[0042]; *see id.*, ¶[0188].

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 solved by the ’426 Patent (*e.g.*, how to determine speeds at which a vehicle should operate, *e.g.*, given road/operating conditions). TESLA-1003, ¶67.

2. Ran

Ran describes a system and method of predicting a travel time and speed for a route. TESLA-1006, 1:12-13; TESLA-1003, ¶¶68-83.

An example of Ran’s system is shown below:

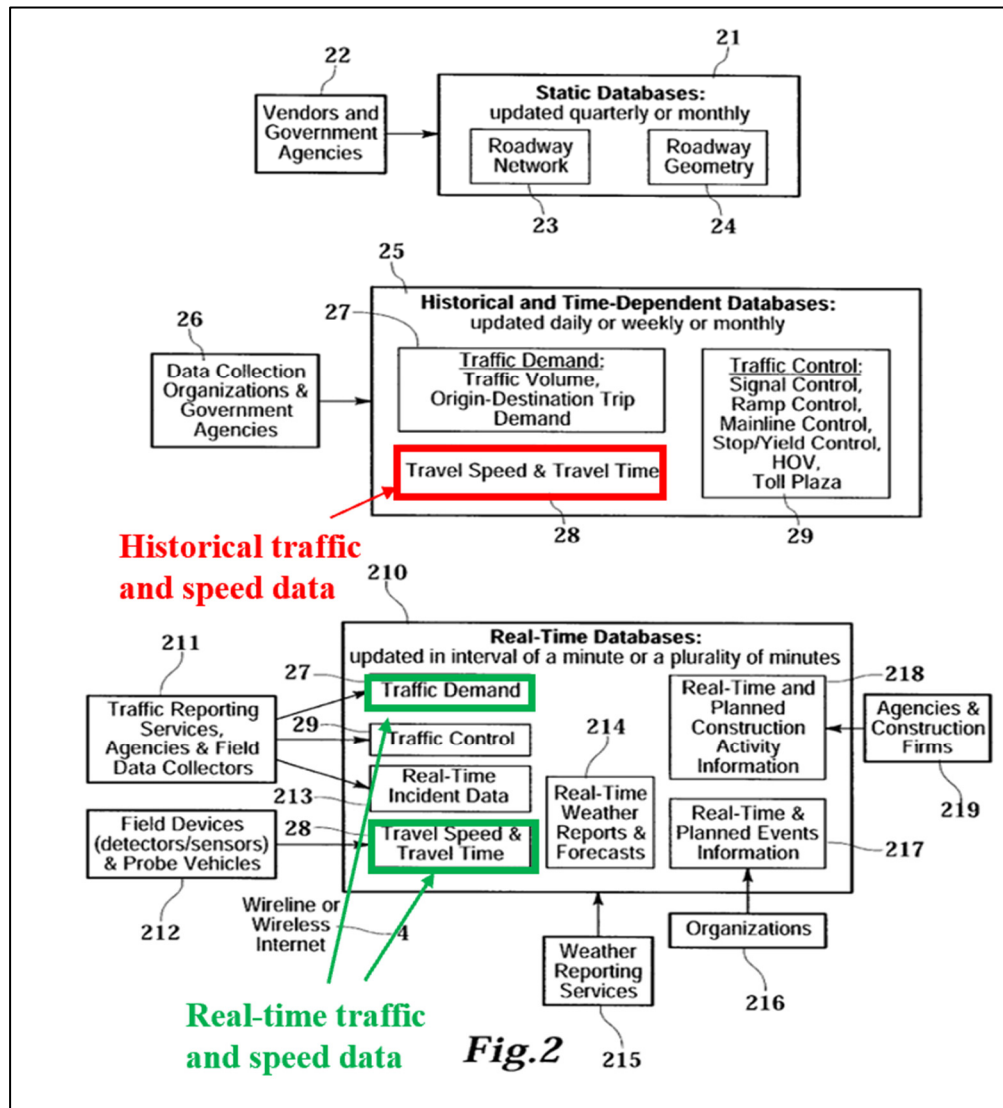


TESLA-1006, FIG. 1

As shown, 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.

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-1003, ¶¶69-71.



TESLA-1006, FIG. 2 (annotated)

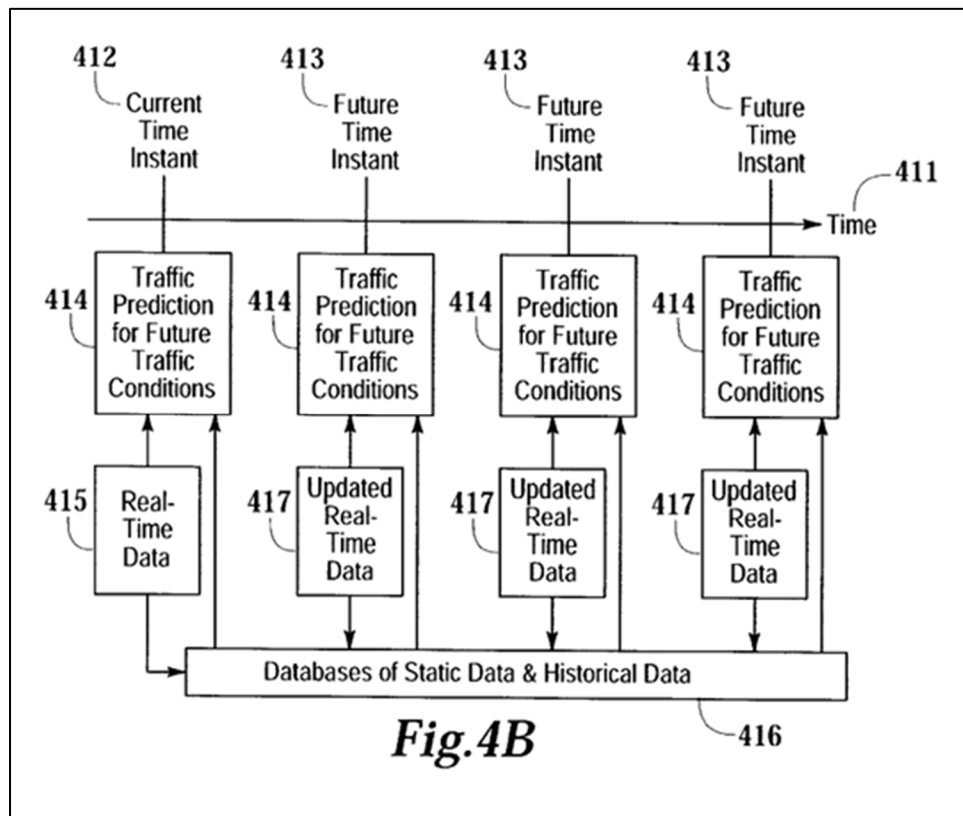
The “static databases 21 define the route segment network 23,” providing information about the road (e.g., links, length, landmarks, speed limits, etc.) along a route. TESLA-1006, 17:38-45.

The “historical and time-dependent databases 25” include “traffic demand 27, travel speed and travel time 28, and traffic control 29” databases,” which collectively provide historical traffic demand, speeds, travel times, and traffic control-related information. *Id.*, 17:48-58.

The “real-time databases 210” include multiple databases, receiving and providing real-time traffic information, including real-time speed (received from, e.g., field devices, traffic reporting services, third party or government agencies, etc.). *Id.*, 17:64-18:17.

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.

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 id.*, FIG. 4B *infra*, 19:19-22, 19:43-49, 20:1-17. For example, “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, 19:19-22, 18:53-56 (providing examples of these models); TESLA-1003, ¶¶72-80.



TESLA-1006, FIG. 4B

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)

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.*; TESLA-1003, ¶¶80-81. A vehicle speed can thus be predicted in light of traffic conditions by applying a function of the existing traffic conditions to a freeflow speed for the route segment. TESLA-1006, 11:23-42; TESLA-1003, ¶¶80-81.

In some instances, the predicted vehicle speed can be sent to various devices, including an “in-vehicle device 18.” TESLA-1006, 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.*, 23:29-32; TESLA-1003, ¶82.

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.*, how to determine speeds at which a vehicle should operate, *e.g.*, given road/operating conditions). *Id.*, ¶83.

3. Combination of Neiss and Ran

As described in §III(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 vehicle’s route. TESLA-1003, ¶¶48-67. The system “adapt[s]

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]. Ran similarly discloses that a vehicle’s speed on a route’s segments may be constrained by traffic, and provides techniques for computing a vehicle’s predicted/probable speed that accounts for real-time traffic conditions. TESLA-1006, 11:23-42, 7:17-29, FIG. 3, 18:18-49; TESLA-1003, ¶¶68-83.

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 speeds*) that account for at least real-time traffic conditions (as disclosed in Ran) at those upcoming route segments. TESLA-1003, ¶86. 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. *Id.*, ¶¶84-86, 91; TESLA-1006, 1:12-36, 4:9-29, 7:1-5, Tables 1-3, 11:23-64, 14:8-15, 17:64-18:61, 20:9-17, 21:43-49. In Neiss-Ran, Neiss’s optimization algorithm would be modified by leveraging Ran’s real-time traffic-based predicted speeds (*calculate[ed] probable speeds*) 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 speeds*) that can be used to control the vehicle. TESLA-1003, ¶¶84-86.

A POSITA would have been motivated to combine Neiss’s and Ran’s teachings for multiple reasons.

First, Neiss seeks to use “real-time” driving conditions to adapt its optimization algorithm (*see* 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/frames using real-time traffic information. TESLA-1003, ¶¶87-88. Indeed, a POSITA would have been motivated to perform Neiss’s optimization for upcoming segments/frames using Ran’s predicted speed values based on real-time traffic conditions in those segments/frames—instead of using a current speed for an existing frame—to account for more accurate, prevailing speeds in those segment/frames, which in turn would enable a more accurate and realistic speed optimization. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421 (2007); MPEP 2143; TESLA-1003, ¶88.

In particular, Neiss’s PCC system implements an optimization algorithm that computes optimal vehicle control parameters, including optimal velocity, for segments/frames of the upcoming route, and stores these computed parameters in a

look-up table. TESLA-1005, ¶¶[0092]-[0098], [0178]-[0191]; §III(A)(1) *supra*. As part of the optimization, the system adapts the control values for the upcoming frames in the lookup table using the vehicle’s “current velocity”—in the current frame where the vehicle is driving. *See, e.g.*, TESLA-1005, ¶¶[0167], [0190].

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 where traffic conditions might be different from those in the current frame/segment. TESLA-1003, ¶¶89-90. A POSITA would have thus been motivated to modify Neiss’s system by performing the optimization using predicted speeds that account for real-time traffic conditions (per Ran) in those upcoming frames/segments (instead of using current velocity of an existing frame where the vehicle is driving) because doing so would yield a more accurate and realistic speed optimization (as described above). TESLA-1003, ¶¶91-95 (citing TESLA-1011, 6:59-7:34; TESLA-1006, 11:23-64, 17:25-18:61, 20:9-21:49, 1:12-36, Tables 1-3).

Second, a POSITA would have also been motivated to combine these references’ teachings to enable more resource efficient optimization calculations. *Id.*, ¶96. Neiss acknowledges that its optimization iterations are computing resource intensive (*see* 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 be performed or the system would switch to a failure mode and stops all computations. TESLA-1003, ¶96 (citing TESLA-1005, ¶¶[0167], [0190]). Using real-time traffic and speed information at upcoming segments (per 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. TESLA-1003, ¶96.

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, per Ran) 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). *See KSR*, 550 U.S. at 415-421; TESLA-1003, ¶97 (citing TESLA-1007, 1:52-56).

A POSITA would have had a reasonable expectation in combining Neiss’s and Ran’s teachings in the above-described manner. TESLA-1003, ¶¶98-99. Each

of Neiss and Ran's systems/components would continue to operate in the same manner they did prior to combination. *Id.*, ¶98. Specifically, Neiss's optimization algorithm would continue to be implemented by its onboard computer and would operate iteratively on upcoming route segments/frames. *Id.*, ¶¶98-99. Ran's predicted speed computation would continue to be performed by traffic prediction models (as implemented at an in-vehicle computer, such as Neiss's onboard computer) using, at least, real-time traffic data from real-time data sources. *Id.*, ¶98 (citing TESLA-1006, 17:7-16, 18:18-31).

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 optimal velocities to account for real-time conditions. *See* §III(A)(1) *supra*. And, Ran teaches calculating expected or predicted velocities for upcoming route segments using real-time traffic conditions (e.g., as obtained from real-time databases and traffic sources). *See* §III(A)(2) *supra*. Thus, it would have been straightforward and well-within the skillset of a POSITA to configure Neiss's onboard computer and the associated optimization algorithm, to substitute its use of current velocity for an existing segment/frame, with the real-time traffic-informed, more accurate velocity for the frame(s)/segment(s) for which the optimization is being performed. TESLA-1003, ¶¶98-99.

4. Claim Analysis

(a) Claim 1

[1pre]

If the preamble is determined to be limiting, Neiss-Ran teaches the preamble. TESLA-1003, ¶¶100-106.

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, corresponds to *managing the power consumption of the vehicle*.³ TESLA-1001, 6:14-17 (“optimizing the power supplied to the engine typically means minimizing the energy loss from the vehicle, thereby increasing fuel efficiency”), 6:14-30 (noting applicability of techniques to fuel, hybrid, and electric-powered vehicles); TESLA-1003, ¶101 (citing, e.g., TESLA-1009; ¶¶[0015], [0025], FIG. 10) (fuel is an example power source; managing power consumption for fuel-powered vehicles thus encompasses managing fuel consumption).

In particular, Neiss-Ran’s “predictive cruise control system utilizes information about the current vehicle position, as well as upcoming terrain...to save fuel....” TESLA-1005, Abstract; TESLA-1003, ¶102. The predictive cruise control system minimizes a vehicle operating cost function, “based on an analysis of vehicle

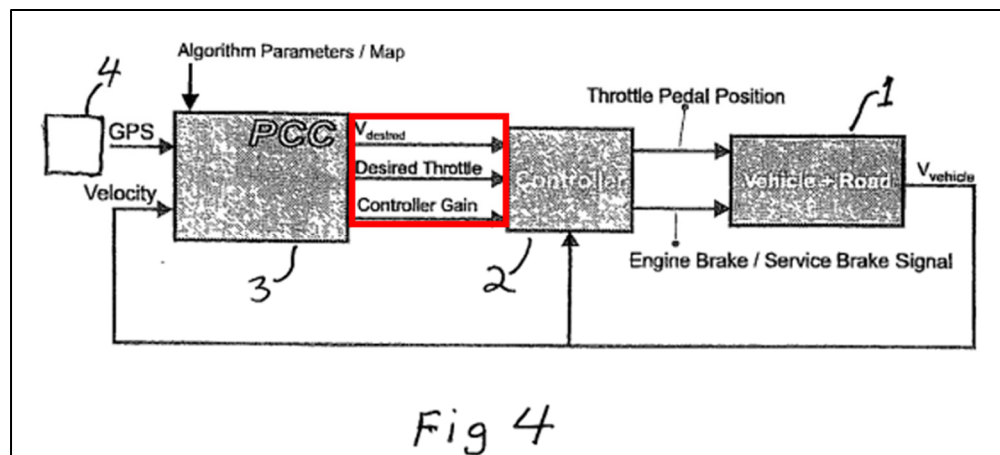
³ Claim language is in bold italics throughout this Petition.

operating parameters and route information regarding a route section ahead of the vehicle.” TESLA-1005, ¶¶[0001], [0015], [0081]-[0082]; TESLA-1003, ¶¶102-103 (citing TESLA-1005, ¶[0081]; explaining that a cost function defines a desired system operating behavior and minimizing the cost function achieves the optimal/desired system operation).

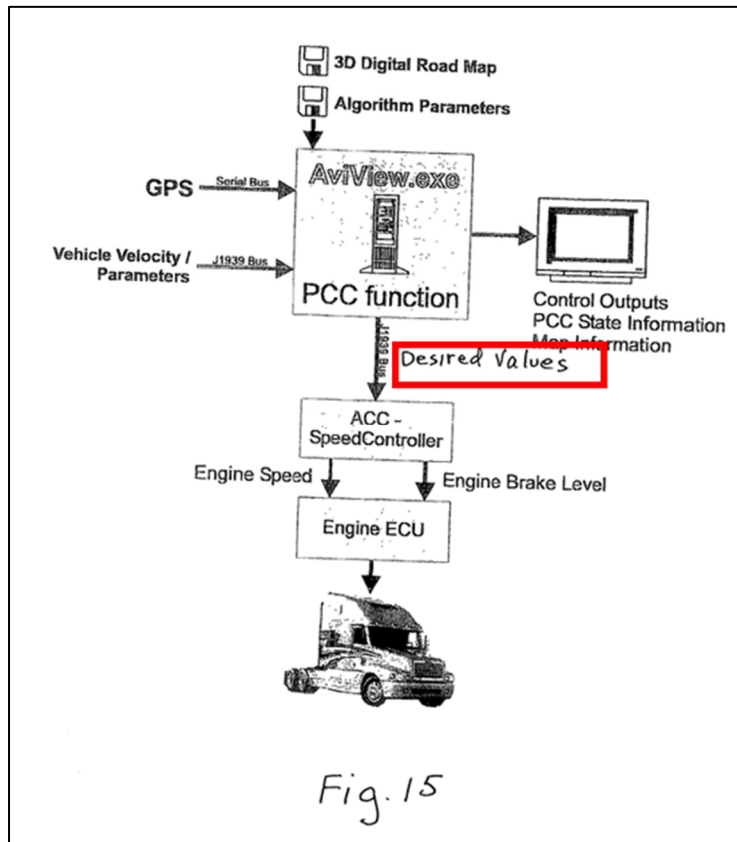
Neiss’s predictive cruise control system identifies “vehicle control parameters,” such as speed and throttle position, that “minimize[] vehicle operating costs,” *e.g.*, optimize vehicle operation, “based on an analysis of vehicle operating parameters and route information regarding a route section ahead 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 ...”), Claims 1 and 10 (“iteratively calculating...vehicle control parameters that minimize said vehicle operating cost function” that includes “vehicle fuel consumption as a function of vehicle speed, road gradient, and throttle position”), ¶¶[0014], [0004]; TESLA-1003, ¶103.

Moreover, Neiss describes controlling the vehicle according to the vehicle parameters determined by optimizing the vehicle operating cost function. TESLA-1003, ¶104. 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, ¶[0042]; TESLA-1003, ¶104. 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]; TESLA-1003, ¶105. As shown 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 vehicle’s engine. *Id.*



TESLA-1005, FIG. 4 (annotated)

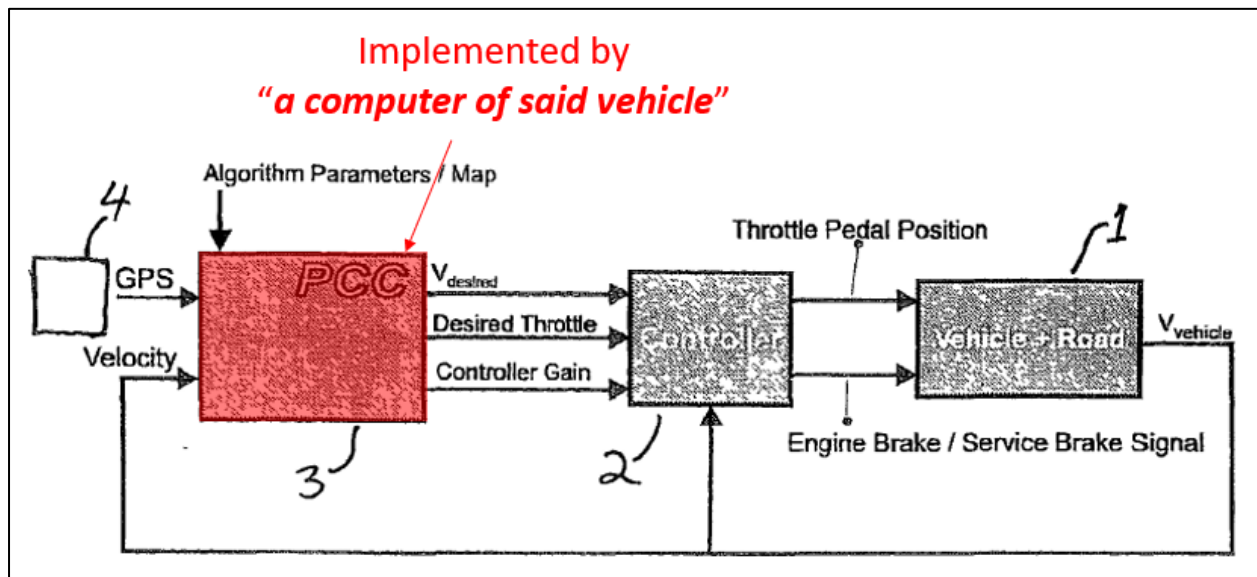


TESLA-1005, FIG. 15 (annotated)

Thus, the computation of optimal vehicle control parameters (including vehicle velocity) that allows 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. TESLA-1003, ¶106.

[1a]

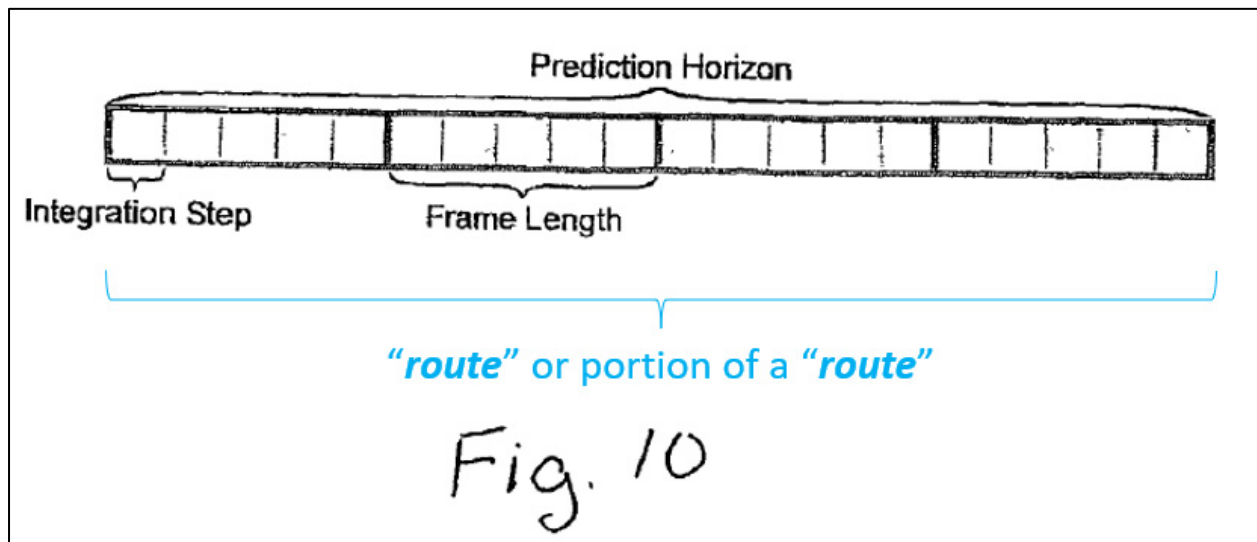
Neiss teaches a *computer of said vehicle*. TESLA-1003, ¶¶107-122. Specifically, Neiss discloses an onboard computer that includes memory and a data processor. *See, e.g.*, TESLA-1005, Abstract (“an onboard computer iteratively calculates and stores in a memory vehicle control parameters that optimize 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); TESLA-1003, ¶¶108-109 (explaining that Neiss’s PCC block is part of the onboard computer given both are described as performing/executing the optimization algorithm).



TESLA-1005, FIG. 4 (annotated)

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 ¶¶[0001] ("a cruise control unit...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; TESLA-1003, ¶110.

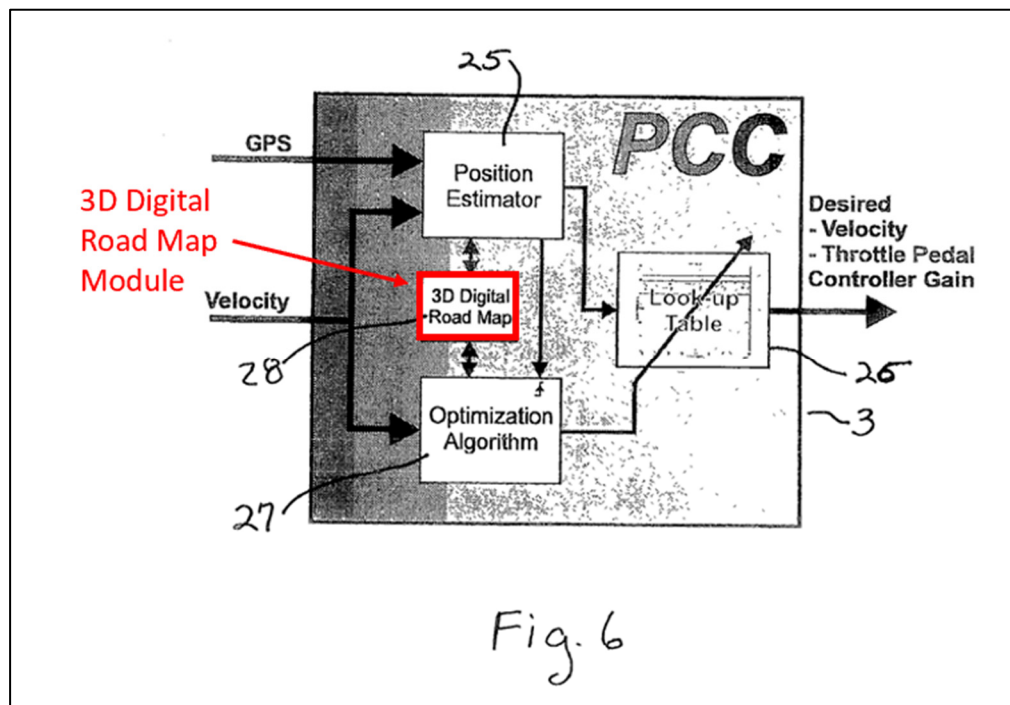


TESLA-1005, FIG. 10 (annotated)

The Prediction Horizon (shown above in blue; *see* FIG. 10, *supra*) corresponds to *a route*, *e.g.*, because it constitutes a path being traveled by the vehicle. TESLA-1003, ¶111. 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 (ahead of the vehicle as the vehicle travels the route to its final destination). TESLA-1005, Abstract, ¶¶[0001], [0015]. Therefore, in Neiss-Ran, a POSITA would have understood that a vehicle travels an overall *route*, with optimizations performed on each iteratively updated prediction horizon (predetermined distance) ahead of the vehicle as it travels this route. *See id.*; TESLA-1003, ¶111.

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). TESLA-1003, ¶112.

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)

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... 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”); TESLA-1003, ¶¶113-114.

Thus, the current vehicle position 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 therefrom about this identified prediction horizon (e.g., “gradient angles and curve radii for the upcoming road”). *See id.*; TESLA-1003, ¶115. 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, ¶[0162].

Neiss's 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.

From these disclosures, a POSITA would have understood that the onboard computer (which implements Neiss's PCC block) *receiv[es] a 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 prediction 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. TESLA-1003, ¶¶108-118.

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. TESLA-1003, ¶¶119-122. 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. *Id.* A POSITA would have found it obvious to modify Neiss-Ran so that its onboard computer receives the entire end-to-end route all at once, instead of receiving it iteratively in the manner described by Neiss. *Id.*

A POSITA would have been motivated to modify Neiss-Ran in this manner for the following reasons. **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. TESLA-1003, ¶120. **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 (TESLA-1003, ¶121 (citing TESLA-1011, 1:42-43)). 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.,

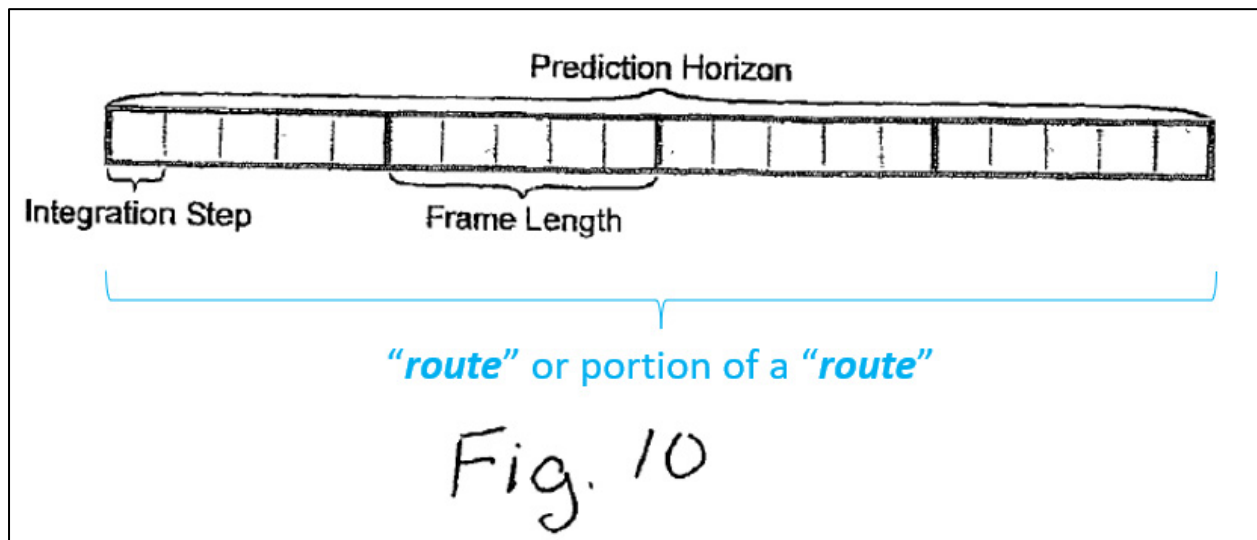
an onboard route computer for predictive cruise control that receives an entire route with respect to which optimizations and/or analytic operations can be performed). *See KSR*, 550 U.S. at 415-421; MPEP 2143; TESLA-1003, ¶121.

A POSITA would have also had a reasonable expectation of success in configuring Neiss-Ran in this manner. TESLA-1003, ¶122 (citing TESLA-1027, 7:52-54). That's because (1) it was well-known and conventional for vehicle onboard computer to receive routes from map applications (*id.*), (2) Neiss already includes a digital map that stores information about routes that the vehicle drives (TESLA-1005, ¶¶[0049], [0031], [0015], [0162], Abstract; TESLA-1003, ¶122), and (3) Neiss's optimization algorithm already obtains/receives information about route the 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). *Id.*

[1b]

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. Thus, in view of the specification, the present limitation would be understood as encompassing segmenting a route into a plurality of segments, where each segment has an endpoint/destination. *See id.*; TESLA-1003, ¶¶123-124.

Similarly, Neiss discloses segmenting the route into segments, each segment having an endpoint or destination. TESLA-1003, ¶¶123-130. Specifically, 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.



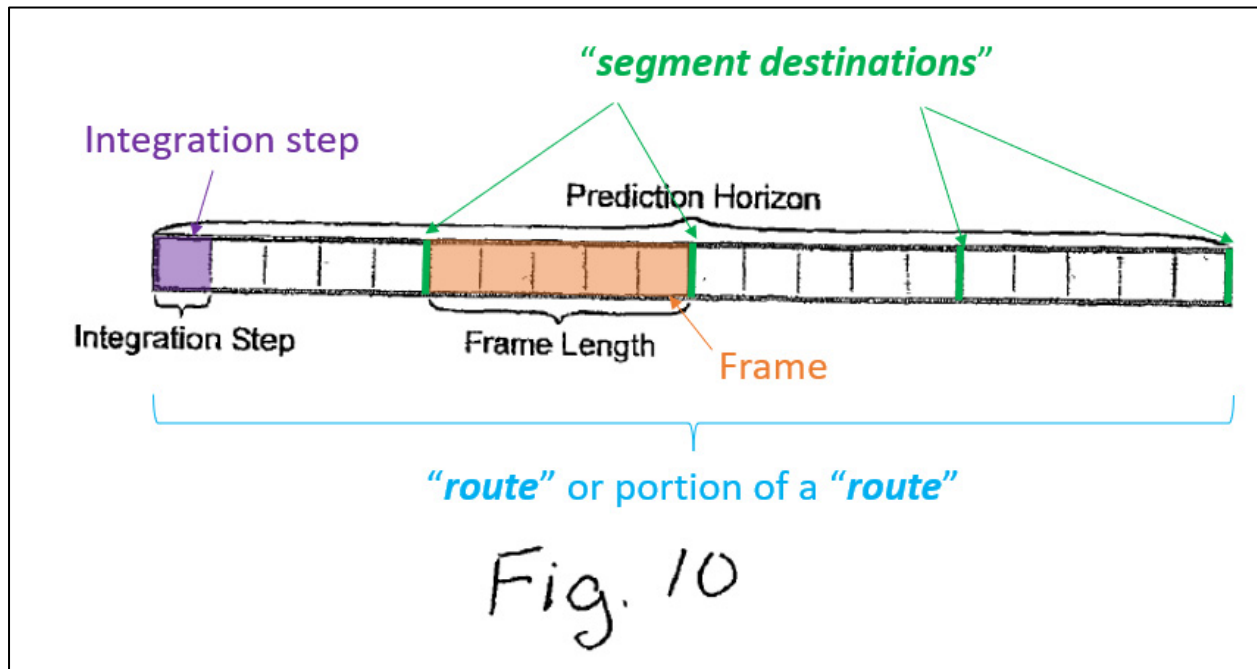
TESLA-1005, FIG. 10 (annotated)

As described for [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* §III(A)(4)[1a] *supra*. 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].

Also, as explained for [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. TESLA-1003, ¶¶107-118, 126-127 (citing TESLA-1005, ¶[0047]) (explaining that, 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).

Each Prediction Horizon (i.e., the **route** or a subset thereof) “is separated into Frames.” TESLA-1005, ¶[0170]; *see id.*, ¶[0192], FIG. 10, *infra*; TESLA-1003, ¶128. As shown below, each frame (one frame highlighted in orange below) is “a specified distance,” i.e., a portion (segment) of the route. TESLA-1005, ¶[0169]. Each frame 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]; TESLA-1003, ¶128. Each frame’s endpoint

corresponds to a *segment destination* (green) because it is the end (destination) of a frame/segment. TESLA-1003, ¶¶128-129.

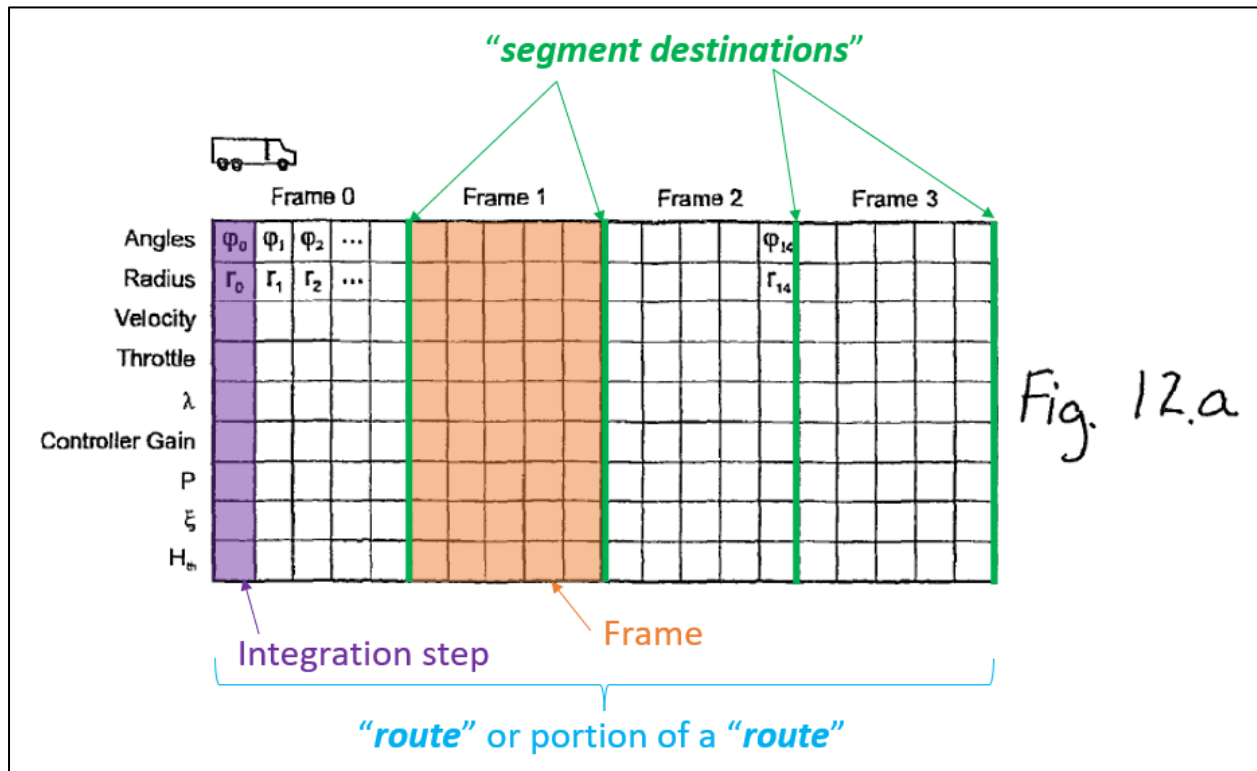


TESLA-1005, FIG. 10 (annotated)

Each Prediction Horizon ahead of the vehicle (*route*) is thus *segment[ed]* ... *into a plurality of segment destinations* at least because each Prediction Horizon is composed of a plurality of frames (*segments*), each having a respective endpoint (*segment destinations*). See TESLA-1003, ¶¶129 (citing TESLA-1005, ¶¶[0169]-[0170], [0192]).

In particular, Neiss-Ran's optimization algorithm is implemented using a lookup table "having a capacity that corresponds to the length (in frames and integration steps) of the Prediction Horizon." TESLA-1005, ¶[0171]; see *id.*, Claim

18, FIGS. 12a(*infra*)-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 the route*. TESLA-1003, ¶130.



TESLA-1005, FIG. 12a (annotated)

[1c)]

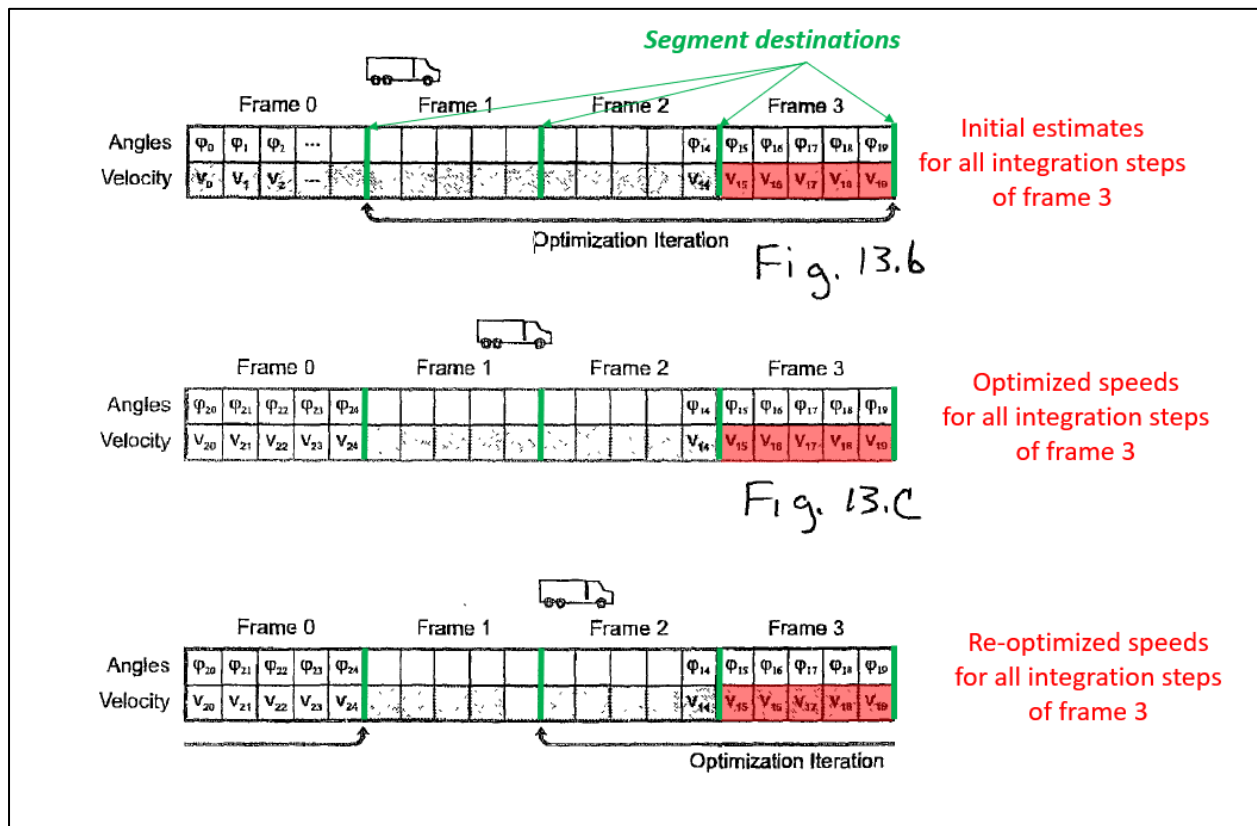
As explained below, Neiss-Ran's optimization algorithm is performed by the PCC block of the onboard computer (i.e., *said computer device*) for each frame in the route ahead of the vehicle (i.e., *for each* segment and its corresponding *segment destination*). TESLA-1003, ¶¶131-137; see analysis for [1a] *supra* (explaining that PCC block is implemented on the vehicle onboard computer).

In particular, the optimization algorithm is iteratively performed for frames 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’...”). 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]; TESLA-1003, ¶¶133.

Moreover, as explained above, Neiss’s optimization algorithm is implemented using a lookup 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 said integration interval.” TESLA-1005, Claim 18; *see also id.*, FIGS. 12a-c, FIGS. 13a-d. Additionally, each frame in the lookup table

corresponds to a route segment and has an endpoint that likewise correspond to the *segment destinations* (as explained for [1b]). TESLA-1003, ¶134.

As discussed further below for limitations [1d1]-[1f], *infra*, Neiss-Ran'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]; *see* §§III(A)(3), (4)[1d1]-[1f] *supra*, TESLA-1003, ¶¶84-99, 138-183. Once the vehicle reaches the end of a frame, "a new calculation is triggered" for upcoming frames along the route. *Id.*, ¶[0188]. As illustrated in FIG. 13 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 destination*). TESLA-1003, ¶135.



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

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). TESLA-1003, ¶136. And because each frame has an end point (*segment destination*)—as explained for [1b]—the optimization algorithm is performed by its *computer device for each of said segment destinations* along the vehicle's travelled route. TESLA-1003, ¶137. Additional details and analysis for how Neiss-Ran's

computer device performs the claimed operations recited in limitations [1d1] to [1g] for each segment destination are described below with reference to those limitations. TESLA-1003, ¶¶138-190.

[1d1]

As explained for [1pre], *minimizing power consumption* encompasses minimizing fuel consumption. TESLA-1003, ¶¶100-106, 138-139; *see* TESLA-1001, 6:14-30 (relating fuel efficiency to power consumption in the context of fuel-powered vehicles).

Neiss-Ran teaches *calculating an optimized speed by minimizing power consumption* (e.g., minimizing fuel consumption) *over said segment destination*. TESLA-1003, ¶¶140-149.

As described below, Neiss-Ran's Optimization Algorithm Module of the PCC block (implemented by 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 the vehicle's fuel consumption. *See, e.g.*, TESLA-1005, ¶¶[0179]-[0181], [0048], [0081]-[0082], [0092]-[0098], FIGS. 12a-c; TESLA-1003, ¶¶140-141.

As explained above for [1a] and [1b], the Prediction Horizon is divided/segmented into multiple Frames, each including multiple Integration Steps.

See §III(A)(4)[1a]-[1b] *supra*. The 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 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. TESLA-1003, ¶142.

Also, as discussed for [1b], the PCC block is implemented as part of Neiss’s “onboard computer” and “iteratively calculate[s] and stor[es]” 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, calculation of the *optimized speed* is performed at the *computer device* of Neiss’s vehicle. TESLA-1003, ¶143.

Moreover, Neiss’s “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] (in Neiss’s Predictive Cruise Control (PCC) system “[f]orward-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] (“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””), [0081]-[0082] (describing that one objective of the cost function is that the “vehicle should consume less fuel”), [0014] (PCC system “utilizes information about the current vehicle position, as well as upcoming terrain in order to save fuel”); TESLA-1003, ¶144.

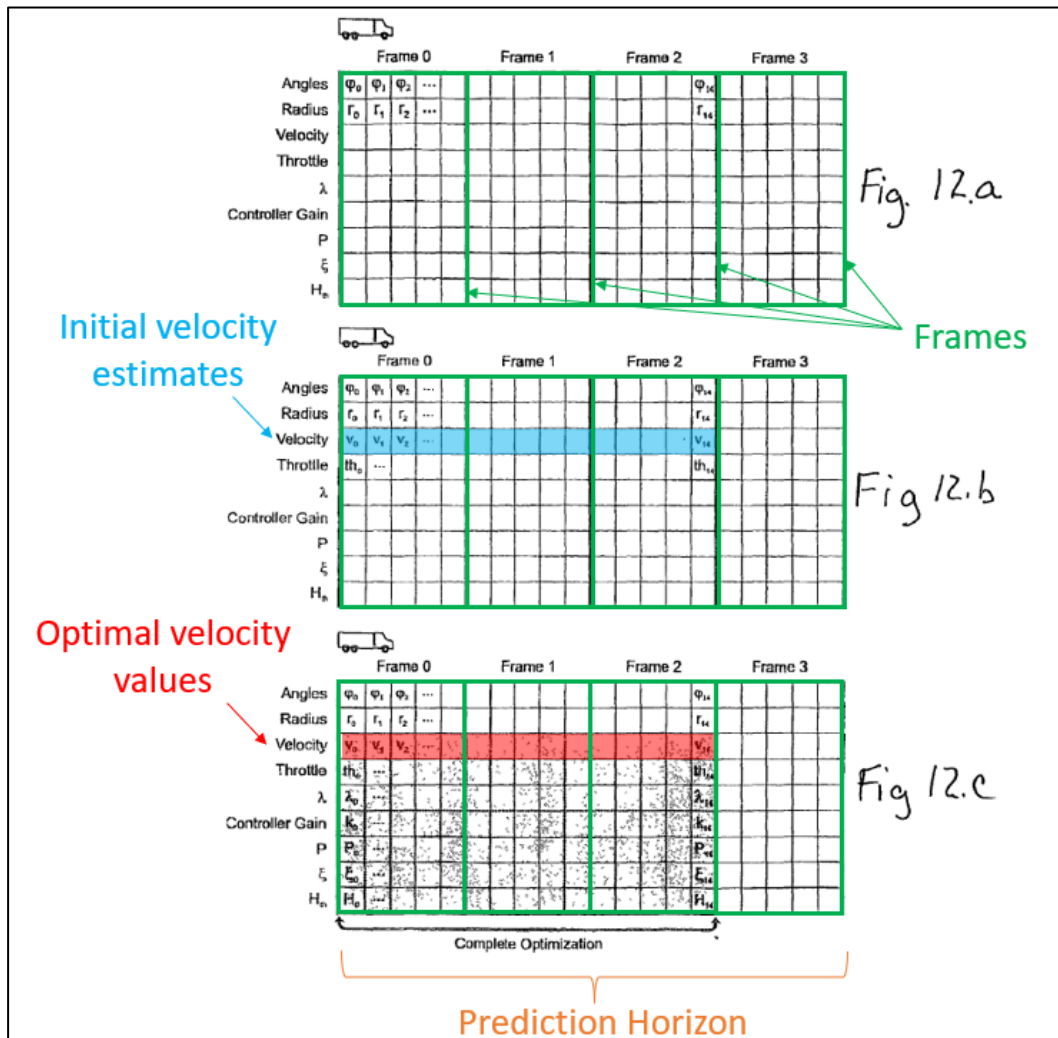
Neiss provides the following cost function with the objective of minimizing fuel consumption (represented as J_{fuel}):

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

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 kth integration step”); TESLA-1003, ¶¶145-147, 51-55, 159-160; *see also* TESLA-1005, ¶¶[0126]-[0128] (including above function as part of overall system cost function being minimized), [0180], [0149]-[0160] (describing algorithmic steps executed in each iteration), [0093]-[0098] (using maximum fuel as parameter for fuel cost function optimization).

The optimization algorithm thus iteratively calculates/outputs for each frame, optimized “velocity values” (which are then stored in a look-up table, as depicted in

FIGS. 12a-c *infra*). TESLA-1005, ¶[0181], FIGS. 12a-c; *see id.*, ¶[0186]; TESLA-1003, ¶¶148, 51-55, 58-64.



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

In this manner, Neiss's PCC block implemented by its onboard computer (i.e., *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 power consumption*) for an upcoming frame with an

endpoint/destination (i.e., *over said segment destination*). TESLA-1003, ¶149. Additionally, the above-described calculation uses as inputs one or more pieces of information, as described for [1d2] below. TESLA-1003, ¶¶150-162.

[1d2]

This limitation identifies four types of information that limit the *calculating* in [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 listing, this limitation is satisfied by showing a single type of information recited herein. TESLA-1003, ¶¶150-151. 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 information), a POSITA would have found that to be obvious in view of Neiss-Ran. *Id.*

As described below, Neiss-Ran discloses or renders obvious calculation of optimal “velocity values” (*optimized speed*) that uses as *input* the four types of information recited in this limitation. TESLA-1003, ¶152; *see, e.g.*, TESLA-1005, ¶¶[0049], [0190], [0031], [0092]-[0098].

First, Neiss-Ran discloses using *information about the external environment of the vehicle* as an *input* to calculating the optimal “velocity values” (*calculating the optimized speed*). TESLA-1003, ¶153. As explained for [1d1], and

consistent with the '426 Patent's specification, Neiss-Ran'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 (calculated *optimized speed*). TESLA-1005, ¶¶[0049], Abstract, [0035], [0046]-[0049], [0167]-[0169], claims 1-2 (optimization of the vehicle operating cost function is performed as "a function of at least ... vehicle position and route parameters," where "said route parameters include at least road gradient and radius of curvature"); TESLA-1003, ¶¶153, 138-149.

Second, Neiss-Ran also discloses using *information about the operational status of the vehicle as input* (e.g., current velocity) to calculating the optimal "velocity values" (i.e., *calculating the optimized speed*). TESLA-1003, ¶154; TESLA-1001, 3:56-58 (explaining "vehicle's current speed" is an example of "information about the operational status of the vehicle"). 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]; TESLA-1003, ¶¶154, 138-149.

Third, Neiss-Ran renders obvious using *information about one or more user command inputs as input* to calculating the optimal “velocity values” (i.e., *calculating the optimized speed*). *Id.*, ¶155-158. 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-1003, ¶155; 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 “determine[s] ... a desired speed inside the speed band...to maximize fuel savings.” *Id.*; TESLA-1003, ¶155, 138-149.

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*). TESLA-1003, ¶¶156, 138-149. A POSITA would have been motivated to provide a user the ability to input these values and thereby enable the calculation of optimal vehicle control parameters to be finetuned/customized to reflect the user’s preferences and/or desired driving speeds. *Id.* (citing TESLA-1016, Abstract; TESLA-1017, [0007]). 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 for calculating optimal vehicle control parameters). *See KSR*, 550 U.S. at 415-21; TESLA-1003, ¶157. Moreover, a POSITA would have had a reasonable expectation of success in so implementing the Neiss-Ran system. TESLA-1003, ¶158. That’s because (1) cruise controllers 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) within Neiss-Ran’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. *Id.* (citing TESLA-1016, ¶[0016]; TESLA-1017, Claim 1).

Fourth, Neiss-Ran’s optimization algorithm uses *one or more operational parameters of the vehicle as input*, in calculating the optimal velocity (*calculating the optimized speed*). TESLA-1003, ¶159. Consistent with the ’426 specification, Neiss discloses using maximum fuel volume (referred to as *fuel_{max}*) as an operational parameter/input that is used to calculate the optimal “velocity values.” *Id.*; TESLA-1001, 25:33-34.

For example, Neiss discloses the use of $fuel_{max}$ 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 $fuel_{max}$).

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

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

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

Given these disclosures, 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***). TESLA-1003, ¶¶159-160.

Further, as described above for [1a] and [1d1]-[1d2], Neiss’s “onboard computer” (within which the PCC block is implemented) performs 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]; TESLA-1003, ¶161. This 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, 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 calculating⁴...by one or more processors*). TESLA-1003, ¶161 (noting the well-known use of processors in computers to execute algorithms); *see* §III(A)(4)[1d1] *supra*.

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*

⁴ [1d1]’s recitation of “said calculating” antecedently refers to the immediately-preceding “calculating” limitation ([1d1]).

processors) as part of such optimization calculations. TESLA-1003, ¶162 (explaining that an algorithm would have to process and analyze received data before it can perform calculations with respect to such information).

[1e]

The '426 specification does not use the phrase “probable speed” and this limitation does not define this phrase or recite any requirements for it. *See* TESLA-1001. The plain meaning of “probable speed” is a speed that is probable or likely. TESLA-1003, ¶164; TESLA-1025; TESLA-1026.

Neiss-Ran renders obvious calculating a probable speed. TESLA-1003, ¶163-170. As described above (*see* §III(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* TESLA-1006, 11:23-42 (describing that historical and real-time data 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; TESLA-1003, ¶¶84-99. 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, 19:19-22; *see id.*, 18:53-56 (providing four examples of statistical models), FIG. 4A; *see also* 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); TESLA-1003, ¶¶197-209. One example of such a traffic prediction model in Ran is a Greendshields model, expressed in below excerpt, which provides a “fundamental speed-density relationship” where speed is a function of traffic density. *See id.*, 11:23-42.

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

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)

This 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.*; TESLA-1003, ¶¶165-167.

Thus, as explained above and as discussed in §III(A)(3) *supra*, Neiss-Ran would leverage 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. TESLA-1003, ¶¶84-99. Because the predicted speed for the segment is

determined based at least on such real-time traffic conditions at the upcoming route segments, this speed is a *probable speed* that the vehicle is likely or expected to travel at the respective road segment (i.e., the frame in the context of the Neiss-Ran combination). TESLA-1003, ¶¶168, 84-99.

Moreover, as described above in §III(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*). TESLA-1003, ¶¶169, 84-99. 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 traffic prediction models and the associated data used by those models). *Id.*; see §III(A)(3) *supra*.

Therefore, Neiss-Ran's onboard computer (*said computer device*) calculates a predicted speed based on real-time traffic conditions (*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*). TESLA-1003, ¶170.

[1f]

As described below, Neiss-Ran teaches this limitation because it discloses calculating a vehicle speed optimal speed for an upcoming segment (*optimized*

speed), and improving this calculated speed iteratively based on the real-time predicted speed (i.e., *probable speed*) to thereby obtain/determine an *efficient speed*. See, e.g., TESLA-1005, ¶¶[0188]-[0191], [0010], [0163], [0167]-[0169], TESLA-1006, 11:23-42; TESLA-1003, ¶¶171-172.

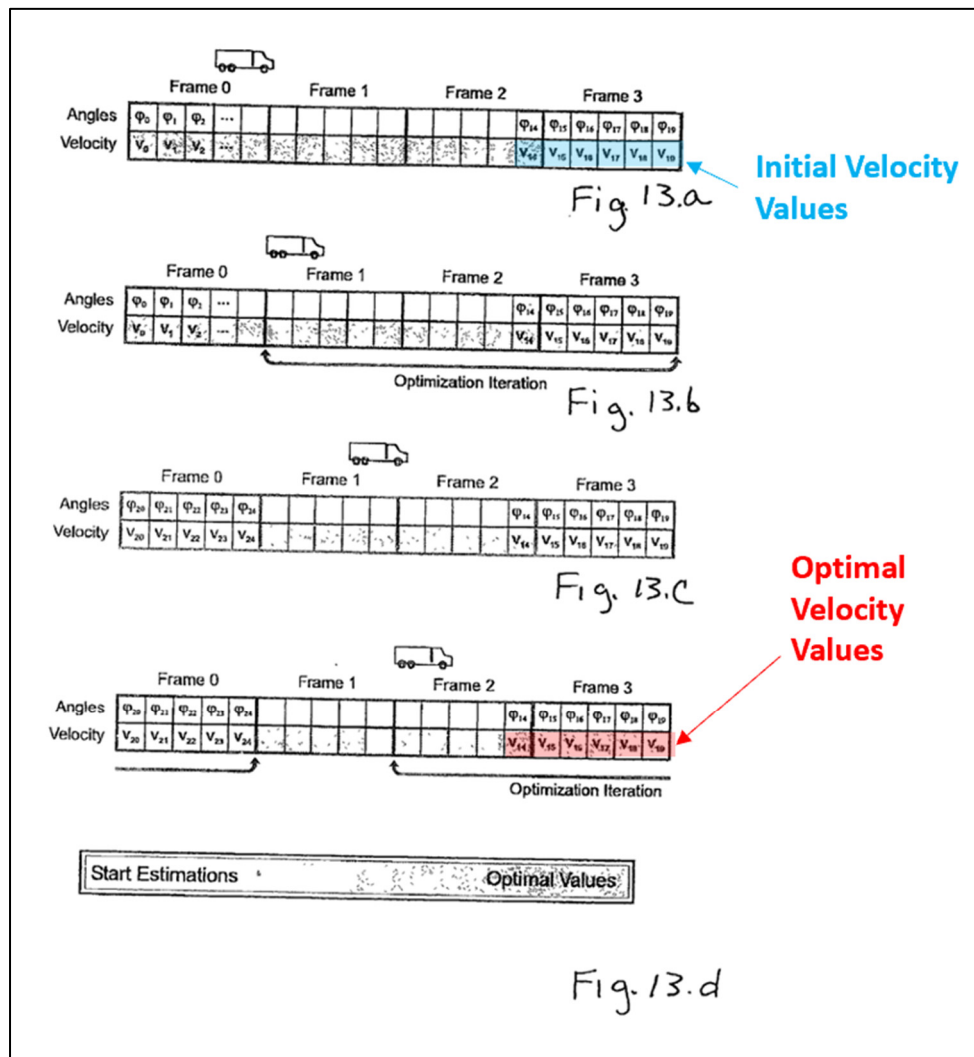
As described above, Neiss discloses the calculation of the *optimized speed* for an upcoming road segment/frame. See §III(A)(4)[1d1] *supra*. Also, as described above, Neiss in view of Ran renders obvious the calculation of a predicted speed for a route segment/frame based on real-time traffic conditions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

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, e.g., 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/segments. *Id.*, ¶[0190]; TESLA-1003, ¶¶171-174, 84-99.

However, as explained in §III(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]; TESLA-1003, ¶¶175, 84-99. As described in §III(A)(3) *supra*, to further increase the accuracy and efficiency of Neiss’s algorithm, 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 (as Ran discloses). *Id.*

Thus, Neiss-Ran performs a first optimization iteration that yield optimized vehicle control parameters (including optimal velocities) for one or more upcoming frames. *See* TESLA-1005, ¶¶[0182]-[0187]; TESLA-1003, ¶176. 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-1003, ¶176.



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

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; *see* TESLA-1003, ¶177) 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.” *Id.*; TESLA-1005, ¶¶[0012], [0163], [0167], [0190].

Neiss-Ran’s system thus accounts for real-time conditions and uses a more accurate, traffic-informed predicted speed to refine the optimized velocities in the lookup table (for minimizing power consumption of the vehicle, *see* §III(A)(4)[1d1] *supra*) and thereby obtains the updated optimal values (including optimal velocity values) for the frame/segment. *See id.*, ¶¶[0190], [0156]-[0159], [0167]-[0169], [0190], [0010], Abstract, [0015]; TESLA-1003, ¶¶177-178, 84-99.

Because the 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”), Abstract, [0015] (“as the optimization iterations are performed, the “optimal vehicle control parameters are continuously updated ... as the vehicle... moves along”), FIGS. 13a-d *supra*; TESLA-1003, ¶178. 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”); TESLA-1003, ¶178.

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. *Id.*

Moreover, as explained above for [1a] and [1d1], Neiss’s onboard computer that implements the PCC block (*said computer device*) performs the optimization algorithm. TESLA-1003, ¶179. Because the PCC block is also responsible for the calculations performed in each subsequent iteration of the algorithm, 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. *Id.*

Additionally, or alternatively, if it is determined that Neiss-Ran does not explicitly teach how to “adapt[] the look-up table during the drive” by taking the predicted real-time traffic-based 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 data fusion techniques that combine two speed values. TESLA-1005, ¶[0163]; TESLA-1003, ¶¶180-182. 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; *see id.*, 18:18-49, FIG. 3 (describing fusion of different speed information, using models including weighted average, fuzzy logic).

Thus, given Neiss’s express 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 [1d1]) and *probable speed* value (as described for [1e]) to obtain a combined/adapted speed, which corresponds to the *efficient speed*. TESLA-1003, ¶¶180-182; *see* TESLA-1005, ¶¶[0008], [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). 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) 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, per Neiss and Ran). *See KSR*, 550 U.S. at 415-21; TESLA-1003, ¶¶180-182.

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 speeds to achieve a more realistic, optimal speed value that accounts for real-time traffic conditions (*efficient speed*). TESLA-1003, ¶183.

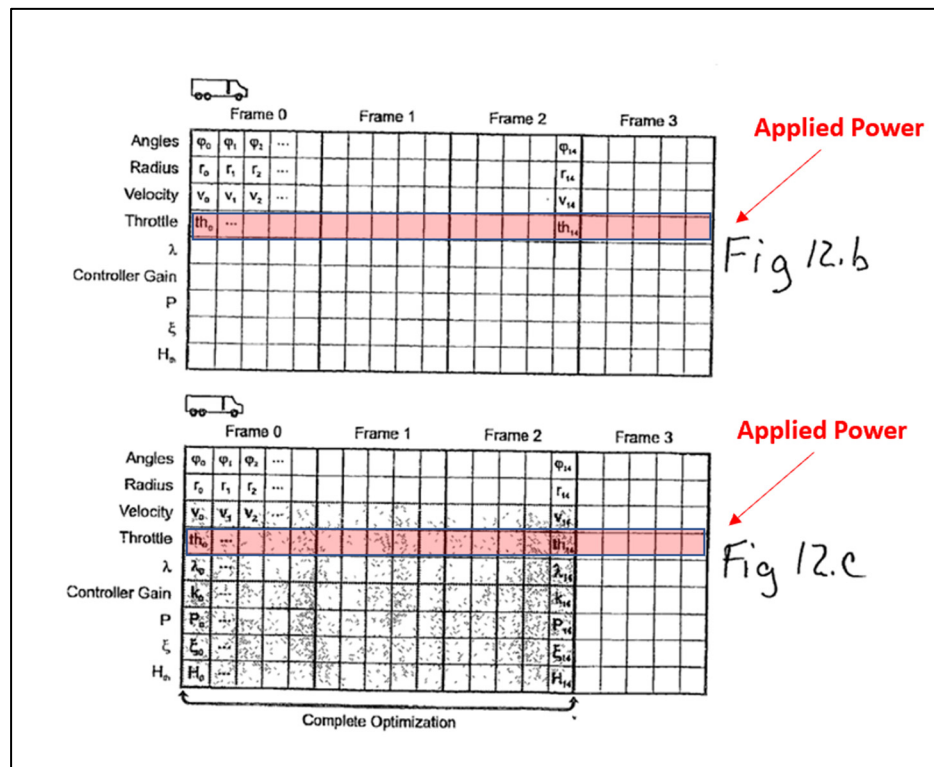
[1g]

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). TESLA-1003, ¶¶184-185.

As described above for [1d1]-[1d2], [1e], and [1f], Neiss-Ran'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], FIGS. 13a-13d, 12a-12d; *see* §III(A)(4)[1d1]-[1f] *supra*.

These values are stored in Neiss's look-up table (which is "contained in a ring buffer memory"). TESLA-1005, ¶[0162]. As the vehicle traverses the route, Neiss's system "reads out" from memory the optimal velocity values and throttle pedal positions corresponding to the current vehicle position. *Id.*; *see id.*, ¶[0004]. Thus, for each *efficient speed* calculated/stored by Neiss-Ran's system, corresponding throttle pedal position and controller gain series values are also calculated. TESLA-1003, ¶¶184-186; *see* TESLA-1005, FIGS. 12b-c *infra*.



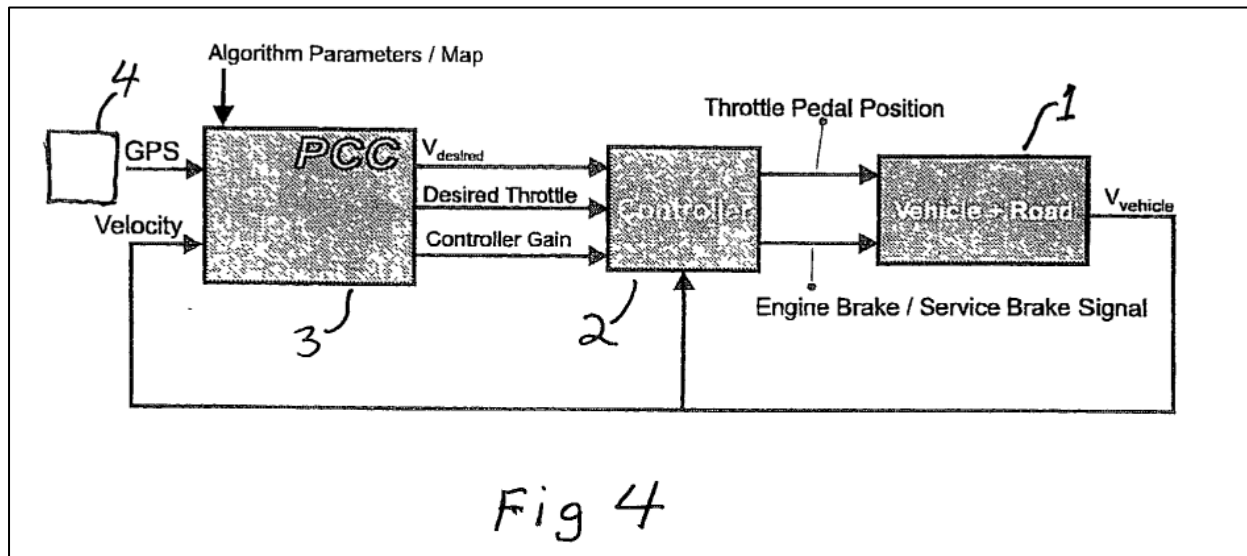
TESLA-1005, FIGS. 12b-c (annotated)

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*. TESLA-1003, ¶187. That's 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. *Id.*; TESLA-1005, ¶¶[0034] (providing a vehicle controller and teaching that the throttle of the vehicle 1 is controlled based on an error signal

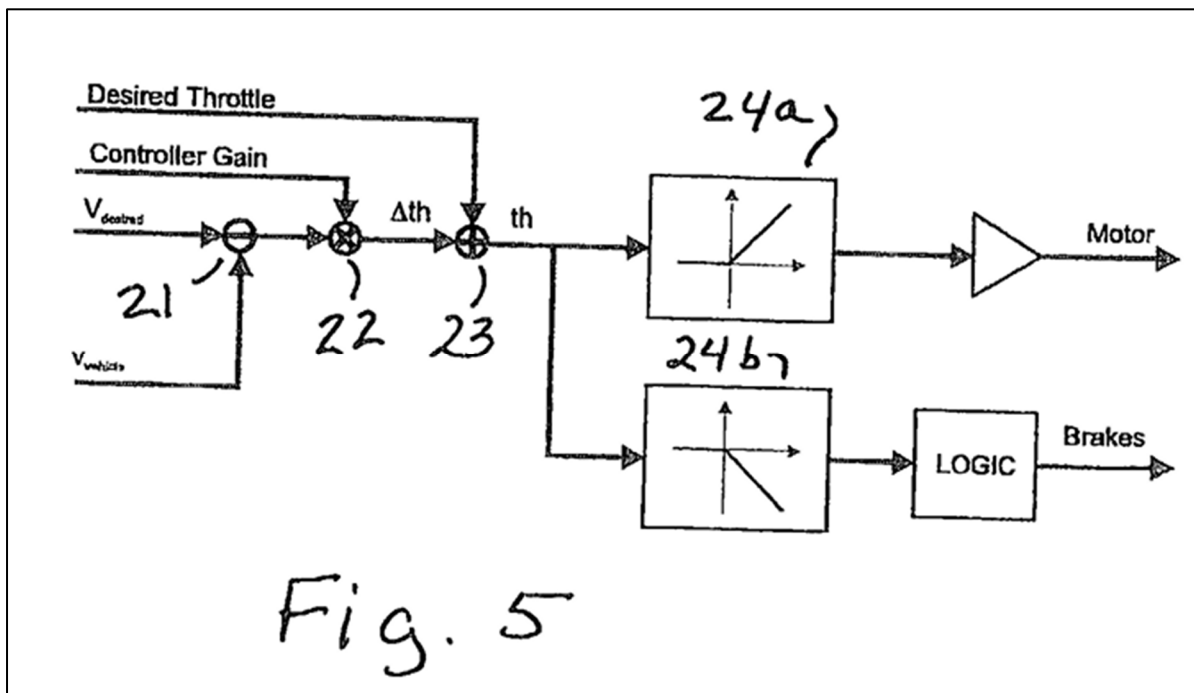
between actual and desired vehicle speed”), [0039] (throttle value represents “Throttle Pedal Position” that “controls...fuel injection to the motor”), [0042] (“amount of fuel injected can be mapped to a pedal or throttle position”); TESLA-1003, ¶187.

Second, Neiss-Ran teaches *applying the calculated power ... to the vehicle engine to travel said segment destination*. TESLA-1003, ¶¶188-189. 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.*; TESLA-1003, ¶¶188-189. 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.” TESLA-1005, ¶[0042]. 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*) and [0043] (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”);
TESLA-1003, ¶¶188-189; *see* TESLA-1001, 19:15-17.



TESLA-1005, FIG. 4



TESLA-1005, FIG. 5

Additionally, the *calculated power* is applied to cause the vehicle *to travel said segment destination*. TESLA-1003, ¶190. As explained above, *efficient speed/s* are calculated by the optimization algorithm for each Integration Step in each Frame. See §III(A)(4)[1f] *supra*. Applying throttle positions that correspond to each *efficient speed* thus causes applying of the *calculated power* for each Integration Step in each Frame, which causes the vehicle to travel through the Frame until it reaches the Frame's endpoint (*segment destination*). TESLA-1003, ¶190; TESLA-1005, Claim 1, ¶¶[0162], [0188]-[0190], [0192].

(b) Claim 2

As described for Claim 1, Neiss-Ran's system iteratively and continuously 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], [0012], [0165]-[0167], [0190]-[0192], [0009]-[0010], claim 1, Abstract. 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*. Because the optimization algorithm operates iteratively and computes optimal vehicle control parameters as the vehicle drives along a route (i.e., for each additional portions 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,” e.g., when a final destination has been reached. TESLA-1003, ¶191. Alternatively, a POSITA would have found obvious that the calculation would continue *until a final destination is reached*, e.g., to allow reaping benefits of continued optimizations throughout the entire trip/route, accounting for changes in driving/operating conditions as the vehicle travels the route, and achieving a predictably operating system that predictably extends Neiss’s existing techniques to the entire route. TESLA-1003, ¶¶191-192 (citing TESLA-1005, ¶¶[0165], [0167], [0012]).

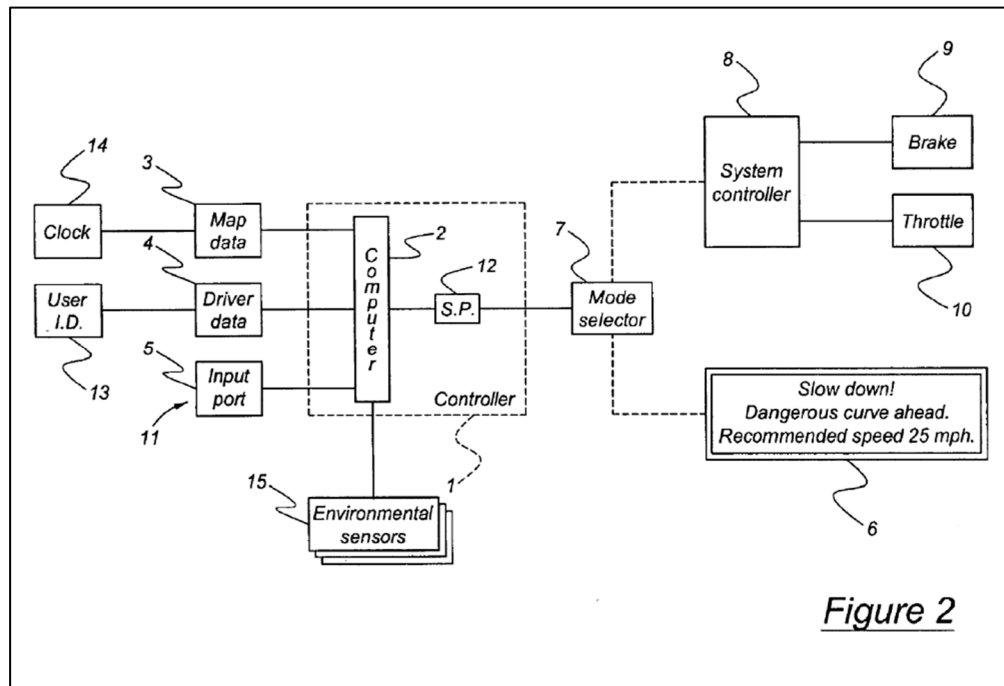
**B. [GROUND 2] – Neiss-Ran-Andersson Render Obvious
Claim 3**

1. Andersson

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, TESLA-1003, ¶¶193-196.

Andersson’s system 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 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” that is used to “predict[] a driver[']s intended speed on a road section never driven before.” *Id.*, 2:4-15, 4:53-60.



TESLA-1007, FIG. 2

As shown 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. This 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.

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.*, how to determine speeds at which a vehicle should travel based, *e.g.*, on road or operating conditions). TESLA-1003, ¶196.

2. Combination of Neiss, Ran, and Andersson

As described in §§III(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 refined in subsequent iterations, where each iteration adapts the optimized speeds using the real-time traffic based predicted speeds in the upcoming frames (*probable speed*). 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; *see* §III(A)(2)-(3) *supra*.

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* §III(B)(1) *supra*; TESLA-1007, 3:58-60.

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, per Andersson. TESLA-1003, ¶¶197-209. 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 (or another appropriate combining technique) of the following speed values—the historical driving speeds (which further include historical driving data for similar segments, as Anderson contemplates) and the speeds based on real-time traffic information (in view of Neiss-Ran, *see* §III(A)(4)[1f] *supra*). TESLA-1003, ¶¶197-199; 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 §III(A)(4), e.g., based on optimized speed and the predicted/probable speed (the latter being augmented in view of Anderson’s teachings, as described above).

A POSITA would have been motivated to implement Andersson’s teachings in combination with Neiss-Ran for the following reasons. TESLA-1003, ¶¶200-209.

First, a POSITA would have recognized the benefit of using historical data in predicting a vehicle’s speed at an upcoming route segment. TESLA-1003, ¶201;

TESLA-1018, ¶[0006]; TESLA-1006, 15:33-37. The use of historical data provides increased accuracy, as historical data reflects previous real-world data samples over a long time horizon. *See id.*

Indeed, 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” for each route segment), 18:18-49; TESLA-1005, Abstract (“The optimal vehicle control parameters...are... continuously updated ... as the vehicle moves along, thereby adjusting the ‘optimal’ control parameters to reflect actual vehicle historical operating experience”). 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 for a vehicle to travel a particular route. TESLA-1003, ¶¶199-202 (citing TESLA-1011, 6:65-7:34; TESLA-1019, ¶[0006]).

Second, 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. TESLA-1003, ¶203 (citing TESLA-1005, ¶[0007]). A POSITA would have therefore been motivated to seek out solutions, such as 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. TESLA-1003, ¶203.

Third, before the Critical Date, it was well known to use historical driving data for similar road segments to predict speed for a current road segment, as Andersson expressly teaches. *See, e.g.*, TESLA-1011, 6:65-7:34; TESLA-1012, ¶[0028]; TESLA-1003, ¶204. 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, per Anderson) 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). *See KSR*, 550 U.S. at 415-21; MPEP 2143; TESLA-1003, ¶205.

A POSITA would have had a reasonable expectation of success in pursuing this combination. Given that (1) Ran discloses traffic prediction models that utilize real-time traffic information and historical driving speed information to predict driving speeds for route segments (TESLA-1006, 17:7-16) and (2) Andersson also discloses prediction of driving speeds using historical data (including data for similar

road segments), modifying Neiss-Ran to additionally account for historical driving data for similar road segments would have been straightforward and within a POSITA's skillset. TESLA-1003, ¶206. Ran discloses, e.g., modeling and weight averaging techniques for combining different types of speed data and Andersson similarly provides computing techniques for computing such speed predictions. TESLA-1003, ¶¶199, 207; TESLA-1006, 18:18-31, 7:17-27; TESLA-1007, 4:47-60; *see also* TESLA-1001, 19:3-11 (also using weighted average to combine speeds).

Thus, 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 use such technique(s) to compute the predicted speed (*probable speed*) using real-time traffic information and historical driving speed data (including historical data for similar route segments). TESLA-1003, ¶¶207-209.

3. Claim 3 Analysis

Beyond the verbatim recitation of this limitation in the '426 specification, there is no explanation of 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." *See* TESLA-1001.

The only guidance 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, the claimed ***similar destination*** encompasses route segments that have endpoints (***destinations***) and have “similar” characteristics as the route segment for which an “efficient speed” is computed. TESLA-1003, ¶¶210-212.

As explained above in §III(B)(2), Neiss-Ran-Andersson leverages 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; TESLA-1003, ¶¶197-209. For example, Andersson teaches that it receives “vehicle position information via a position input connection” and extract “map information relating to ... a current and an upcoming road section,” and then generates a “speed plan” for these road sections. TESLA-1007, 1:46-57. To do this, 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.*; TESLA-1003, ¶213.

Given Andersson’s teaching of using historical driving data/speed for similar travel segments in predicting speed and Ran’s teaching of predicting speed using real-time traffic data and 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, per Andersson). TESLA-1006, 7:28-29; TESLA-1003, ¶214; *see* §III(B)(2) *supra*.

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. 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; TESLA-1003, ¶215. Thus, in view of Ran and Andersson and as explained above in §III(B)(2), Neiss-Ran-Andersson computes 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 Anderson contemplates) and the speeds based on real-time traffic information (per Neiss-Ran, *see* §III(A)(4)[1e] *supra*). TESLA-1003, ¶¶216, 84-99; TESLA-1006, 11:23-42; TESLA-1007, 4:47-60.

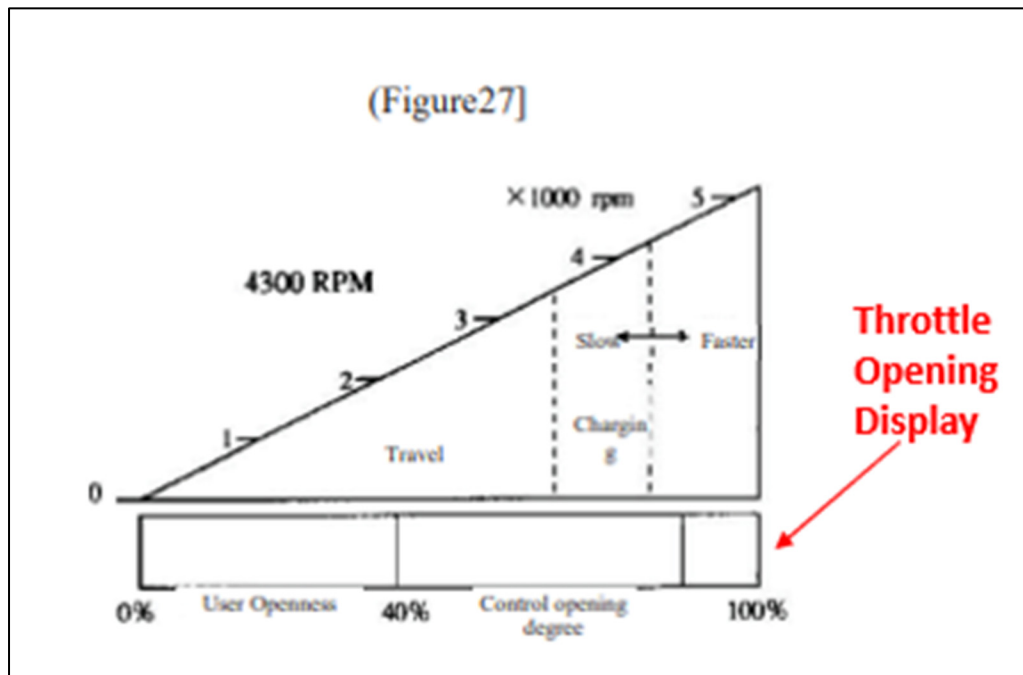
Moreover, because (1) determining *efficient speed* (as described for [1f]) is based at least on the *probable speed* and (2) as described above (and in §III(B)(2)), *probable speed* is further calculated using historical data for a similar route segment, the resulting Neiss-Ran-Andersson combination 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). TESLA-1003, ¶217.

C. [GROUND 3] – Neiss-Ran-Hosoda Render Obvious Claim 4

1. Hosoda

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]-[0005], [0028]-[0064] (providing examples of data that can be displayed). TESLA-1003, ¶¶218-219. Hosoda discloses a “[d]isplay control ECU 200” that “performs display control to the display 201,” which is “a liquid crystal display for displaying an instrument panel or 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].



TESLA-1009, FIG. 27 (annotated)

As shown above, in addition to the engine speed, the display provides the “ratio of each throttle opening” using “a message and 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

As described in §III(A)(3) and for [1g] *supra*, Neiss-Ran’s predictive cruise control system is configured to determine an *efficient speed* for an upcoming route segment, where a *calculated power* corresponding to the *efficient speed* is applied *to the vehicle engine to travel said segment destination*. See §III(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 §III(C)(1) *supra*.

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 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. TESLA-1003, ¶¶220-229 (citing TESLA-1013, ¶¶[0025]-[0029]; TESLA-1020, 4:14-28).

A POSITA would have been motivated to implement Hosoda’s teachings in combination with Neiss-Ran for the following reasons. TESLA-1003, ¶¶225-229.

First, a POSITA would have recognized the benefit of notifying a vehicle operator of the power applied to the vehicle, which as Hosoda explains, 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]; TESLA-1003, ¶226.

Second, a POSITA would have been motivated to pursue this implementation because doing so is merely the application of known techniques (*e.g.*, displaying operational parameters of 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 control parameters, including velocities and corresponding throttle values to control the vehicle on upcoming route segments/frames) to yield predictable results (*e.g.*, displaying outputs of these already-computed optimal control parameters at an in-vehicle display). *See KSR*, 550 U.S. at 415-21; MPEP 2143; TESLA-1003, ¶227.

As explained below, a POSITA would have had a reasonable expectation of success in implementing a display in the vehicle (as per Hosoda) and configuring the system to notify the vehicle operator, via such display, of the applied power (*e.g.*, the throttle pedal position output by the Neiss-Ran optimization algorithm). TESLA-1003, ¶228. 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. *Id.* And, before the Critical Date and as Hosoda expressly teaches, displays in vehicles were well-known and conventional means by which the vehicle notified operator of the vehicle's operations, including the control

parameters that are used to control the vehicle's engine. *Id.*; TESLA-1020, 4:14-28; TESLA-1013, ¶¶[0025]-[0029].

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 express teaching of displaying throttle-based control values on an in-vehicle display, it would have been straightforward and well-within the skillset of a POSITA to modify Neiss-Ran in view of Hosoda to implement an in-vehicle display and provide for display Neiss-Ran's already-computed and stored vehicle control parameter values, including the throttle-based power values being applied to the vehicle's engine. TESLA-1003, ¶¶228-229.

3. Claim 4 Analysis

As described above in §§III(C)(1)-(2), Neiss-Ran-Hosoda leverages Hosoda's teachings relating to displaying throttle-related values on a vehicle display. TESLA-1003, ¶¶218-232; TESLA-1009, ¶[0024]; *see id.*, ¶¶[0001]-[0005], [0023]-[0064]. 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].

Neiss-Ran's system already calculated and stored, in memory, throttle related information that corresponds to the calculated efficient speed (i.e., ***calculated applied power***). See §III(A)(4)[1f] *supra*. And, as described in §III(C)(2) *supra*, a POSITA would have found it obvious to display the already-calculated throttle related information at such an in-vehicle display. TESLA-1003, ¶¶231, 220-229. Therefore, Neiss-Ran-Hosoda teaches an in-vehicle display that displays the ***calculated applied power*** output by the Neiss-Ran optimization algorithm—thereby ***notifying the operator of the vehicle*** of the same. *Id.*, ¶232.

IV. PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION

A. 35 U.S.C. §325(d) – Discretionary Denial Under 325(d) is Not Warranted

Discretionary denial under the Board's §325(d) *Advanced Bionics* analysis is not warranted. See *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6, 8-10 (PTAB Feb. 13, 2020) (precedential).

None of the references advanced herein was previously before the Office. See *generally* TESLA-1002.

If it is argued that the same or substantially the same art or arguments were previously presented to the Office, the second prong of *Advanced Bionics* is not met. See TESLA-1003, ¶¶28-232.

B. The *Fintiv* Factors Favor Institution and No Denial Under §314(a)

Consistent with Congressional intent, *Fintiv*'s goals, and Director Vidal's interim guidance (TESLA-1023), Tesla asks the Board to consider the challenges raised herein. IPR2020-00019, Paper 11, at 6 (PTAB Mar. 20, 2020) ("*Fintiv*") (precedential). As explained below, a holistic analysis of the *Fintiv* factors favors institution.⁵

1. Factor 1: Institution Will Enable a Stay

Factor 1 is neutral because neither Tesla nor iQar have requested a stay in co-pending litigation (*iQar Inc. v. Tesla, Inc.*, 1-23-cv-00694, (W.D. Tex.)) ("WDTX Litigation"). Moreover, the Board does "not attempt to predict" how an appeals or district court might adjudicate a future motion to stay. *Fintiv*, 12; IPR2019-01393, Paper 24, at 7 (PTAB June 16, 2020) ("*Sand Revolution*") (informative); IPR2020-01019, Paper 12, at 14 (PTAB Dec. 1, 2020) ("*Sotera*") (precedential). Should Tesla

⁵ Apart from Tesla's showing that *Fintiv* favors institution, the *Fintiv* framework is legally invalid and should not be followed. The framework (1) exceeds the Director's authority, (2) is arbitrary and capricious, and (3) was adopted without notice-and-comment rulemaking.

file an IPR petition against another of the patents involved in the WDTX litigation, this Factor is subject to change to favor institution.

2. Factor 2: Uncertain District Court Schedule

Factor 2 favors institution because no trial date is set in the WDTX Litigation, and regardless, any scheduled trial date is speculative since “scheduled trial dates are unreliable and often change.” TESLA-1023, 8. As of the filing of this petition, WDTX Litigation is still conducting venue discovery and the parties have agreed to stay all merits-based fact discovery until transfer is decided. Further, the median time-to-trial for WDTX is currently 26 months,⁶ which suggests a likely trial date no sooner than March 2026, and the pending Motion to Transfer Venue (TESLA-1024), if granted, may extend the trial date further. In contrast, if this proceeding is instituted, the Board will likely issue its Final Written Decision around July 2025.

3. Factor 3: Early Stage of Parallel Proceedings

This petition is being filed at an early stage of the WDTX Litigation, which “has weighed against exercising the authority to deny institution[.]” IPR2020-00156, Paper 10, 9 & n.8 (June 15, 2020) (“*Seven*”), 11-12 (quoting *Fintiv*, 11). The District Court has yet to expend significant resources, and indeed, a Motion to Transfer Venue is still pending (TESLA-1024). *Seven*, 10-13; IPR2020-01019,

⁶ Over the past year, according to Docket Navigator.

Sotera, 14, 16-17; IPR2020-00820, Paper 15, 11-13 (Oct. 21, 2020) (“*Snap*”) (precedential). By contrast, Tesla has diligently filed this petition roughly four-to-five months ahead of the one-year time bar, while the WDTX Litigation remains in its early stages.

4. Factor 4: Petition Raises Unique Issues

The WDTX Litigation is still in its infancy, with no due date for initial invalidity contentions. Thus, at present, there is no overlap of issues between the WDTX Litigation and any IPR resulting from this petition. Such a lack of overlap “has tended to weigh against exercising discretion to deny institution.” *Fintiv*, 12-14. iQar also hasn’t identified allegedly infringing claims in the district court, while this petition addresses all claims.

5. Factor 5: Tesla’s Involvement in the WDTX Litigation

Tesla’s status as a defendant in the WDTX Litigation is at worst neutral under Factor 5. Indeed, *Fintiv* “says nothing about situations in which the petitioner is the same as, or is related to, the district court defendant.” IPR2020-00122, Paper 15, at 10 (Crumbley, dissenting); *Fintiv*, 13-14. At worst, where the parties are the same, Factor 5 is neutral. *Id.*

6. Factor 6: Other Considerations Support Institution

No party has previously challenged the ’426 Patent in a PTAB post-issuance proceeding. Further, the Board has not previously considered this petition’s grounds

or substantively similar grounds. Director Vidal has also explained that “the PTAB will not deny institution...under *Fintiv*...when a petition presents compelling evidence of unpatentability....” TESLA-1023, 3-5, 9; *see also Fintiv*, 14-15.

This Petition’s merits are compelling. As demonstrated herein and with reference to Dr. Papanikolopoulos’ testimony and additional evidence, institution would result in a finding of unpatentability of the Challenged Claims, which are obvious based on prior art references that materially differ from those considered during prosecution, and render obvious features relied upon by the Examiner in allowing the ’426 Patent. *See generally* TESLA-1003; TESLA-1002, 21-54.

Even if the WDTX Litigation may be relevant, the strength of the merits alone outweigh any inefficiencies borne of parallel litigation. TESLA-1023, 3-5, 9; *Fintiv*, 14-15. Accordingly, Factor 6 favors institution.

V. CONCLUSION AND FEES

The Challenged Claims are unpatentable. Petitioner authorizes charging fees to Deposit Account 06-1050.

VI. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

Tesla, Inc. is the petitioner and real party-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

The '426 Patent is the subject of a civil action, *iQar Inc. v. Tesla, Inc.*, 1:23-cv-00694, W.D. Tex., filed June 20, 2023 (TESLA-1021). Petitioner is not aware of any other disclaimers, reexamination certificates, or IPR petitions addressing the '426 Patent.

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation of counsel.

Lead Counsel	Backup Counsel
W. Karl Renner, Reg. No. 41,265 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR49649-0018IP1@fr.com	Karan Jhurani, Reg. No. 71,777 Vivian Chew Keller, Reg. No. 79,387 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 IPR49649-0018IP1@fr.com Ashraf A. Fawzy, Reg. No. 67,914 Tesla, Inc. 800 Connecticut Ave NW, 5th Floor Washington, DC 20005 Tel: 202-905-9221 afawzy@tesla.com

D. Service Information

Please address all correspondence and service to the address listed above. Petitioner consents to electronic service by email at 49649-0018IP1@fr.com and afawzy@tesla.com.

Respectfully submitted,

Dated: February 5, 2024

/Karan Jhurani/

W. Karl Renner, Reg. No. 41,265
Karan Jhurani, Reg. No. 71,777
Vivian Chew Keller, Reg. No. 79,387
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

Ashraf A. Fawzy, Reg. No. 67,914
Tesla, Inc.
800 Connecticut Ave NW, 5th Floor
Washington, DC 20005
Tel: 202-905-9221

(Control No. IPR2024-00541)

Counsel for Petitioner

CERTIFICATION UNDER 37 CFR § 42.24

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,981 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated: February 5, 2024

/Karan Jhurani/

W. Karl Renner, Reg. No. 41,265
Karan Jhurani, Reg. No. 71,777
Vivian Chew Keller, Reg. No. 79,387
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

Ashraf A. Fawzy, Reg. No. 67,914
Tesla, Inc.
800 Connecticut Ave NW, 5th Floor
Washington, DC 20005
Tel: 202-905-9221

Counsel for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on February 5, 2024, a complete and entire copy of this Petition for *Inter Partes* Review and all supporting exhibits were provided by Federal Express, to the Patent Owner, by serving the correspondence address of record as follows:

STEVENS LAW GROUP - INVENTLY AUTOMOTIVE
1754 Technology Drive, Suite 226
San Jose, CA 95110
UNITED STATES
(408) 288-7588

/Crena Pacheco/

Crena Pacheco
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
pacheco@fr.com