(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 September 2013 (06.09.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/130282 A1

(51) International Patent Classification:
H02P 21/13 (2006.01) *H02P 23/12* (2006.01)

(21) International Application Number:
PCT/US2013/026276

(22) International Filing Date:
15 February 2013 (15.02.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/410,623 2 March 2012 (02.03.2012) US

(71) Applicants: **DEERE & COMPANY** [—/US]; One John Deere Place, Moline, Illinois 61265 (US). **THE BOARD OF REGENTS OF THE UNIVERSITY OF NEBRASKA** [—/US]; University of Nebraska, Varner Hall, 3835 Holdrege, Lincoln, Nebraska 68583 (US).

(72) Inventors: **WU, Long**; 2824 37th Ave S, Fargo, North Dakota 58104 (US). **ZHAO, Yue**; 209N Scott Engineering Center, University of Nebraska-Lincoln, Lincoln, Nebraska 68588-0511 (US). **QIAO, Wie**; 9545 Forest Glen Drive, Lincoln, Nebraska 68508-1515 (US).

(74) Agents: **YACURA, Gary** et al.; Harness, Dickey, and Pierce PLC, 11730 Plaza America Drive, Suite 600, Reston, VA 20190 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

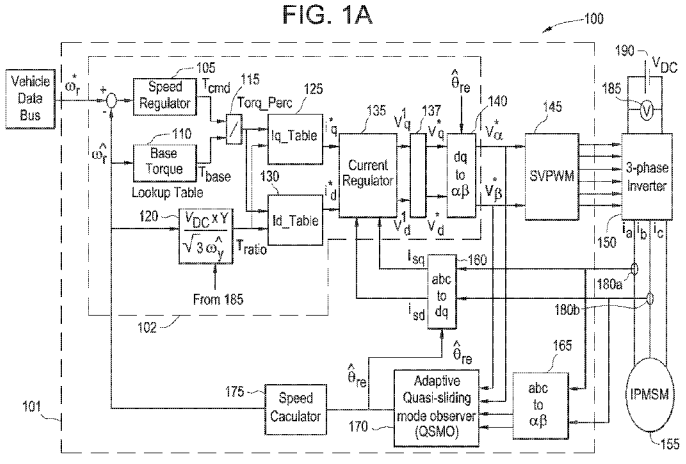
Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) Title: DRIVE SYSTEMS INCLUDING SLIDING MODE OBSERVERS AND METHODS OF CONTROLLING THE SAME

FIG. 1A



(57) Abstract: At least one example embodiment discloses a method of estimating a position of a rotor in a motor. The method includes determining a first estimated position of the rotor using a first algorithm, determining a second estimated position of the rotor using a second algorithm, the second algorithm being different than the first algorithm, determining a first error based on the first estimated position and the second estimated position, and determining a third estimated position of the rotor based on the first error.

WO 2013/130282 A1

DRIVE SYSTEMS INCLUDING SLIDING MODE OBSERVERS AND METHODS OF CONTROLLING THE SAME

FIELD

Example embodiments are related to electric drive device systems and/or methods for controlling electric drive devices such as Interior Permanent Magnet (IPM) motors or machines.

BACKGROUND

IPM synchronous motors (IPMSMs) are used in hybrid and electric vehicle systems. Rotor position of the IPMSM is used for high-performance traction or voltage control of the IPMSM.

Electromechanical type of position sensors, e.g., resolvers, optical encoders, and hall-effect sensors, are used to obtain the rotor position and/or speed in IPMSM drive systems. The use of these electromechanical sensors increases cost, size, weight, and hardware wiring complexity of the IPMSM drive systems. Moreover, mounting electromechanical sensors on a rotor of the motor affects the robustness of the IPMSM. Sensors are often subject to failures in harsh environments, such as excessive ambient temperature, super high-speed operation, and other adverse or heavy load conditions.

As an alternative to sensors, sensorless drives including observers are used. One type of an art recognized observer is a sliding mode observer (SMO). A SMO is embodied in a specific purpose computer, mainly a microcontroller or digital signal processor specifically programmed to execute the SMO. In general, a SMO is an observer having inputs that are discontinuous functions of an error between estimated and measured outputs. In a SMO, a manifold is designed such that a system state trajectory exhibits a certain behavior when confined to the manifold. The manifold may also be referred to as a sliding surface.

SUMMARY

At least one example embodiment discloses an adaptive Quasi-SMO (QSMO) to estimate the rotor position from extended back electromagnetic force (EMF) quantities in an IPMSM. The QSMO parameters are adaptive to the load and rotor speed.

At least one example embodiment discloses an extended back EMF-based adaptive QSMO for rotor position estimation for a sensorless IPMSM drive. In the context of the present application, a sensorless system is a position/speed sensorless system, where position sensors may not be used to measure rotor position.

In the present application, state trajectory may refer to a difference between measured current and estimated current.

The inventors have discovered that a discrete-time sliding mode observer (DSMO) with conventional switching functions, will keep tight regulation to force the state trajectory close to the sliding surface even when tracking error is within the width of the boundary layer. This could cause a chattering problem during a steady state. To mitigate this chattering problem and achieve a bounded motion within a limited boundary layer, a switching function may be implemented. The switching function leads to a quasi-sliding mode motion of the DSMO at steady state. Since the magnitude of the extended back EMF of the IPMSM changes with both load and speed variations, the parameters allow better performance than conventional SMOs.

At least another example embodiment discloses speed aided stabilizers to improve drive system stability and help the drive system go smoothly at load/speed transients. The inventors have discovered that commonly used methods to improve the stability are system state decoupling, which are used to disconnect direct input/output relationship between each sub-system. However, a second order IPMSM model is a simplified model and d-q axis equations are also coupling with each other. The inductances change with stator current and a gamma angle, and the stator resistance changes with

temperature. These parameter variations are difficult to accurately model in a machine model. For other modules in a close-loop system, the inverter and SMO are nonlinear, and linearization methods as well as other approximation methods are needed. So the whole sensorless control system is a high order, nonlinear, coupling system with unmodeled parameters and model uncertainties. The transfer function and pole placement based decoupling methods are difficult for both analysis and implementation.

The speed aided stabilizers are based on the idea that motor rotor speed changes much slower than the position changes in medium and high speed ranges. Thus, during the time interval of consecutive two sampling points, speed can be assumed as a constant value, and can be used to predict the position for a next step. This predicted position for a next step can be used as a reference to adjust the estimated position, so as to help the system go through the transient with high accuracy.

At least one example embodiment discloses a method of estimating a position of a rotor in a motor. The method includes determining a first estimated position of the rotor using a first algorithm, determining a second estimated position of the rotor using a second algorithm, the second algorithm being different than the first algorithm, determining a first error based on the first estimated position and the second estimated position, and determining a third estimated position of the rotor based on the first error.

In one example embodiment, the determining a first estimated position determines the first estimated position for a same point in time as the determining a second estimated position determines the second estimated position.

In one example embodiment, the determining a third estimated position determines the third estimated position for a point in time subsequent to the point in time associated with the determining a first estimated position and the determining a second estimated position.

In one example embodiment, the determining a first estimated position determines the first estimated position by a sliding mode observer by comparing an estimated current to a measured current associated with the motor.

In one example embodiment, the determining a second estimated position determines an estimated speed of the rotor, the second estimated position being based on the speed of the rotor and the third estimated position.

In one example embodiment, the determining a second estimated position determines an estimated speed of the rotor, the second estimated position being based on the estimated speed of the rotor.

In one example embodiment, the determining a third estimated position includes determining if the first error exceeds an error margin, and setting the third estimated position to the second estimated position if the first error exceeds the error margin.

In one example embodiment, the determining a third estimated position includes determining if the first error exceeds an error margin, and setting the third estimated position to the first estimated position if the first error is below the error margin.

In one example embodiment, the determining a first error determines the first error a difference between the first estimated position and the second estimated position.

In one example embodiment, the determining a first error includes determining a current error between the first estimated position and the second estimated position, and determining the first error based on the current error and a previous error.

In one example embodiment, the determining a first estimated position determines the first estimated position by a sliding mode observer.

In one example embodiment, the determining a second estimated position determines an estimated speed of the rotor, the second estimated position being based on the speed of the rotor.

In one example embodiment, the determining a second estimated position determines an estimated speed of the rotor, the second estimated position being based on the speed of the rotor.

In one example embodiment, the determining the first error based on a current error and a previous error adds the current error and the previous error.

In one example embodiment, the determining a third estimated position includes determining if the first error exceeds an error margin, and setting the third estimated position to the first estimated position if the first error is below the error margin.

In one example embodiment, the determining a third estimated position includes determining if the first error exceeds an error margin, and setting the third estimated position to the second estimated position minus the first error if the first error exceeds the error margin.

At least one example embodiment discloses a drive system including a controller configured to determine a first estimated position of the rotor using a first algorithm, determine a second estimated position of the rotor using a second algorithm, the second algorithm being different than the first algorithm, determine a first error based on the first estimated position and the second estimated position, and determine a third estimated position of the rotor based on the first error.

In one example embodiment, the drive system further includes a buffer coupled to the controller and configured to store data for determining the second estimated position.

BRIEF DESCRIPTION OF THE DRAWINGS

Example embodiments will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings. FIGS. 1A-6 represent non-limiting, example embodiments as described herein.

FIG. 1A illustrates a drive system for controlling an IPM machine according to an example embodiment;

FIG. 1B illustrates a data processing system of the drive system of FIG. 1A according to an example embodiment;

FIG. 2A illustrates an example embodiment of a QSMO shown in FIG. 1A;

FIG. 2B illustrates an example embodiment of a variable switching function implemented by a switching block according to an example embodiment;

FIG. 2C illustrates an example embodiment of a portion of the QSMO shown in FIG. 1A;

FIG. 2D illustrates an example embodiment of a portion of the QSMO shown in FIG. 1A;

FIG. 2E illustrates an example embodiment of a parameter lookup table;

FIG. 3 illustrates a method of estimating a rotor position in a motor according to an example embodiment;

FIGS. 4A-4D illustrate an implementation of a speed buffer for a speed aided stabilizer, according to an example embodiment;

FIG. 5 illustrates a first method of stabilizing speed according to an example embodiment; and

FIG. 6 illustrates a second method of stabilizing speed according to an example embodiment.

DETAILED DESCRIPTION

Various example embodiments will now be described more fully with reference to the accompanying drawings in which some example embodiments are illustrated.

Accordingly, while example embodiments are capable of various modifications and alternative forms, embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that there is no intent to limit example embodiments to the particular forms disclosed, but on the contrary, example embodiments are to cover all modifications, equivalents, and alternatives falling within the scope of the claims. Like numbers refer to like elements throughout the description of the figures.

It will be understood that, although the terms first, second, etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and, similarly, a second element could be termed a first element, without departing from the scope of example embodiments. As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

It will be understood that when an element is referred to as being "connected" or "coupled" to another element, it can be directly connected or coupled to the other element or intervening elements may be present. In contrast, when an element is referred to as being "directly connected" or "directly coupled" to another element, there are no intervening elements present. Other words used to describe the relationship between elements should be interpreted in a like fashion (*e.g.*, "between" versus "directly between," "adjacent" versus "directly adjacent," etc.).

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of

example embodiments. As used herein, the singular forms "a," "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises," "comprising," "includes" and/or "including," when used herein, specify the presence of stated features, integers, steps, operations, elements and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components and/or groups thereof.

It should also be noted that in some alternative implementations, the functions/acts noted may occur out of the order noted in the figures. For example, two figures shown in succession may in fact be executed substantially concurrently or may sometimes be executed in the reverse order, depending upon the functionality/acts involved.

Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which example embodiments belong. It will be further understood that terms, e.g., those defined in commonly used dictionaries, should be interpreted as having a meaning that is consistent with their meaning in the context of the relevant art and will not be interpreted in an idealized or overly formal sense unless expressly so defined herein.

Portions of example embodiments and corresponding detailed description are presented in terms a processor specifically programmed to execute software, or algorithms and symbolic representations of operation on data bits within a computer memory. These descriptions and representations are the ones by which those of ordinary skill in the art effectively convey the substance of their work to others of ordinary skill in the art. An algorithm, as the term is used here, and as it is used generally, is conceived to be a self-consistent sequence of steps leading to a result. The steps are those requiring physical manipulations of physical quantities. Usually, though not

necessarily, these quantities take the form of optical, electrical, or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

In the following description, illustrative embodiments will be described with reference to acts and symbolic representations of operations (e.g., in the form of flowcharts) that may be implemented as program modules or functional processes including routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types and may be implemented using existing hardware. Such existing hardware may include one or more Central Processing Units (CPUs), digital signal processors (DSPs), application-specific-integrated-circuits, field programmable gate arrays (FPGAs) computers or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise, or as is apparent from the discussion, terms such as “processing” or “computing” or “calculating” or “determining” or “displaying” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical, electronic quantities within the computer system’s registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

Note also that the software implemented aspects of example embodiments are typically encoded on some form of tangible (or recording) storage medium or implemented over some type of transmission medium. The tangible storage medium may be magnetic

(e.g., a floppy disk or a hard drive) or optical (e.g., a compact disk read only memory, or "CD ROM"), and may be read only or random access.

In a discrete-time sliding mode controller (DSMC) or observer, to facilitate DSP or micro-controller based applications, controller inputs are calculated once per sampling period and held constant during this interval. The inventors have discovered that due to a finite sampling period or PWM switching frequency, the state trajectory is difficult to precisely move along the sliding surface, which will lead to a quasi-sliding mode motion only. For IPMSM drive applications, limitations in control loop frequency and CPU loading make it challenging to achieve high accuracy in position estimation. Moreover, a magnitude of extended back electromagnetic force (EMF) contains both a speed-related term and current-related terms, which means that both load and speed will affect the magnitude of the extended back EMF.

At least one example embodiment discloses an adaptive Quasi-SMO (QSMO) to estimate the rotor position from the extended back electromagnetic force (EMF) quantities in an IPMSM. The QSMO parameters are adaptive to the load and rotor speed.

FIGS. 1A-2D illustrate a drive system including a motor having a rotor, the motor configured to receive a measured current, a controller configured to generate a voltage command for the motor, a sliding mode observer configured to determine an estimated current for the motor based on the voltage command, determine a difference between the measured current and the estimated current, and determine a switching control vector and an estimator configured to estimate a rotor position based on the switching control vector, the switching control vector being determined based on the difference and adaptive parameters of the sliding mode observer. The controller is configured to control the motor based at least in part on the estimated rotor position

In accordance with an example embodiment, FIG. 1A illustrates a drive system 100 for controlling an IPM machine such as a motor

155 (e.g., an interior permanent magnet synchronous motor (IPMSM)) or another alternating current machine. The drive system 100 may also be referred to as an IPMSM drive system.

It should be understood that the drive system 100 may include additional features that are not illustrated in FIG. 1A. For example, the drive system 100 may include a rotor magnet temperature estimation module, a current shaping module, and a terminal voltage feedback module. The features shown in FIG. 1A are illustrated for the convenience of describing the drive system 100 and it should be understood that the drive system 100 should not be limited to the features shown in FIG. 1A.

The system 100 includes electronic modules, software modules, or both. In an example embodiment, the drive system 100 includes an electronic data processing system 101 to support storing, processing or execution of software instructions of one or more software modules. The electronic data processing system 101 is indicated by the dashed lines in FIG. 1A and is shown in greater detail in FIG. 1B.

The data processing system 101 is coupled to an inverter circuit 150. The inverter circuit 150 may be a three-phase inverter. The inverter circuit 150 includes a semiconductor drive circuit that drives or controls switching semiconductors (e.g., insulated gate bipolar transistors (IGBT) or other power transistors) to output control signals for the motor 155. In turn, the inverter circuit 150 is coupled to the motor 155. The motor 155 is associated with sensors 180a and 180b.

Throughout the specification, the sensors 180a and 180b are referred to as current transducers. However, it should be understood that the sensors 180a and 180b may be another type of current sensor.

The current transducers 180a and 180b and the motor 155 are coupled to the data processing system 101 to provide feedback data (e.g., current feedback data, such as phase current values ia and ib),

raw position signals, among other possible feedback data or signals, for example. While only two current transducers 180a and 180b are shown, it should be understood that the driver system 100 may implement three current transducers.

The data processing system 101 includes a software controller 102, converters 160, 165, a pulse width generation module 145, a QSMO 170 and a speed calculator 175.

The software controller 102, converters 160, 165, the pulse width generation module 145, the QSMO 170 and the speed calculator 175 are software modules. While the software controller 102, converters 160, 165, the pulse width generation module 145, the QSMO 170 and the speed calculator 175 are described as executing functions, it should be understood that a data processor, such as a digital signal processor or a microcontroller, is specifically programmed to execute the software controller 102, converters 160, 165, the pulse width generation module 145, the QSMO 170 and the speed calculator 175. For example, a data processor 264 is specifically programmed to execute the software controller 102, converters 160, 165, the pulse width generation module 145, the QSMO 170 and the speed calculator 175, as will be described in FIG. 1B.

The controller 102 includes a speed regulator 105, a base torque lookup table (LUT) 110, a torque processor 115, a ratio calculator 120, a q-axis current (iq) command LUT 125, a d-axis current (id) command LUT 130, a current regulator 135, a voltage compensator 137 and a converter 140.

In an example embodiment, a speed regulator 105 receives input data representing a difference between an estimated rotor speed $\hat{\omega}_r$ and a command rotor speed ω_r^* as input.

The command rotor speed ω_r^* may be input by a controller (shown as 266 in FIG. 1B) via a vehicle data bus 118. For example, if an operator wants the motor 155 to run at 5,000 RPM, the operator

inputs 5,000 RPM into the controller and the vehicle data bus 118 inputs the command rotor speed ω_r^* to the controller 102. The speed regulator 105 converts the received input data into a torque command T_{cmd}.

While the term command is used throughout the specification, it should be understood that command refers to a target value.

The base torque LUT 110 determines a base torque value T_{base} based on the estimated rotor speed $\hat{\omega}_r$.

From the estimated rotor speed $\hat{\omega}_r$, base torque values are respectively associated with discrete speed points with a nominal dc bus voltage level. In other words, the two-dimensional base torque LUT 110 is established from a motor characterization procedure. During the IPM motor characterization procedure, each rotor shaft speed has a maximum output torque, which is defined as the base torque at that speed. Thus, the base torque may also be referred to as peak torque.

The base torque LUT 110 outputs the associated base torque value as the base torque value T_{base} to the torque processor 115.

The torque processor 115 receives the base torque value T_{base} and the torque command T_{cmd}. The torque command T_{cmd} may be in Nm.

The torque processor 115 is configured to determine an absolute value of the torque command T_{cmd}. The torque processor 115 is configured to convert the absolute value of the torque command T_{cmd} into a percentage Torq_Perc of the base torque value T_{base}. The torque processor 115 outputs the percentage Torq_Perc to the q-axis current (iq) command LUT 125 and the d-axis current (id) command LUT 130.

In addition to sending the estimated rotor speed $\hat{\omega}_r$ to the base torque LUT 110, the speed calculator 175 sends the estimated rotor speed $\hat{\omega}_r$ to the ratio calculator 120.

In addition to receiving the estimated rotor speed $\hat{\omega}_r$, the ratio

calculator 120 is configured to receive the measured operating dc bus voltage value. The measured operating dc bus voltage value is provided by a voltage sensor 185 which measures the DC bus in the inverter circuit 150. The inverter circuit 150 is powered by a direct current (dc) voltage bus. The ratio calculator 120 adjusts the operating DC bus voltage V_{DC} by the voltage sensor 185 to the detected operating rotor shaft speed ratio as follows:

$$T_{ratio} = \frac{V_{DC} \times Y}{\sqrt{3}\,\hat{\omega}_r} \qquad (1)$$

where T_{ratio} is the adjusted detected operating DC bus voltage to the detected operating rotor shaft speed ratio and Y is a coefficient. For example, the coefficient Y may be 0.9. The ratio T_{ratio} is output by the ratio calculator 120 to the d-q axis current command LUTs 125 and 130.

The q-axis current command (iq) LUT 125 and the d-axis current command (id) LUT 130 are configured to receive the ratio T_{ratio}. The q-axis current command LUT 125 and the d-axis current command LUT 130 store q-axis and d-axis current commands, respectively, each of which is associated with a pair of ratio and torque percentage values. The development of the q-axis current command LUT 125 and the d-axis current command LUT 130 may be done using any known method.

The d-q axis current refers to the direct axis current and the quadrature axis current as applicable in the context of vector-controlled alternating current machines, such as the motor 155.

The d-axis current command LUT 130 is configured to output a d-axis current command i_d^* that is associated with the received torque percentage Torq_Perc and the ratio T_{ratio}. As shown in FIG. 1A, the d-axis current command i_d^* is output to the current regulator 135.

The q-axis current command LUT 125 is configured to output a q-axis current command i_q^* that is associated with the received torque percentage Torq_Perc and ratio T_{ratio}.

It should be understood that i_d^* and i_q^* are current commands for a stator of the motor 155.

While the q-axis current command LUT 125 and d-axis current command LUT 130 are illustrated and described as LUTs, it should be understood that the q-axis current command LUT 125 and d-axis current command LUT 130 may be implemented as a set of equations that relate respective torque commands to corresponding direct and quadrature axes currents, or a set of rules (e.g., if-then rules) that relates respective torque commands to corresponding direct and quadrature axes currents.

As shown in FIG. 1A, the q-axis current command i_q^* is output to the current regulator 135.

The current regulator 135 is capable of communicating with the pulse-width modulation (PWM) generation module 145 (e.g., space vector PWM generation module). The current regulator 135 receives respective d-q axis current commands (e.g., i_d^* and i_q^*) and measured d-q axis currents (e.g., i_d and i_s) for the stator and outputs corresponding pre-compensated d-q axis voltage commands v_d' and v_q' to a voltage compensator 137.

The voltage compensator 137 provides voltage adjustment data to adjust the pre-compensated d-q axis voltage commands v_d' and v_q' and outputs d-q axis voltage commands v_d^* and v_q^*. It should be understood that the voltage compensator 137 may generate the d-q axis voltage commands (e.g., v_d^* and v_q^* commands) using any known method such as current feed forward compensation.

The converter 140 receives the d-q axis voltage commands v_d^*

and v_q^* and performs an inverse Park transformation to generate α-β axis voltage commands v_α^* and v_β^*. While at least one example embodiment is described using the α-β axis, it should be understood that example embodiments may be implemented using the d-q axis or three phase representation of a control vector.

In an example embodiment, the PWM generation module 145 converts the α axis voltage and β axis voltage data (voltage commands v_α^* and v_β^*) from two phase data representations into three phase representations (e.g., three phase voltage representations, such as va*, vb* and vc*) for control of the motor 155, for example. Outputs of the PWM generation module 145 are coupled to the inverter circuit 150.

The inverter circuit 150 includes power electronics, such as switching semiconductors to generate, modify and control pulse-width modulated signals or other alternating current signals (e.g., pulse, square wave, sinusoidal, or other waveforms) applied to the motor 155. The PWM generation module 145 provides inputs to a driver stage within the inverter circuit 150. An output stage of the inverter circuit 150 provides a pulse-width modulated voltage waveform or other voltage signal for control of the motor 155. In an example embodiment, the inverter 150 is powered by the direct current (dc) voltage bus voltage V_{DC}.

The current transducers 180a, 180b measure two of three phase current data ia and ib, respectively, applied to the motor 155. It should be understood that an additional current transducer may also measure a third phase current data ic.

The converter 160 may apply a Clarke transformation or other conversion equations (e.g., certain conversion equations that are suitable and are known to those of ordinary skill in the art) to convert the measured three-phase representations of current into two-phase representations of current based on the current data ia and ib from the current transducers 180a, 180b and an estimated rotor position

$\hat{\theta}_{re}$ from the QSMO 170. The output of the converter 160 module (i_d, i_q) is coupled to the current regulator 135.

The converter 165 may apply a Park transformation or other conversion equations (e.g., certain conversion equations that are suitable and are known to those of ordinary skill in the art) to convert the measured three-phase representations of current into two-phase representations of current based on the current data ia and ib from the current transducers 180a, 180b. The output of the converter 160 module (measured currents i_α, i_β) is coupled to the QSMO 170.

The QSMO 170 receives the measured currents i_α, i_β and the voltage commands v_α^* and v_β^*. Based on the measured currents i_α, i_β and the voltage commands v_α^* and v_β^*, the QSMO 170 is configured to output the estimated rotor position $\hat{\theta}_{re}$ to the speed calculator 175 and the converter 160, as will be described in greater detail in FIG. 2.

The speed calculator 175 may convert the estimated rotor position $\hat{\theta}_{re}$ provided by the QSMO 170 into the estimated rotor speed $\hat{\omega}_r$.

In FIG. 1B, the electronic data processing system 101 includes an electronic data processor 264, a data bus 262, a data storage device 260, and one or more data ports (268, 270, 272 and 274). The data processor 264, the data storage device 260 and one or more data ports are coupled to the data bus 262 to support communications of data between or among the data processor 264, the data storage device 260 and one or more data ports.

In an example embodiment, the data processor 264 may include an electronic data processor, a digital signal processor, microprocessor, a microcontroller, a programmable logic array, a logic circuit, an arithmetic logic unit, an application specific integrated

circuit, a digital signal processor, a proportional-integral-derivative (PID) controller, or another data processing device.

The data storage device 260 may include any magnetic, electronic, or optical device for storing data. For example, the data storage device 260 may include an electronic data storage device, an electronic memory, non-volatile electronic random access memory, one or more electronic data registers, data latches, a magnetic disc drive, a hard disc drive, an optical disc drive, or the like.

Moreover, in one example embodiment the data storage device 260 may store the controller 102, pulse width generation module 145, converters 160, 165, the QSMO 170 and the speed controller 175 to be used executed by the data processor 264. The data processor 264 may access the data storage device 260 and execute the controller 102, pulse width generation module 145, converters 160, 165, the QSMO 170 and the speed controller 175 via the data bus 262.

As shown in FIG. 1B, the data ports include a first data port 268, a second data port 270, a third data port 272 and a fourth data port 274, although any suitable number of data ports may be used. Each data port may include a transceiver and buffer memory, for example. In an example embodiment, each data port may include any serial or parallel input/output port.

In an example embodiment as illustrated in FIG. 1B, the first data port 268 is coupled to the vehicle data bus 118. In turn, the vehicle data bus 118 is coupled to a controller 266. In one configuration, the second data port 270 may be coupled to the inverter circuit 150; the third data port 272 may be coupled to the voltage sensor 185; and the fourth data port 274 may be coupled to the transducers 180a and 180b.

In an example embodiment of the data processing system 101, the speed regulator 105 is associated with or supported by the first data port 268 of the electronic data processing system 101. The first data port 268 may be coupled to a vehicle data bus 118, such as a

controller area network (CAN) data bus. The vehicle data bus 118 may provide data bus messages with torque commands to the speed regulator 105 via the first data port 268. The operator of a vehicle may generate the torque commands via a user interface, such as a throttle, a pedal, the controller 266, or other control device.

FIG. 2A illustrates an example embodiment of the QSMO 170 shown in FIG. 1A. The QSMO 170 is configured to, receive a measured current for a motor, determine an estimated current for a motor, determine a difference between the measured current and the estimated current, determine a switching control vector, and estimate a rotor position based on the switching control vector, the switching control vector being determined based on the difference and adaptive parameters of the sliding mode observer.

The dynamics of the motor 155 can be modeled in the *d-q* rotating reference frame as:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R + pL_d & -\omega_{re} L_q \\ \omega_{re} L_d & R + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{re} \psi_m \end{bmatrix} \tag{2}$$

where p is the derivative operator, v_{sd}, v_{sq}, i_{sd}, and i_{sq} are the stator voltages and currents, respectively, ω_{re} is the rotor electrical speed, and Ψ_m is the magnetic flux linkage of the motor 155, L_d and L_q are the *d*-axis and *q*-axis inductances, respectively; and R is the stator resistance. As should be understood, ω_{re} is the rotor electrical speed and ω_r is the mechanical speed and ω_{re} is ω_r multiplied by the pole-pairs number.

Using the inverse Park transformation, the dynamic model of the motor 155 in the *α-β* stationary reference frame can be expressed as:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = p \begin{bmatrix} L + \Delta L \cos(2\theta_{re}) & \Delta L \sin\theta_{re} \\ \Delta L \sin\theta_{re} & L - \Delta L \cos(2\theta_{re}) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + R \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \omega_{re} \psi_m \begin{bmatrix} -\sin\theta_{re} \\ \cos\theta_{re} \end{bmatrix}$$

(3)

where $L = \frac{L_d + L_q}{2}$; $\Delta L = \frac{L_d - L_q}{2}$; and θ_{re} is the rotor position angle. While the electrical quantity θ_{re} is used described as the rotor position angle, it should be understood that a mechanical quantity may be used where the mechanical quantity is the electrical quantity θ_{re} divided by a number of magnetic pole pairs p_o of the motor 155.

Due to the saliency of the motor (i.e., $L_d \neq L_q$), both the back EMF and the inductance matrix contain the information of the rotor position angle. Moreover, equation (3) contains both $2\theta_{re}$ and θ_{re} terms. To facilitate rotor position observation, an extended back EMF-based model for the motor 155 may be used as follows:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} R + pL_d & \omega_{re}(L_d - L_q) \\ \omega_{re}(L_q - L_d) & R + pL_d \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \left[(L_d - L_q)(\omega_{re} i_d - p i_q) + \omega_{re}\psi_m \right] \begin{bmatrix} -\sin\theta_{re} \\ \cos\theta_{re} \end{bmatrix} \quad (4)$$

In equation (4) only the extended back EMF term contains the information of the rotor position. Thus, the rotor position can be extracted by an inverse tangent method or angle tracking observer using an estimated extended back EMF.

Back EMF is defined as:

$$\omega_{re}\psi_m \begin{bmatrix} -\sin\theta_{re} \\ \cos\theta_{re} \end{bmatrix}$$

Whereas extended back EMF may be defined as:

$$\left[(L_d - L_q)(\omega_{re} i_d - p i_q) + \omega_{re}\psi_m \right] \begin{bmatrix} -\sin\theta_{re} \\ \cos\theta_{re} \end{bmatrix}$$

A magnitude of the extended back EMF may be:

$$\eta = (L_d - L_q)\,(\omega_{re} i_d - p i_q) + \omega_{re}\psi_m \tag{5}$$

The dynamic current equations of the motor 155 can be expressed in matrix form:

$$X = AX + BV + E \tag{6}$$

where

$$\mathbf{V} = [v_\alpha \quad v_\beta]^T \tag{7}$$

$$\mathbf{E} = \eta/L_d\,[\sin\theta_{re} \quad -\cos\theta_{re}]^T = [E_\alpha \quad E_\beta]^T \tag{8}$$

$$\mathbf{X} = [i_\alpha \quad i_\beta]^T \tag{9}$$

which is selected as the system state;

$$\mathbf{A} = \begin{bmatrix} -R/L_d & \omega_{re}\,L_q - L_d/L_d \\ \omega_{re}\,L_d - L_q/L_d & -R/L_d \end{bmatrix} \tag{10}$$

and

$$\mathbf{B} = \begin{bmatrix} 1/L_d & 0 \\ 0 & 1/L_d \end{bmatrix} \tag{11}$$

Using the first-order Euler method, equation (6) may be written into a discrete-time model as follows:

$$\begin{cases} i_\alpha[k+1] = T_s\left(\frac{v_\alpha}{L_d} + \omega_{re}\frac{L_q - L_d}{L_d} i_\beta[k] + E_\alpha[k]\right) + \left(1 - \frac{T_s R}{L_d}\right) i_\alpha[k] \\ i_\beta[k+1] = T_s\left(\frac{v_\beta}{L_d} - \omega_{re}\frac{L_q - L_d}{L_d} i_\alpha[k] + E_\beta[k]\right) + \left(1 - \frac{T_s R}{L_d}\right) i_\beta[k] \end{cases} \quad (12)$$

wherein T_s is the sampling period.

Referring back to FIG. 2A, a QSMO 170a includes a sliding mode current estimator 205, a switching block 210, a low-pass filter 215, a position calculator 220 and a phase shift compensator 225. The QSMO 170a may be executed by a digital signal processor or microcontroller. For example, the QSMO 170a may be software stored in a tangible computer readable medium and executed by the data processor 264.

The QSMO 170a receives the voltage commands υ_α and υ_β and the measured currents i_α and i_β converted from measured phase currents i_a, i_b and i_c.

The sliding mode current estimator 205 receives the voltage commands v_α^* and v_β^* and a switching control vector $Z_{\alpha\beta}$. It should be understood that terms having "αβ" as subscripts represent a vector having both the α-axis term and the β-axis term. For example, voltage commands v_α^* and v_β^* may be represented as $v_{\alpha\beta}^*$ such that $(\upsilon^*_{\alpha\beta})^T$ equals $[\upsilon^*_\alpha \; \upsilon^*_\beta]$.

Using equation (12), the sliding mode current estimator 205 samples the voltage commands v_α^* and v_β^* and the switching control vector $Z_{\alpha\beta}$ and generates an estimated current value $\hat{i}_{\alpha\beta}$ ($\hat{i}_\alpha$ and $\hat{i}_\beta$) as follows:

$$\begin{cases} \hat{i}_\alpha[k+1] = T_s\left(\frac{v_\alpha^*}{L_d} + \omega_{re}\frac{L_q - L_d}{L_d} i_\beta[k] + lZ_\alpha[k]\right) + \left(1 - \frac{T_s R}{L_d}\right)\hat{i}_\alpha[k] \\ \hat{i}_\beta[k+1] = T_s\left(\frac{v_\beta^*}{L_d} - \omega_{re}\frac{L_q - L_d}{L_d} i_\alpha[k] - lZ_\beta[k]\right) + \left(1 - \frac{T_s R}{L_d}\right)\hat{i}_\beta[k] \end{cases} \quad (13)$$

As should be noted, equation (13) has the same form as equation (12). However, in equation (13), the voltage commands v^*_α and v^*_β are used, which are received from the converter 140, so that the terminal voltage does not need to be measured. However, if a dead-time effect is not fully compensated for, the voltage command will not equal the terminal voltage. This voltage mismatch will bring some error to the estimated position.

A comparator 207 receives the measured currents i_α and i_β and the estimated currents $\hat{i}_\alpha$ and $\hat{i}_\beta$. The comparator 207 determines errors ε_α and ε_β between the measured currents i_α and i_β and the estimated currents $\hat{i}_\alpha$ and $\hat{i}_\beta$, respectively. The errors ε_α and ε_β may be differences between the measured currents i_α and i_β and the estimated currents $\hat{i}_\alpha$ and $\hat{i}_\beta$, respectively.

The comparator 207 sends the errors ε_α and ε_β to the switching block 210. Based on the errors ε_α and ε_β, the switching block 210 determines the switching control vector $Z_{\alpha\beta}$. The switching control vector $Z_{\alpha\beta}$ contains the information of back EMF. Since the switching control vector $Z_{\alpha\beta}$ is the output of switching block 210, it contains heavy switching noise. The low-pass filter 215 receives the switching control vector $Z_{\alpha\beta}$ to smooth the estimated back EMF profile. The low-pass filter 215 adds a phase delay to original signal input. The phase shift compensator 225 compensates for the phase delay.

Thus, a tracking error between the measured current and estimated current is:

$$\boldsymbol{\varepsilon}[k]^T = \left[I_\alpha[k] - \hat{I}_\alpha[k] \quad I_\beta[k] - \hat{I}_\beta[k] \right] \tag{14}$$

More specifically, equation (15) can be obtained by subtracting (13) from (12):

$$\begin{cases} \varepsilon_\alpha[k+1] = \left(1 - \dfrac{T_s R}{L_d}\right)\varepsilon_\alpha[k] + T_s E_\alpha[k] - T_s l Z_\alpha \\ \varepsilon_\beta[k+1] = \left(1 - \dfrac{T_s R}{L_d}\right)\varepsilon_\beta[k] + T_s E_\beta[k] - T_s l Z_\beta \end{cases} \tag{15}$$

The QSMO 170a has a sliding surface s[*k*], which is designed to equal the tracking error ε[*k*] and 0.

The switching block 210 implements a variable switching function as follows:

$$Z_{\alpha\beta} = \begin{cases} Z_0 & \varepsilon[k] > Z_0 \\ \varepsilon[k] & -Z_0 < \varepsilon[k] < Z_0 \\ -Z_0 & \varepsilon[k] < -Z_0 \end{cases} \tag{16}$$

where Z_0 is an adaptive parameter. A boundary layer is formed between $-Z_0$ and Z_0.

FIG. 2B illustrates an example embodiment of the variable switching function implemented by the switching block 210. As shown in FIG. 2B, if the tracking error ε[*k*] is larger than the adaptive parameter Z_0, the switching block 210 outputs Z_0 as the switching control vector $Z_{\alpha\beta}$. If the tracking error ε[*k*] is smaller than $-Z_0$, the switching block 210 outputs $-Z_0$ as the switching control vector $Z_{\alpha\beta}$. If the tracking error ε[*k*] is between $-Z_0$ and Z_0, the output of switching function will be the tracking error ε[*k*]. In this case, the tracking error ε[*k*] is the switching control vector $Z_{\alpha\beta}$. The switching block 210 outputs the switching control vector $Z_{\alpha\beta}$ to the sliding mode current

estimator 205 and the low-pass filter 215.

If the reaching condition can be satisfied, the state trajectory will approach the sliding surface. When the tracking error $\varepsilon[k]$ is limited within a boundary layer (e.g., between $-Z_0$ and Z_0), the output of the switching block 210 is equal to the extended back EMF with heavy noise. In other words, $Z_{\alpha\beta}$ equals the extended back EMF with high order harmonics. The determination of the adaptive parameter Z_0 is described in greater detail below.

It should be understood that the variable switching function may be implemented as hardware or software used with an associated processor to execute the switching function.

The low-pass filter 215 receives the switching control vector $Z_{\alpha\beta}$ from the switching block 210 and filters the switching control vector $Z_{\alpha\beta}$ to remove noise. The low-pass filter 215 may be a second order low pass Butterworth filter generated in the S-plane, for example.

The low-pass filter 215 outputs the filtered switching control vector $Z_{\alpha\beta}$ as an estimated back EMF $\hat{e}_{\alpha\beta}$ to the position calculator 220. The output of the low-pass filter 215 has a same magnitude as the extended back EMF, however, the output of the low-pass filter 215 has a phase difference with respect to the extended back EMF.

The position calculator 220 determines a rotor position of the motor 155 based on the estimated back EMF $\hat{e}_{\alpha\beta}$. The rotor position determined by the position calculator 220 is not compensated for the phase shift caused by the low-pass filter 215. Therefore, the phase shift compensator 225 adds a phase shift $\Delta\theta_{re}$ to the rotor position at a logic unit 227 to compensate for the phase shift. More specifically, the phase shift compensator 225 compensates for the phase difference caused by the low-pass filter 215 using the phase-frequency characteristic of the low-pass filter 215 determined by the phase shift compensator 225.

The QSMO 170 sums the output from the position calculator

220 and the phase shift $\Delta\theta_{re}$ to produce the estimated rotor position $\hat{\theta}_{re}$.

The position calculator 220 and the phase shift compensator 225 may be referred to as an estimator configured to estimate a position of the rotor based on the switching control vector. As described above, the switching control vector is determined by the QSMO 170a based on the difference between differences between the measured currents i_α and i_β and the estimated currents $\hat{i}_\alpha$ and $\hat{i}_\beta$, respectively, and the adaptive parameters Z_0 and l.

Determination of the Adaptive Parameters

The QSMO 170a is configured to determine adaptive parameters Z_0 and l.

Equation (15) depicts dynamics of the state trajectory (difference between the measured current and the estimated current) of the QSMO 170. In order to ensure the state trajectory can move from an initial point to the sliding surface after a finite time step, the dynamic of state trajectory has a convergent behavior. Thus, the adaptive parameter Z_0 and an adaptive parameter l satisfy this convergent condition. The adaptive parameter l is the gain of the QSMO 170.

Alpha axis and beta axis equations have an identical structure in equation (15). Thus, if the convergence can be proved for the equation in alpha axis, the equation in beta axis can be identically proved. In one example embodiment, the following procedures are implemented by the QSMO 170:

(i) The state trajectory moves in the direction of the sliding surface when the error is outside the width of the boundary layer ($|\varepsilon[k]|$ is greater than Z_0). Consequently, when the tracking error $\varepsilon[k]$ is greater than the adaptive parameter Z_0, the tracking error $\varepsilon[k+1]$ is

less than the tracking error $\varepsilon[k]$; while when the tracking error $\varepsilon[k]$ is less than $-Z_0$, the tracking error $\varepsilon[k+1]$ is greater than the tracking error $\varepsilon[k]$.

(ii) In order to reduce the state trajectory change between the kth and $(k+1)^{th}$ samples, the tracking error $\varepsilon[k]$ is greater than Z_0 when the tracking error $\varepsilon[k+1]$ plus the tracking error $\varepsilon[k]$ is greater than 0; and when the tracking error $\varepsilon[k]$ is less than $-Z_0$ when the tracking error $\varepsilon[k+1]$ plus the tracking error $\varepsilon[k]$ is less than zero 0.

Condition (i) controls the direction of the state trajectory (error goes up then goes down) and condition (ii) controls the amount of change between two consecutive samples. Hence, when both conditions (i) and (ii) are implemented, the state trajectory can move from an initial condition to the sliding surface and remains in the boundary layer (between $-Z_0$ and Z_0).

For example, in condition (i), if the tracking error $\varepsilon_\alpha[k]$ is greater than the adaptive parameter Z_0, such that Z_a equals the adaptive parameter Z_0, the tracking error $\varepsilon_\alpha[k+1]$ would be less than the tracking error $\varepsilon_\alpha[k]$, which can be formulated based on equation (15) as:

$$\varepsilon_\alpha[k+1]-\varepsilon_\alpha[k]=-\frac{T_s R}{L_d}\varepsilon_\alpha[k]+T_s E_\alpha[k]-T_s l Z_0<0 \qquad (17)$$

which can also be formulated as:

$$lZ_0>-\frac{R}{L_d}\varepsilon_\alpha[k]+E_\alpha[k] \qquad (18)$$

because the adaptive parameter Z_0 is less than the tracking error $\varepsilon_\alpha[k]$

and greater than 0, and $-\frac{R}{L_d}Z_0 > -\frac{R}{L_d}\varepsilon_\alpha[k]$.

If the following inequality is satisfied, equations (17) and (18) are satisfied:

$$\left(l+\frac{R}{L_d}\right)Z_0 > E_\alpha[k] \tag{19}$$

Because R/L_d is positive, a stronger condition can be obtained as lZ_0 is greater than the amplitude $E_\alpha[k]$, which means if lZ_0 is greater than the magnitude of the extended back EMF η, the tracking error $\varepsilon_\alpha[k+1]$ being greater than the tracking error $\varepsilon_\alpha[k]$ can be satisfied when the tracking error $\varepsilon_\alpha[k]$ is greater than adaptive parameter Z_0.

If the tracking error $\varepsilon_\alpha[k]$ is less than $-Z_0$, such that Z_a equals $-Z_0$, the tracking error $\varepsilon_\alpha[k+1]$ would be greater than the tracking error $\varepsilon_\alpha[k]$, which can be formulated based on equation (15) as:

$$\varepsilon_\alpha[k+1]-\varepsilon_\alpha[k] = -\frac{T_s R}{L_d}\varepsilon_\alpha[k]+T_s E_\alpha[k]+T_s lZ_0 > 0 \tag{20}$$

and can also be formulated as:

$$lZ_0 > \frac{R}{L_d}\varepsilon_\alpha[k]-E_\alpha[k] \tag{21}$$

because the tracking error $\varepsilon_\alpha[k]$ is greater than $-Z_0$ and less than 0, $-\frac{R}{L_d}Z_0 > \frac{R}{L_d}\varepsilon_\alpha[k]$. If the following inequality is satisfied, equations (20) and (21) are satisfied:

$$\left(l+\frac{R}{L_d}\right)Z_0 > -E_\alpha[k] \tag{22}$$

Since R/L_d is positive, a stronger condition can also be proposed as lZ_0 is greater than $-E_\alpha[k]$, which means if lZ_0 is greater than the an amplitude of the back EMF $E_\alpha[k]$, the tracking error $\varepsilon_\alpha[k+1]$ is greater than the tracking error $\varepsilon_\alpha[k]$ can be satisfied when the tracking error $\varepsilon_\alpha[k]$ is less than $-Z_0$.

Thus, if the product of the QSMO gain l and the adaptive parameter Z_0 is greater than the magnitude of the extended back EMF η, condition (i) is satisfied. When the tracking error $\varepsilon_\alpha[k]$ is outside the width of the boundary layer, the state trajectory will move in the direction of the sliding surface.

In condition (ii), if the tracking error $\varepsilon_\alpha[k]$ is greater than the adaptive parameter Z_0, such that Z_a equals Z_0, the sum of tracking errors $\varepsilon[k+1] + \varepsilon[k]$ is greater than 0, which can be formulated based on equation (15) as:

$$\varepsilon_\alpha[k+1]+\varepsilon_\alpha[k]=\left(2-\frac{T_sR}{L_d}\right)\varepsilon_\alpha[k]+T_sE_\alpha[k]-T_slZ_0>0 \tag{23}$$

and can also be formulated as:

$$lZ_0 < \left(\frac{2}{T_s}-\frac{R}{L_d}\right)\varepsilon_\alpha[k]+E_\alpha[k] \tag{24}$$

because the tracking error $\varepsilon_\alpha[k]$ is greater than the adaptive parameter Z_0 and 0, $\left(\frac{2}{T_s}-\frac{R}{L_d}\right)Z_0 < \left(\frac{2}{T_s}-\frac{R}{L_d}\right)\varepsilon_\alpha[k]$.

If the following inequality is satisfied, equations (23) and (24)

are satisfied:

$$lZ_0 < \left(\frac{2}{T_s} - \frac{R}{L_d}\right) Z_0 + E_\alpha[k] \tag{25}$$

If the tracking error $\varepsilon_\alpha[k]$ is less than $-Z_0$, such that $Z_a = -Z_0$, the sum tracking errors $\varepsilon[k+1]$ and $\varepsilon[k]$ is less than 0, which can be formulated based on equation (15) as:

$$\varepsilon_\alpha[k+1]+\varepsilon_\alpha[k] = \left(2 - \frac{T_s R}{L_d}\right)\varepsilon_\alpha[k] + T_s E_\alpha[k] + T_s lZ_0 < 0 \tag{26}$$

and can also be formulated as:

$$lZ_0 < -\left(\frac{2}{T_s} - \frac{R}{L_d}\right)\varepsilon_\alpha[k] - E_\alpha[k] \tag{27}$$

because the tracking error $\varepsilon_\alpha[k]$ is less than $-Z_0 < 0$, $\left(\frac{2}{T_s} - \frac{R}{L_d}\right) Z_0 < -\left(\frac{2}{T_s} - \frac{R}{L_d}\right)\varepsilon_\alpha[k]$. If the following inequality is satisfied, equations (26) and (27) are satisfied:

$$lZ_0 < \left(\frac{2}{T_s} - \frac{R}{L_d}\right) Z_0 - E_\alpha[k] \tag{28}$$

If lZ_0 is smaller than $\left(\frac{2}{T_s} - \frac{R}{L_d}\right) Z_0 - |E_\alpha[k]|$, where $|E_\alpha[k]|$ is the amplitude of the extended back EMF η, the condition (ii) is satisfied and the change in the state trajectory between the kth and $(k+1)^{th}$

samples will also be limited.

The amplitude of the extended back EMF $|E_a[k]|$ may be related to the magnitude of the EMF η as follows:

$$|E_a[k]| = \eta^* |\sin\theta|$$

Condition (i) provides a lower boundary for the product of the gain l and the adaptive parameter Z_0; while condition (ii) provides an upper boundary for the product of the gain l and the adaptive parameter Z_0. In one example embodiment, the upper boundary is larger than the lower boundary, which can be formulated as:

$$Z_0 > \frac{2T_s L_d |E_\alpha[k]|}{2L_d - RT_s} = \frac{2|\eta|}{2L_d f_s - R} \tag{29}$$

where f_s is the sampling frequency. According to the discussion above, in order to converge the state trajectory to the sliding surface $s[k]$ from the initial state after finite time steps, the adaptive parameters Z_0 and l of the switching function satisfy the following:

$$\begin{cases} |E_\alpha[k]| < lZ_0 < \left(\frac{2}{T_s} - \frac{R}{L_d}\right) Z_0 - |E_\alpha[k]| \\ Z_0 > \frac{2|\eta|}{2L_d f_s - R} \end{cases} \tag{30}$$

In one example embodiment, based on equation (30), lZ_0 is a bounded value and is larger than the magnitude of the back EMF η in the current time sample k. When the speed ω_r increases, the magnitude of the back EMF will also increase.

The adaptive parameter Z_0 is proportional to the magnitude of the estimated back EMF $\hat{e}_{\alpha\beta}$. For example, when the magnitude of the

estimated back EMF $\hat{e}_{\alpha\beta}$ increases, the change in the back EMF between two consecutive samples will also increase; as a consequence, if the sampling frequency f_s remains the same, the tracking error $\varepsilon[k]$ will increase.

If the sampling frequency f_s increases, the sampling time T_s will decrease, and the minimum value of the adaptive parameter Z_0 will also decrease, which indicates that increasing the sampling frequency f_s will maintain the state trajectory in a smaller boundary layer and improve the tracking performance.

FIG. 2C illustrates an example embodiment of a portion of the QSMO 170 shown in FIG. 1A. As shown in FIG. 2C, a portion 170b includes the switching block 210 and sliding mode current estimator 205. It should be understood, that the QSMO 170a, shown in FIG. 2A, may include the portion 170b, and the portion 170b is illustrated to show an example embodiment of how the adaptive parameters Z_0 and l may be determined by the QSMO 170.

As shown in FIG. 2C, a back EMF magnitude estimator 230 estimates the magnitude of back EMF η using equation (5).

Based on the magnitude of back EMF η, a parameter estimator 235 calculates the adaptive parameters Z_0 and l as follows:

$$Z_0 = m\frac{2|\eta|}{2L_d f_s - R} \tag{31}$$

$$l = m\frac{n}{Z_0} \tag{32}$$

where m is a parameter used to satisfy equation (30), and can be between 1-1.6, and preferably 1.1~1.2, which means 10%~20% larger than the minimum value of Z_0. The parameter m is a testing-tuned coefficient to provide margin for the adaptive parameters Z_0 and l.

Equation (5) indicates that the magnitude of the extended back EMF η is a function of the currents i_d and i_q as well as the rotor speed

ω_r. In steady state di_q/dt can be assumed 0. Thus, if the current i_d and the speed ω_r are known, the value of η can be determined.

The back EMF magnitude estimator 230 determines the current i_d from the torque command T_{cmd}. For an IPMSM such as the motor 155, a generated electromagnetic torque T_e can be expressed as:

$$T_e = \frac{3}{2} p_o i_q [(L_d - L_q) i_d + \psi_m] \quad (33)$$

where p_o is the number of magnetic pole pairs of the motor 155. The relationship between i_d and i_q depends on the control algorithm used for the motor 155. For example, if a maximum torque per ampere (MTPA) control is used, the relationship between i_d and i_q can be obtained by taking Taylor's series expansion as follows:

$$i_d^* = \frac{(L_d - L_q)}{\psi_m} i_q^2 \quad (34)$$

Therefore, once the back EMF magnitude estimator 230 receives the torque command T_{cmd} and the estimated speed $\hat{\omega}_r^*$, the back EMF magnitude estimator 230 can determine the values of i_d and i_q using equations (33) and (34). The relationship between the command torque T_{cmd} and currents i_d and i_q can be implemented by using a look-up table or a high-order polynomial.

The back EMF magnitude estimator 230 and the parameter estimator 235 may be software executed by digital signal processor or microcontroller, for example, to implement the EMF estimations and parameter estimations. The back EMF magnitude estimator 230 and the parameter estimator 235 may be executed by the data processor 264.

FIG. 2D illustrates an example embodiment of a portion of the QSMO 170 shown in FIG. 1A. As shown in FIG. 2D, a portion 170c

includes the switching block 210 and sliding mode current estimator 205. It should be understood, that the QSMO 170a, shown in FIG. 2A may include the portion 170c and the portion 170c is illustrated to show an example embodiment of how the adaptive parameters Z_0 and l may be determined by the QSMO 170.

The portion 170c is the same as the portion 170b except the portion 170c includes a parameter LUT 240 instead of the back EMF magnitude estimator 230 and the parameter estimator 235. The parameter LUT 240 receives the torque command T_{cmd} and speed command ω_r^*. The parameter LUT 240 is a 3-D lookup table. Based on equation (30), the parameter LUT 240 generates the adaptive parameters Z_0 and l as follows:

$$Z_0 = m\frac{2|\eta|}{2L_d f_s - R} \tag{35}$$

$$l = m\frac{|\eta|}{Z_0} \tag{36}$$

In an example of the parameter LUT 240, the product of QSMO gain l and adaptive parameter Z_0 are determined based on machine speed and torque command. For example, for a constant sampling frequency, the QSMO gain l is a constant value, e.g., 8000, for 5000 Hz sampling frequency.

FIG. 2E illustrates an example embodiment of the parameter LUT 240. FIG. 2E illustrates a 3-D lookup table. Based on the torque percentage Torq_Perc and the speed command ω_r^*, the parameter LUT 240 may determine the product of QSMO gain l and adaptive parameter Z_0. In FIG. 2E, portion 2100 represents a product between 200,000 and 400,000, portion 2200 represents a product between 400,000 and 600,000, portion 2300 represents a product between 600,000 and 800,000, portion 2400 represents a product between

800,000 and 1,000,000 and portion 2500 represents a product between 1,000,000 and 1,200,000.

By using the lookup table shown in FIG. 2E or the portion in FIG. 2C, the observer parameters (Z_0 and l) are selected according to the command torque and speed, to manage both load/speed variation and machine parameter variation from given values provided by a machine manufacturer. The QSMO 170 achieves accurate position estimation without phase shift.

While FIGS. 2C-2D illustrate the back EMF estimator 230 as configured to receive the torque command T_{cmd} and the command speed ω_r^*, the percentage Torq_Perc may be directly used from the torque processor 115 instead of the torque command T_{cmd} and the command speed ω_r^*.

Double Sampling Frequency

In one example embodiment, the QSMO 170 is configured to sample the voltage command $v_{\alpha\beta}^*$ and the measured current $i_{\alpha\beta}$ at twice a frequency as the PWM frequency of the pulse width generation module 145.

For example, a PWM frequency of 6,000 Hz is a relatively low switching frequency compared with 20 kHz. For motor speeds around 3,000 RPM, the QSMO 170 with 6,000 Hz sampling frequency has good performance to limit a rotor position error within 3 electric degrees. However, for a much wider speed operation range, e.g., 5,000 RPM, and heavy load conditions, the sampling frequency is incremented.

Although the PWM frequency may be limited, high sampling frequency or control loop rate can be achieved by increasing the execution rate of the QSMO 170 and controller 102. For example, doubling the execution rate per PWM cycle improves the performance

of the QSMO 170 while maintaining the same level of switching losses. FIG. 3 illustrates a method of estimating a rotor position in a motor according to an example embodiment. More specifically, FIG. 3 illustrates a method of estimating a rotor position in a motor. The method includes obtaining a measured current for the motor, determining an estimated current using a sliding mode observer, determining a difference between the measured current and the estimated current, generating a switching control vector based on the difference and adaptive parameters of the sliding mode observer, and estimating the rotor position based on the switching control vector.

The method of FIG. 3 may be in implemented in a sensorless drive system such as the drive system 100, shown in FIG. 1.

At S310, the drive system generates a measured current. For example, with reference to FIG. 1, the current transducers 180a, 180b measure current data ia and ib, respectively, applied to the motor 155.

The converter 160 may apply a Clarke transformation or other conversion equations (e.g., certain conversion equations that are suitable are known to those of ordinary skill in the art) to convert the measured three-phase representations of current into two-phase representations of current based on the three-phase current data ia and ib from the current transducers 180a, 180b and an estimated rotor position $\hat{\theta}_{re}$ from the QSMO 170. The output of the converter 160 module (i_{sd}, i_{sq}) is coupled to the current regulator 135.

The converter 165 may apply a Park transformation or other conversion equations (e.g., certain conversion equations that are suitable are known to those of ordinary skill in the art) to convert the measured three-phase representations of current into two-phase representations of current based on the current data ia and ib from the current transducers 180a, 180b.

The QSMO 170 receives the measured currents i_{α}, i_{β} and the

voltage commands v_α^* and v_β^*.

Referring back to FIG. 3, a QSMO determines an estimated current at S320. For example, in FIG. 2A, the QSMO 170a generates estimated current value $\hat{i}_{\alpha\beta}$ ($\hat{i}_\alpha$ and $\hat{i}_\beta$) based on the voltage command $v_{\alpha\beta}^*$ (v_α^* and v_β^*) and the switching control vector $Z_{\alpha\beta}$.

At S330, the QSMO determines a difference between the measured current and the estimated current. For example, in FIG. 2A, the comparator 207 receives the measured currents i_α and i_β and the estimated current values $\hat{i}_\alpha$ and $\hat{i}_\beta$. The comparator 207 determines errors ε_α and ε_β between the measured currents i_α and i_β and the estimated currents $\hat{i}_\alpha$ and $\hat{i}_\beta$, respectively. The errors ε_α and ε_β may be differences between the measured currents i_α and i_β and the estimated currents $\hat{i}_\alpha$ and $\hat{i}_\beta$, respectively.

At S340, the QSMO generates a switching control vector based on the difference and adaptive parameters of the QSMO. For example, The switching block 210 implements a variable switching function where Z_0 is an adaptive parameter.

FIG. 2B illustrates an example embodiment of the variable switching function implemented by the switching block 21. As shown in FIG. 2B, if the tracking error $\varepsilon[k]$ is larger than Z_0, the switching block 210 outputs Z_0 as the switching control vector $Z_{\alpha\beta}$, and if the tracking error $\varepsilon[k]$ is smaller than $-Z_0$, the switching block 210 outputs $-Z_0$ as the switching control vector $Z_{\alpha\beta}$. If the tracking error $\varepsilon[k]$ is limited in a boundary layer between $-Z_0$ and Z_0, the output of saturation function will be the tracking error $\varepsilon[k]$ as the switching control vector $Z_{\alpha\beta}$. The switching block 210 outputs the switching control vector $Z_{\alpha\beta}$ to the sliding mode current estimator 205 and the low-pass filter 215.

Moreover, condition (i) provides a lower boundary for the

product of the gain l and the adaptive parameter Z_0; while condition (ii) provides an upper boundary for the product of the gain l and the adaptive parameter Z_0.

Back to FIG. 3, the QSMO estimates the rotor position based on the switching control vector at S350. For example, as shown in FIG. 2A, the position calculator 220 determines a rotor position of the motor 155 based on the estimated back EMF $\hat{e}_{\alpha\beta}$. The rotor position determined by the position calculator 220 is not compensated for the phase shift caused by the low-pass filter 215. Therefore, the phase shift compensator 225 adds a phase shift $\Delta\theta_{re}$ to the rotor position at the logic unit 227 to compensate for the phase shift. The sum of the output from the position calculator 220 and the phase shift $\Delta\theta_{re}$ is the estimated rotor position $\hat{\theta}_{re}$. Once the rotor position is estimated, the controller (e.g., 102) may control the motor based on the estimated rotor position.

Speed Aided Stabilizers

In FIG. 1A, a loop from the current regulator 135 to the inverter 150, to the QSMO 170 and back to the current regulator 135 may be referred to an inner position feedback loop. In the inner position feedback loop, QSMO 170, the controller 102 and motor 155 are coupling with each other.

The output of the QSMO 170 is the estimated rotor position $\hat{\theta}_{re}$ without any reference value, and the estimated rotor position $\hat{\theta}_{re}$ is used by the converters 140 and 160.

As described above, the pulse width generation module 145 receives the transformed voltage commands v_α^* and v_β^*, which are used to generate three phase voltages/currents for the motor 155. The QSMO 170 receives the measured currents i_α and i_β and voltage

commands v_α^* and v_β^*, as input. Thus, FIG. 1A shows a high order, nonlinear, and highly coupled system including error propagation and self-exciting oscillation.

In order to improve the drive system 100 stability and help the drive system 100 go through smoothly at a load/speed transient, the inventors propose speed aided stabilizers.

The drive system 100 is configured to implement the speed aided stabilizers. The speed aided stabilizers are based on the idea that motor rotor speed changes much slower than the position changes in medium and high speed ranges. Thus, during the time interval of each two sampling points, speed can be assumed as a constant value, and can be used to predict the position for a next sample. This predicted position for a next sample can be used as a reference to adjust the estimated position, so as to help the drive system 100 go through the transient with high accuracy.

FIGS. 4A-4C illustrate an implementation of a speed buffer for a speed aided stabilizer, according to an example embodiment. FIG. 4D illustrates a structure of the speed buffer implemented in DSP RAM.

In FIGS. 4A and 4B, the QSMO 170 determines an estimated rotor position $\hat{\theta}[n]$ for every PWM cycle having a period of $T_s[n]$. As shown in FIG. 4D, a controller 400 is shown. The controller 400 is the same as the controller 102 except the controller 400 includes a comparator 402 and a delay 405. While the comparator 402 and the delay 405 are illustrated as outside of the QSMO 170, it should be understood that the comparator 402 and the delay 405 may be implemented as a part of the QSMO 170.

The QSMO 170 supplies the estimated rotor position to the comparator 402 and the delay 405. The delay 405 delays the estimated rotor position $\hat{\theta}[n]$ and outputs the delayed rotor position to the comparator 402, which is the estimated rotor position in the previous sample. Thus, the comparator 402 determines a difference

$\Delta\hat{\theta}[n]$ between a current estimated rotor position $\hat{\theta}[n]$ and a subsequent estimated rotor position $\hat{\theta}[n+1]$. The difference $\Delta\hat{\theta}[n]$ represents a position change.

The comparator 402 and the delay 405 may be software used executed by a digital signal processor or microcontroller. For example, the comparator 402 and the delay 405 may be software executed by the data processor 264.

The difference $\Delta\hat{\theta}[n]$ and associated time period $T_s[n]$ are sent to and stored in a speed buffer 410. As shown, the speed buffer 410 may be stored in DSP RAM 415 or any other known type of tangible computer readable medium that is configured to be executed by a digital signal processor or microcontroller, for example.

Fig. 4C illustrates the speed buffer 410 in more detail. As shown, the speed buffer stores the difference $\Delta\hat{\theta}[n]$ and associated time period $T_s[n]$ and indexes the values according to n.

The buffer 410 is a rolling buffer, which means if a new position change $\Delta\hat{\theta}[n]$ is obtained, it will stored at buffer[0], and original buffer[0]~buffer[N−2] will be shift right, and become stored at buffer[1]~buffer[N−1]. Previous information stored in buffer [N−1] will be lost. N is the buffer size. The buffer size N may be determined based on the speed response, for example. If the size of the buffer 410 is small, the speed response will be fast, however, the estimated speed normally has larger oscillation. If the buffer size is large, the speed will be filtered smoothly, however, the speed response will be slower.

As shown in FIG. 4D, an output of the speed buffer 410 becomes similar to a moving average. More specifically, for each PWM cycle, the speed calculator 175 divides a sum of the differences stored in the speed buffer by a sum of the time periods. The speed calculator 175 outputs the divided result as an estimated speed $\hat{\omega}[n]$. In more

detail, an estimated speed $\hat{\omega}[n]$ based on estimated position $\hat{\theta}[n]$ can be expressed as:

$$\hat{\omega}[n] = \frac{\sum_{n-N+1}^{n} \Delta\theta[k]}{\sum_{n-N+1}^{n} T_s[k]} \quad (37)$$

When the controller 400 implements equation (37), the speed error between estimated and measured speed may be smaller than 1%.

FIGS. 5 and 6 illustrate methods of estimating a position of a rotor in a motor. The methods include determining a first estimated position of the rotor using a first algorithm, determining a second estimated position of the rotor using a second algorithm, the second algorithm being different than the first algorithm, determining a first error based on the first estimated position and the second estimated position and determining a third estimated position of the rotor based on the first error.

It should be understood that the QSMO 170 in the data processing system 101 of the drive system 100 is configured to implement the methods shown in FIGS. 5-6. Therefore, the drive system 100 includes a controller configured to, determine a first estimated position of the rotor using a first algorithm, determine a second estimated position of the rotor using a second algorithm, the second algorithm being different than the first algorithm, determine a first error based on the first estimated position and the second estimated position, and determine a third estimated position of the rotor based on the first error.

First Stabilizer

FIG. 5 illustrates a first method of stabilizing speed according to an example embodiment.

At S505, the speed stabilizer determines if drive is enabled. If drive is not enabled, the speed stabilizer sets a selected position (third estimated position) $\hat{\theta}[n]$ and a second estimated position $\hat{\theta}_2[n]$ to equal a first estimated position $\hat{\theta}_1[n]$. The first estimated position $\hat{\theta}_1[n]$ is the estimated rotor position output from the QSMO 170. The speed stabilizer returns to S505 to monitor if drive is enabled.

If drive is enabled at S505, the QSMO 170 supplies the first estimated position $\hat{\theta}_1[n]$ at S515 and the speed stabilizer determines a first error $\varepsilon[n]$ at S520. The speed stabilizer determines the first error $\varepsilon[n]$ by determining an absolute value of a difference between the second estimated position $\hat{\theta}_2[n]$ and the first estimated position $\hat{\theta}_1[n]$.

At S525, the speed stabilizer determines if the position error is smaller than the error margin E. The error margin E indicates whether the QSMO 170 is stable. If the speed stabilizer determines that the first error $\varepsilon[n]$ is less than the error margin E, then the speed stabilizer sets the selected estimated position $\hat{\theta}[n]$ as the output of the QSMO $\hat{\theta}_1[n]$, at S530. If the first error $\varepsilon[n]$ is larger than the error margin E, which means QSMO 170 is unstable or large transient occurs, then the speed stabilizer sets the selected estimated position $\hat{\theta}[n]$ as second estimated position $\hat{\theta}_2[n]$, at S535.

At S540, the controller controls the motor based on the selected estimated position $\hat{\theta}[n]$.

At S550, the speed stabilizer determines a second estimated position $\hat{\theta}_2[n]$ for a next PWM cycle based on a speed prediction algorithm. The second estimated position $\hat{\theta}_2[n]$ may be determined by the speed stabilizer as:

$$\hat{\theta}_2[n+1] = \hat{\theta}[n] + \hat{\omega}[n] \times T_s \quad (38)$$

The error margin E is a design parameter that is determined based on empirical data.

Second Stabilizer

FIG. 6 illustrates a second method of stabilizing speed according to an example embodiment. It should be understood that the QSMO 170 is configured to implement the method shown in FIG. 6.

In the method of FIG. 6, the speed stabilizer implements a position error rolling sum $\sum \varepsilon[n]$. The position error rolling sum $\sum \varepsilon[n]$ adds each sample error $\varepsilon[n]$ between $\hat{\theta}_2[n]$ and $\hat{\theta}_1[n]$ together.

In the method of FIG. 6, the position error rolling sum $\sum \varepsilon[n]$ may be referred to as the first error.

At S605, the speed stabilizer determines if drive is enabled. If drive is not enabled, the speed stabilizer sets a selected position (third estimated position) $\hat{\theta}[n]$ and the second estimated position $\hat{\theta}_2[n]$ equal the first estimated position $\hat{\theta}_1[n]$. The first estimated position $\hat{\theta}_1[n]$ is the estimated rotor position output from the QSMO 170. The speed stabilizer returns to S605 to monitor if drive is enabled.

If drive is enabled at S605, the QSMO 170 supplies the first estimated position $\hat{\theta}_1[n]$ at S615 and the speed stabilizer determines a

current error $\varepsilon[n]$ at S620. The speed stabilizer determines the current error $\varepsilon[n]$ by determining an absolute value of a difference between the second estimated position $\hat{\theta}_2[n]$ and the first estimated position $\hat{\theta}_1[n]$.

At S625, the speed stabilizer determines if the current error $\varepsilon[n]$ is smaller than an error margin for a single sample $\tilde{E}$. The error margin $\tilde{E}$ indicates whether the QSMO 170 is stable. If the speed stabilizer determines that the current error $\varepsilon[n]$ is larger than the error margin for a single sample $\tilde{E}$, then the speed stabilizer sets the selected the second estimated position $\hat{\theta}_2[n]$, at S630.

If the current error $\varepsilon[n]$ is less than error margin for a single sample $\tilde{E}$, then the speed stabilizer adds the current error $\varepsilon[n]$ to a sum of previous errors $\sum\varepsilon[n-1]$ to determine the position error rolling sum $\sum\varepsilon[n]$, at S635.

At S640, the speed stabilizer determines if the position error rolling sum $\sum\varepsilon[n]$ is smaller than the error margin E. The error margin E indicates whether the QSMO 170 is stable. If the speed stabilizer determines that the position error rolling sum $\sum\varepsilon[n]$ is less than the error margin E, then the speed stabilizer sets the selected estimated position $\hat{\theta}[n]$ as the output of the QSMO $\hat{\theta}_1[n]$, at S645. If position error rolling sum $\sum\varepsilon[n]$ is larger than the error margin E, which means QSMO 170 is unstable or large transient occurs, then the speed stabilizer sets the selected estimated position $\hat{\theta}[n]$ as

second estimated position $\hat{\theta}_2[n]$ minus the position error rolling sum $\sum \varepsilon[n]$, at S650.

At S655, the controller controls the motor based on the selected estimated position $\hat{\theta}[n]$.

At S660, the speed stabilizer determines a second estimated position $\hat{\theta}_2[n]$ for a next PWM cycle based on a speed prediction algorithm. The step S660 is the same as S550 and, thus, will not be described in greater detail for the sake of clarity.

The error margin for a single sample $\tilde{E}$ is a small value, and normally generally within 0.5 electric degree. Moreover, the error margin E may be selected as 3 electric degrees.

As described above, the inventors have discovered an adaptive Quasi-SMO (QSMO) to estimate the rotor position from the extended back electromagnetic force (EMF) quantities in an IPMSM. The QSMO parameters are adaptive to the load and rotor speed.

At least one example embodiment discloses an extended back EMF-based adaptive QSMO for rotor position estimation for a sensorless IPMSM drive. The inventors have discovered that a discrete-time sliding mode observer (DSMO) with conventional switching functions, e.g., a sign function, will keep tight regulation to force the state trajectory close to the sliding surface even when tracking error is within the width of the boundary layer. This could cause a chattering problem during steady state. To mitigate this chattering problem, and to reach a global stability as well as a bounded motion within a limited boundary layer, a switching function may be implemented. The switching function leads to a quasi-sliding mode motion of the DSMO at steady state. Since the magnitude of the extended back EMF of the IPMSM changes with both load and speed variations, the parameters allow better performance than conventional SMOs.

At least another example embodiment discloses speed aided stabilizers to improve the drive system stability and help the drive system go through smoothly at load/speed transients. The speed aided stabilizers are based on the idea that motor rotor speed changes much slower than the position changes in medium and high speed ranges. Thus, during the time interval of each two sampling points, speed can be assumed as a constant value, and can be used to predict the position for a next sample. This predicted position for a next sample can be used as a reference to adjust the estimated position, so as to help the system go through the transient with high accuracy.

Example embodiments being thus described, it will be obvious that the same may be varied in many ways. Such variations are not to be regarded as a departure from the spirit and scope of example embodiments, and all such modifications as would be obvious to one skilled in the art are intended to be included within the scope of the claims.

CLAIMS

What is claimed is:

1. A method of estimating a position of a rotor in a motor, the method comprising:

determining a first estimated position of the rotor using a first algorithm;

determining a second estimated position of the rotor using a second algorithm, the second algorithm being different than the first algorithm;

determining a first error based on the first estimated position and the second estimated position; and

determining a third estimated position of the rotor based on the first error.

2. The method of claim 1, wherein the determining a first estimated position determines the first estimated position for a same point in time as the determining a second estimated position determines the second estimated position.

3. The method of claim 2, wherein the determining a third estimated position determines the third estimated position for a point in time subsequent to the point in time associated with the determining a first estimated position and the determining a second estimated position.

4. The method of claim 1, wherein the determining a first estimated position determines the first estimated position by a sliding mode observer by comparing an estimated current to a measured current associated with the motor.

5. The method of claim 4, wherein the determining a second estimated position determines an estimated speed of the rotor, the

second estimated position being based on the speed of the rotor and the third estimated position.

6. The method of claim 1, wherein the determining a second estimated position determines an estimated speed of the rotor, the second estimated position being based on the estimated speed of the rotor.

7. The method of claim 1, wherein the determining a third estimated position includes,

determining if the first error exceeds an error margin, and

setting the third estimated position to the second estimated position if the first error exceeds the error margin.

8. The method of claim 1, wherein the determining a third estimated position includes,

determining if the first error exceeds an error margin, and

setting the third estimated position to the first estimated position if the first error is below the error margin.

9. The method of claim 1, wherein the determining a first error determines the first error as a difference between the first estimated position and the second estimated position.

10. The method of claim 1, wherein the determining a first error includes,

determining a current error between the first estimated position and the second estimated position, and

determining the first error based on the current error and a previous error.

11. The method of claim 10, wherein the determining a first estimated position determines the first estimated position by a sliding mode observer.

12. The method of claim 11, wherein the determining a second estimated position determines an estimated speed of the rotor, the second estimated position being based on the speed of the rotor.

13. The method of claim 10, wherein the determining a second estimated position determines an estimated speed of the rotor, the second estimated position being based on the speed of the rotor.

14. The method of claim 10, wherein the determining the first error based on a current error and a previous error adds the current error and the previous error.

15. The method of claim 10, wherein the determining a third estimated position includes,

determining if the first error exceeds an error margin, and

setting the third estimated position to the first estimated position if the first error is below the error margin.

16. The method of claim 10, wherein the determining a third estimated position includes,

determining if the first error exceeds an error margin, and

setting the third estimated position to the second estimated position minus the first error if the first error exceeds the error margin.

17. A drive system comprising:

a controller configured to,

determine a first estimated position of the rotor using a first algorithm,

determine a second estimated position of the rotor using a second algorithm, the second algorithm being different than the first algorithm,

determine a first error based on the first estimated position and the second estimated position, and

determine a third estimated position of the rotor based on the first error.

18. The drive system of claim 17, further comprising:

a buffer coupled to the controller and configured to store data for determining the second estimated position.

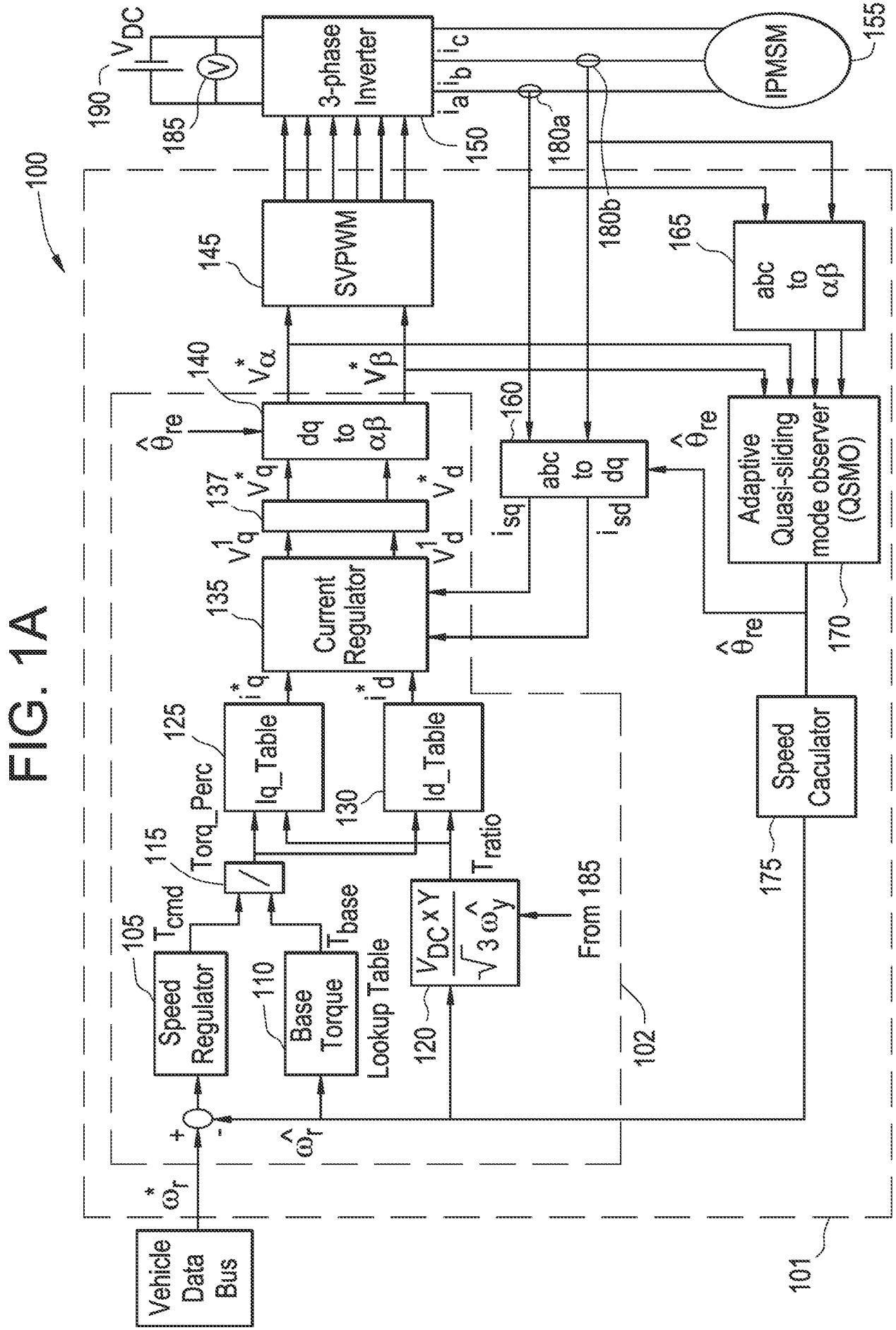

100
190
V_{DC}
185
3-phase Inverter
150
i_a i_b i_c
IPMSM
155
180a
180b
145
SVPWM
165
abc to αβ
140
$\hat{\theta}_{re}$
V^*_α
V^*_β
dq to αβ
160
abc to dq
i_{sq}
i_{sd}
Adaptive Quasi-sliding mode observer (QSMO)
137
V^*_q
V^*_d
V^1_q
V^1_d
135
Current Regulator
i^*_q
i^*_d
125
Iq_Table
Id_Table
130
115
Torq_Perc
T_{cmd}
T_{base}
T_{ratio}
105
Speed Regulator
110
Base Torque Lookup Table
$\frac{V_{DC} \times Y}{\sqrt{3}\,\hat{\omega}_y}$
From 185
120
102
$\hat{\omega}_r$
ω^*_r
Vehicle Data Bus
170
Speed Caculator
175
101

FIG. 1A

FIG. 1B

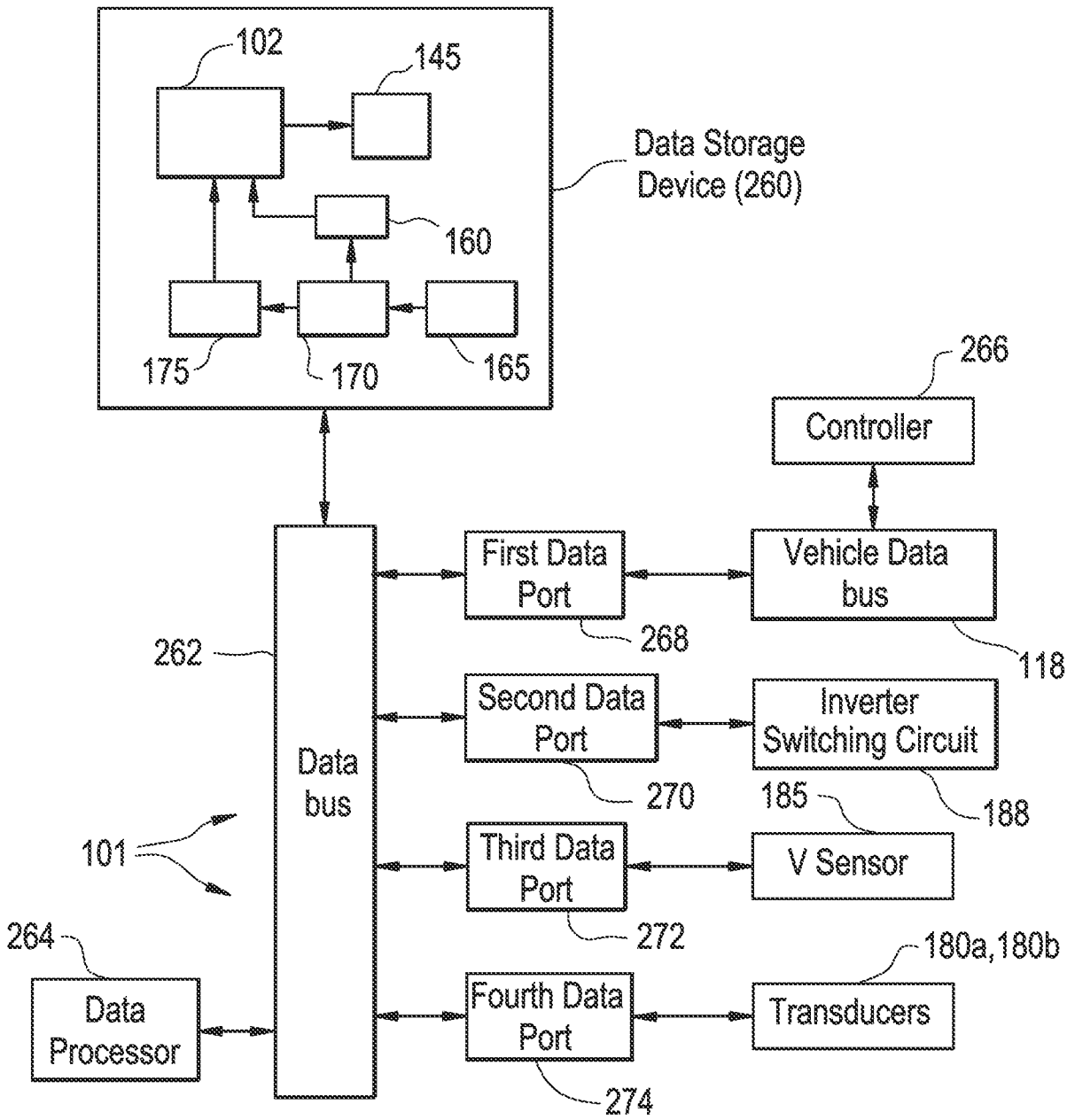

102
145
Data Storage Device (260)
160
175
170
165
266
Controller
First Data Port
Vehicle Data bus
268
118
262
Second Data Port
Inverter Switching Circuit
Data bus
270
185
188
101
Third Data Port
V Sensor
264
272
180a,180b
Data Processor
Fourth Data Port
Transducers
274

FIG. 2A

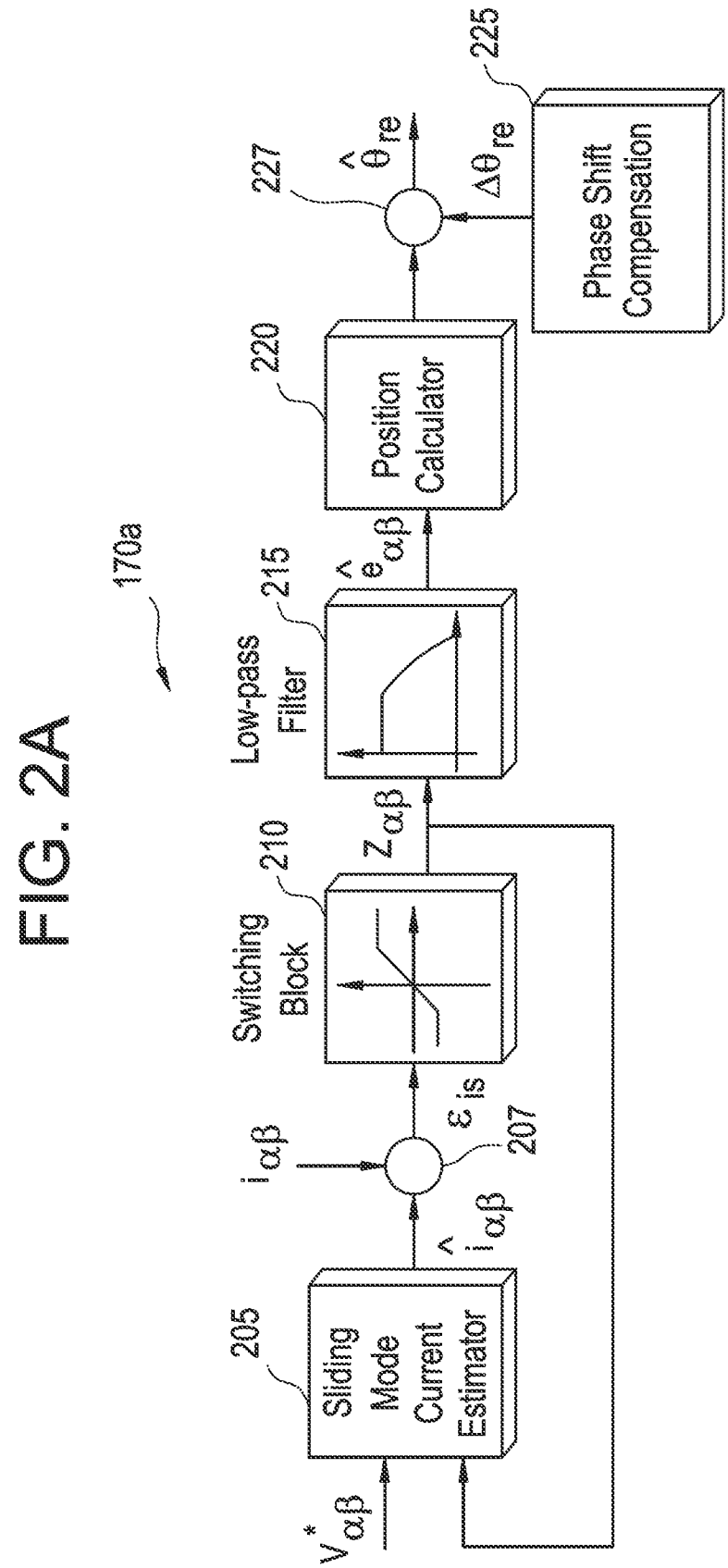

170a
205
Sliding Mode Current Estimator
$V^*_{\alpha\beta}$
$\hat{i}_{\alpha\beta}$
207
$i_{\alpha\beta}$
ε_{is}
210
Switching Block
$Z_{\alpha\beta}$
215
Low-pass Filter
$\hat{e}_{\alpha\beta}$
220
Position Calculator
227
$\hat{\theta}_{re}$
$\Delta\theta_{re}$
225
Phase Shift Compensation

FIG. 2B

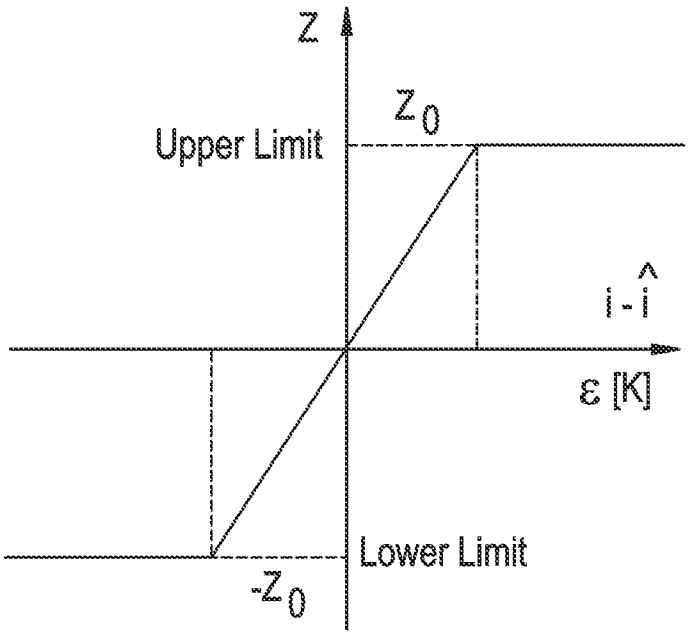


FIG. 2C

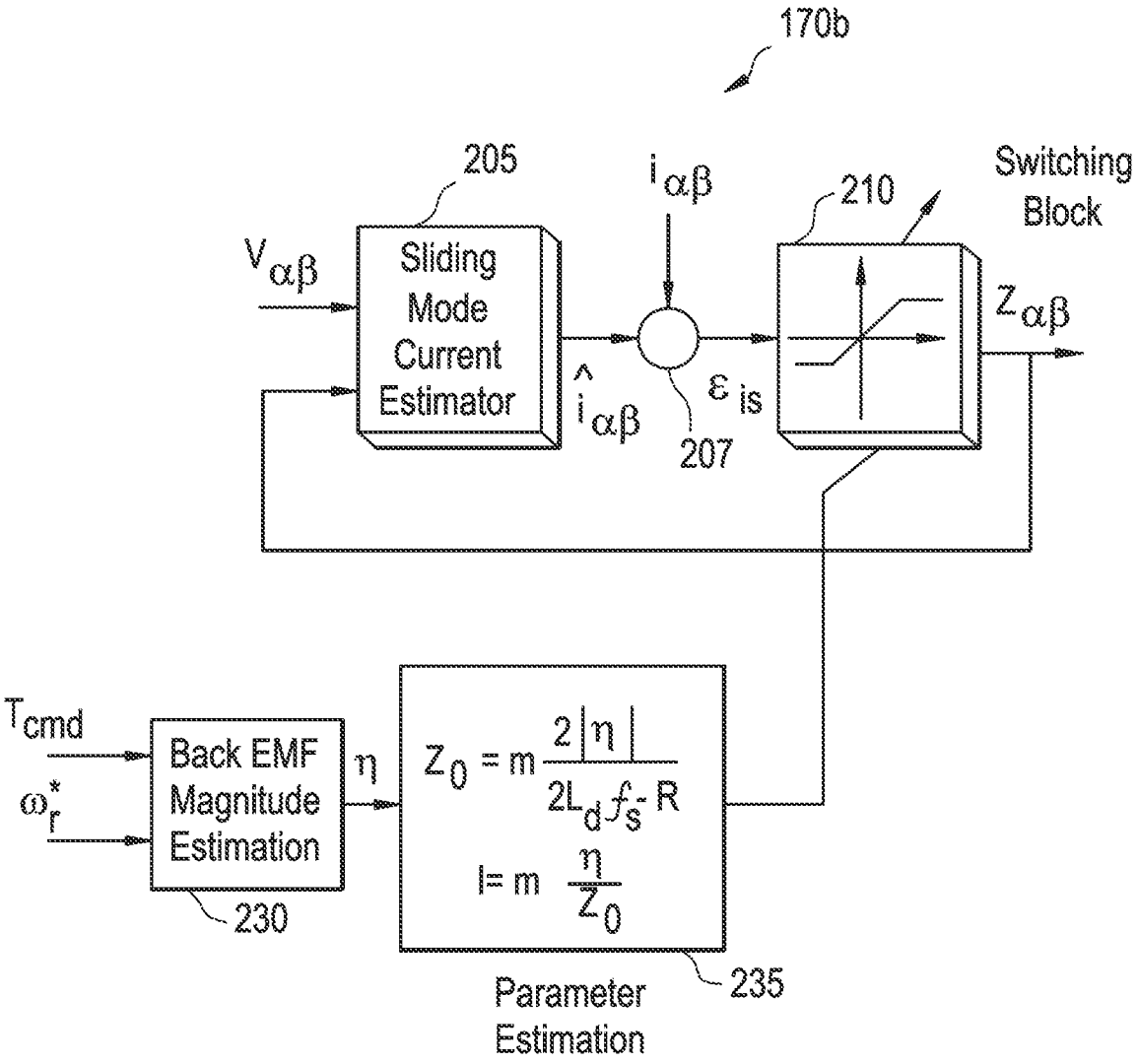

170b
205
$i_{\alpha\beta}$
210
Switching Block
$V_{\alpha\beta}$
Sliding Mode Current Estimator
$Z_{\alpha\beta}$
$\hat{i}_{\alpha\beta}$
ε_{is}
207
T_{cmd}
ω_r^*
Back EMF Magnitude Estimation
230
η
$Z_0 = m \frac{2|\eta|}{2L_d f_s - R}$
$I = m \frac{\eta}{Z_0}$
235
Parameter Estimation

FIG. 2D

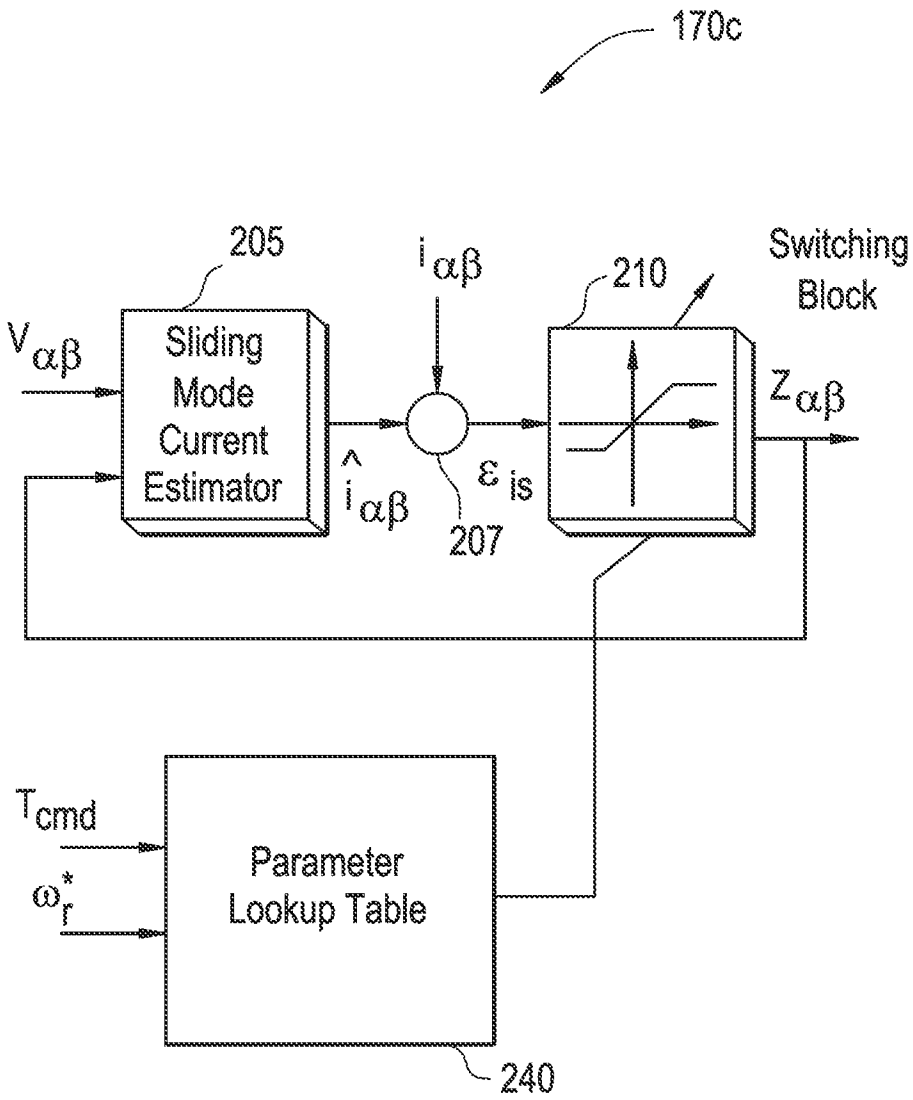

170c
205
Sliding Mode Current Estimator
$V_{\alpha\beta}$
$i_{\alpha\beta}$
$\hat{i}_{\alpha\beta}$
207
ε_{is}
210
Switching Block
$Z_{\alpha\beta}$
T_{cmd}
ω_r^*
Parameter Lookup Table
240

FIG. 2E

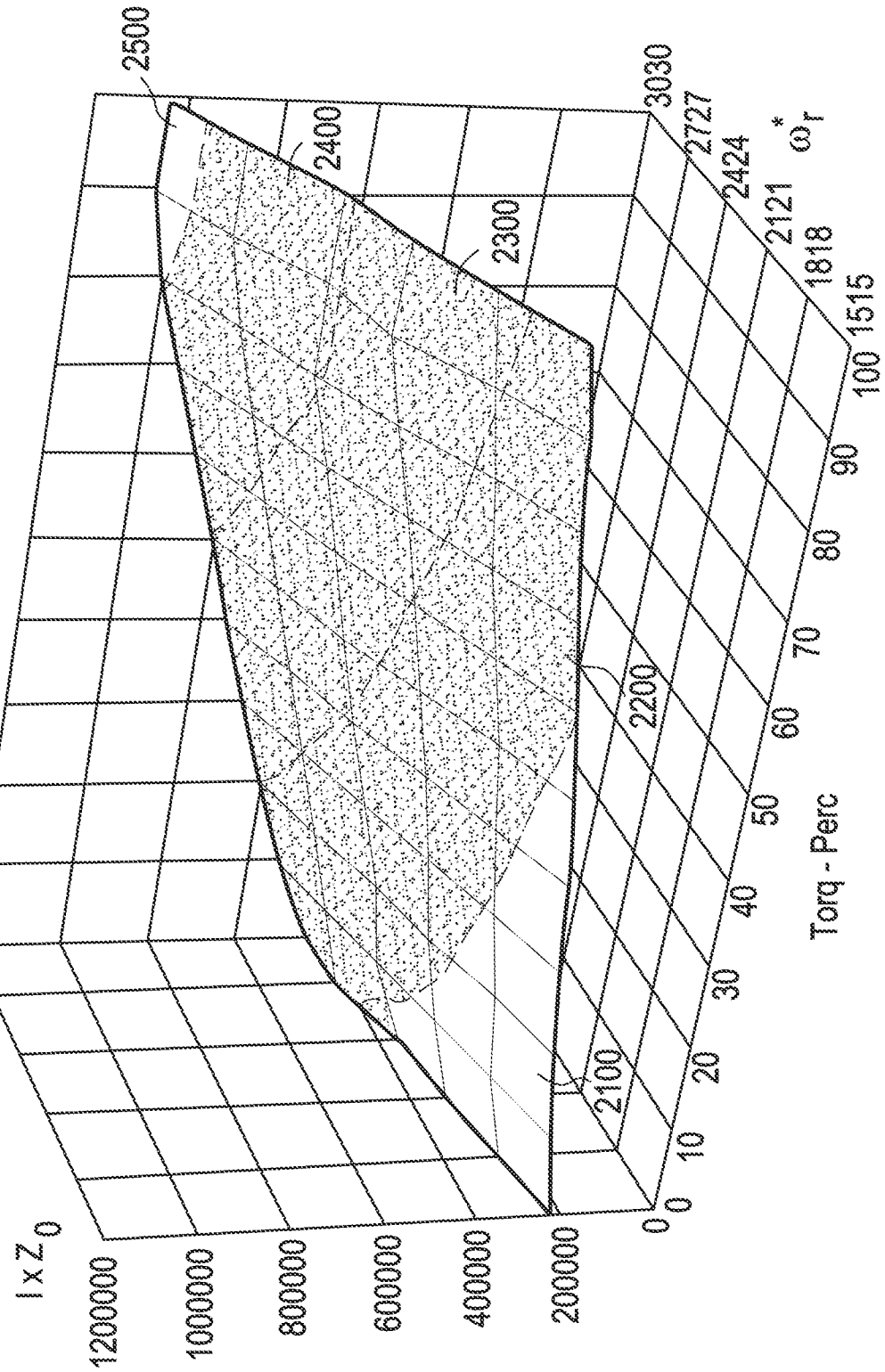
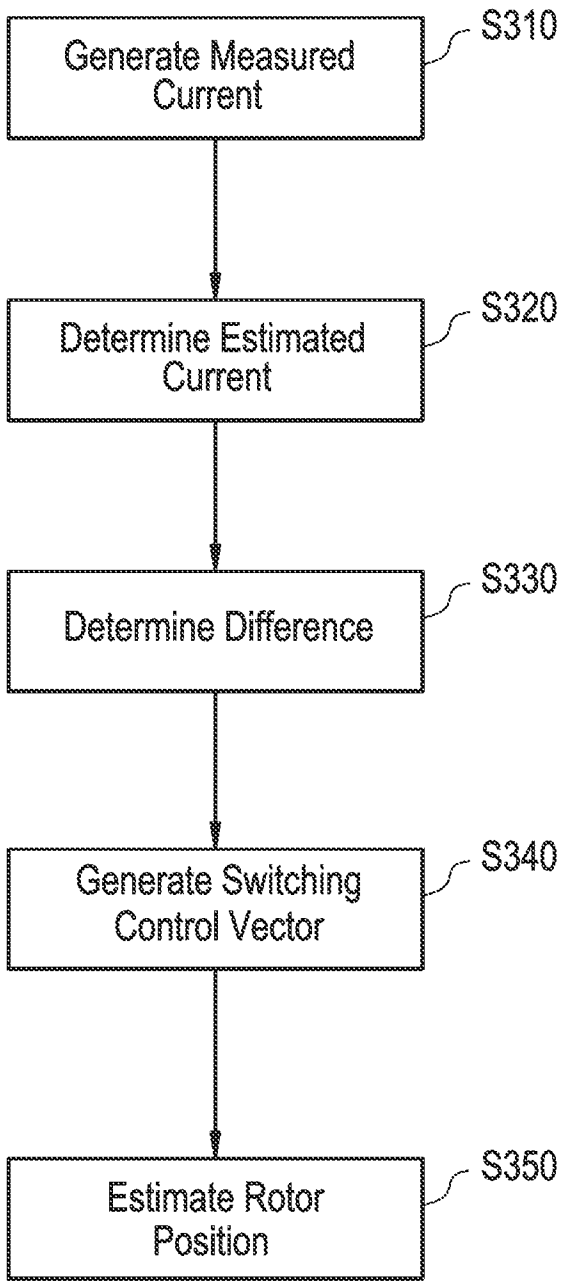

I x Z0
1200000
1000000
800000
600000
400000
200000
2500
2400
2300
2200
2100
0
10
20
30
40
50
60
70
80
90
100
Torq - Perc
3030
2727
2424
2121
1818
1515
ω*r

FIG. 3


Generate Measured Current
S310
Determine Estimated Current
S320
Determine Difference
S330
Generate Switching Control Vector
S340
Estimate Rotor Position
S350

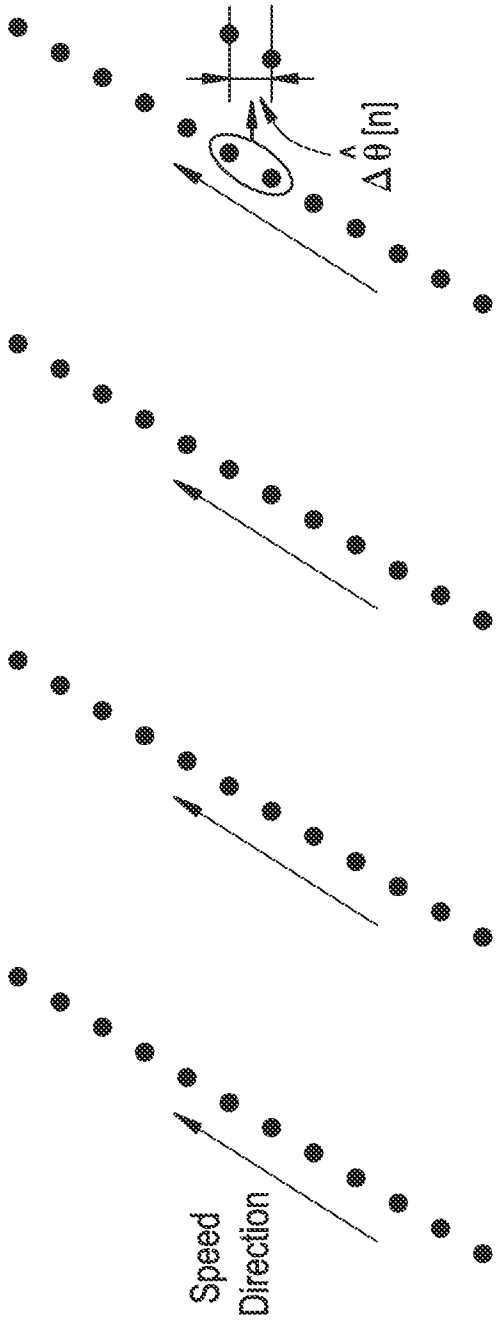


FIG. 4A

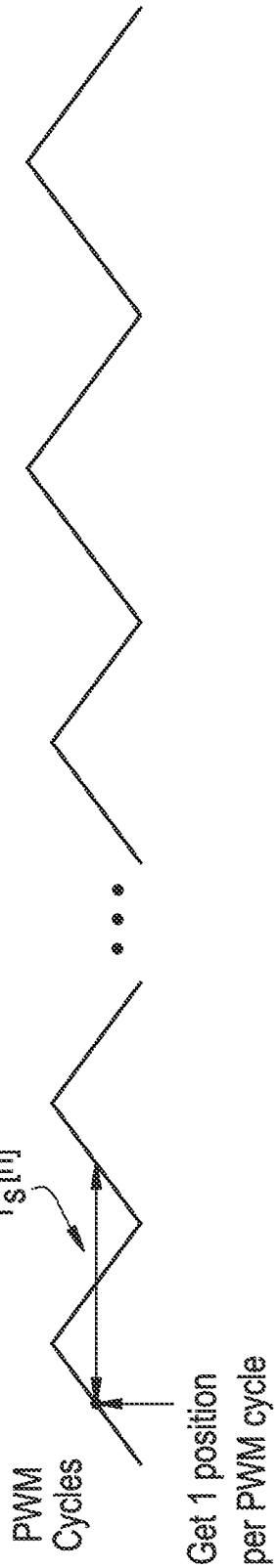


FIG. 4B

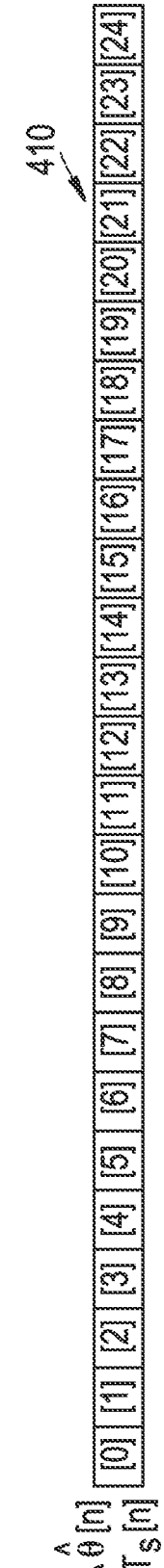


FIG. 4C

FIG. 4D

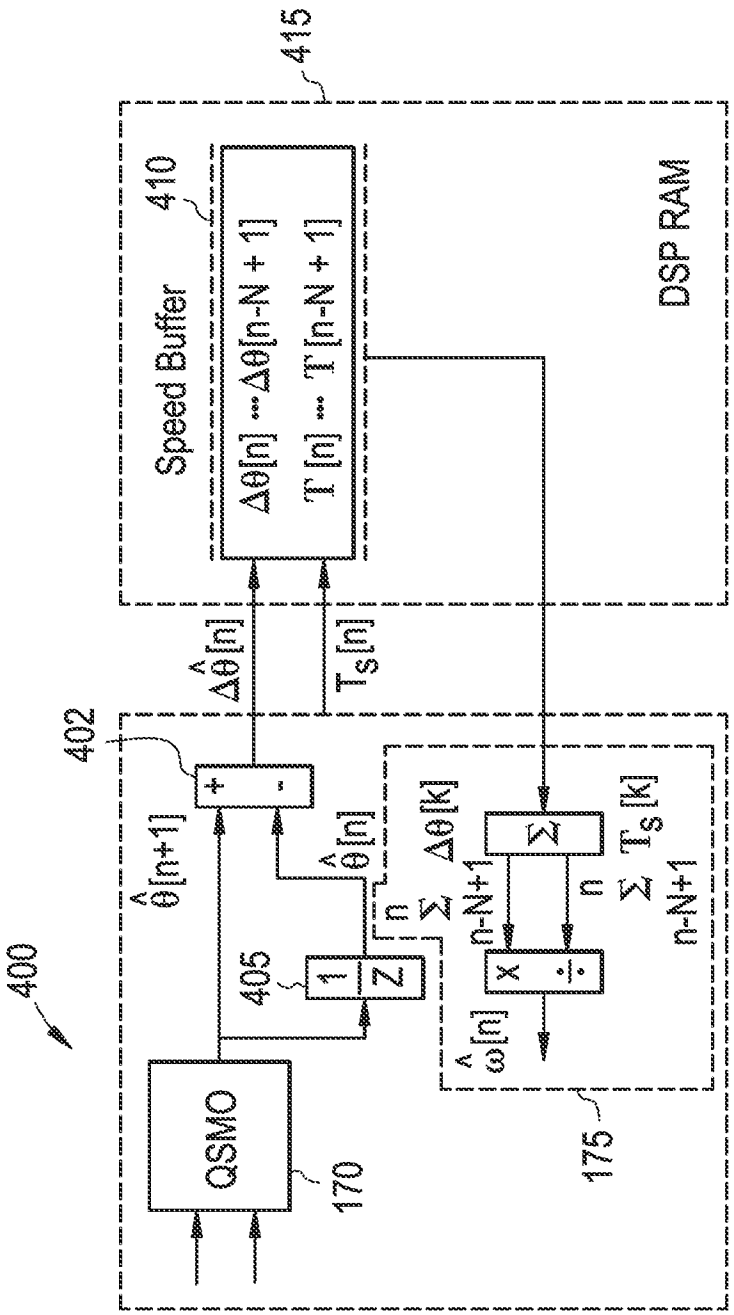

400
415
Speed Buffer
410
Δθ[n] ··· Δθ[n-N + 1]
T [n] ··· T [n-N + 1]
DSP RAM
Δθ̂[n]
Ts[n]
402
θ̂[n+1]
θ̂[n]
405
1
Z
QSMO
170
ω̂[n]
175
Σ Δθ[k]
n
n-N+1
Σ Ts[k]
n
n-N+1

FIG. 5

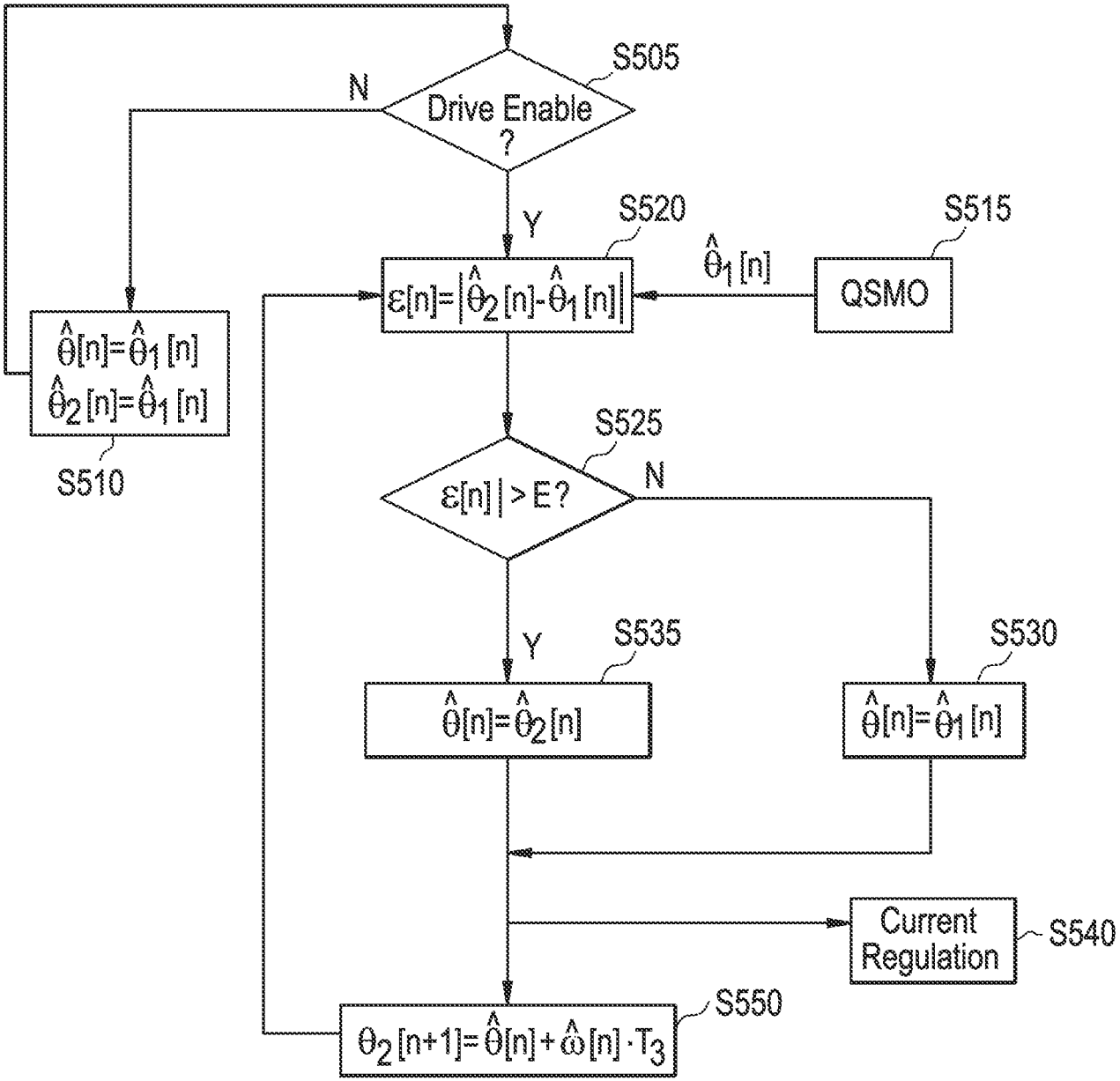
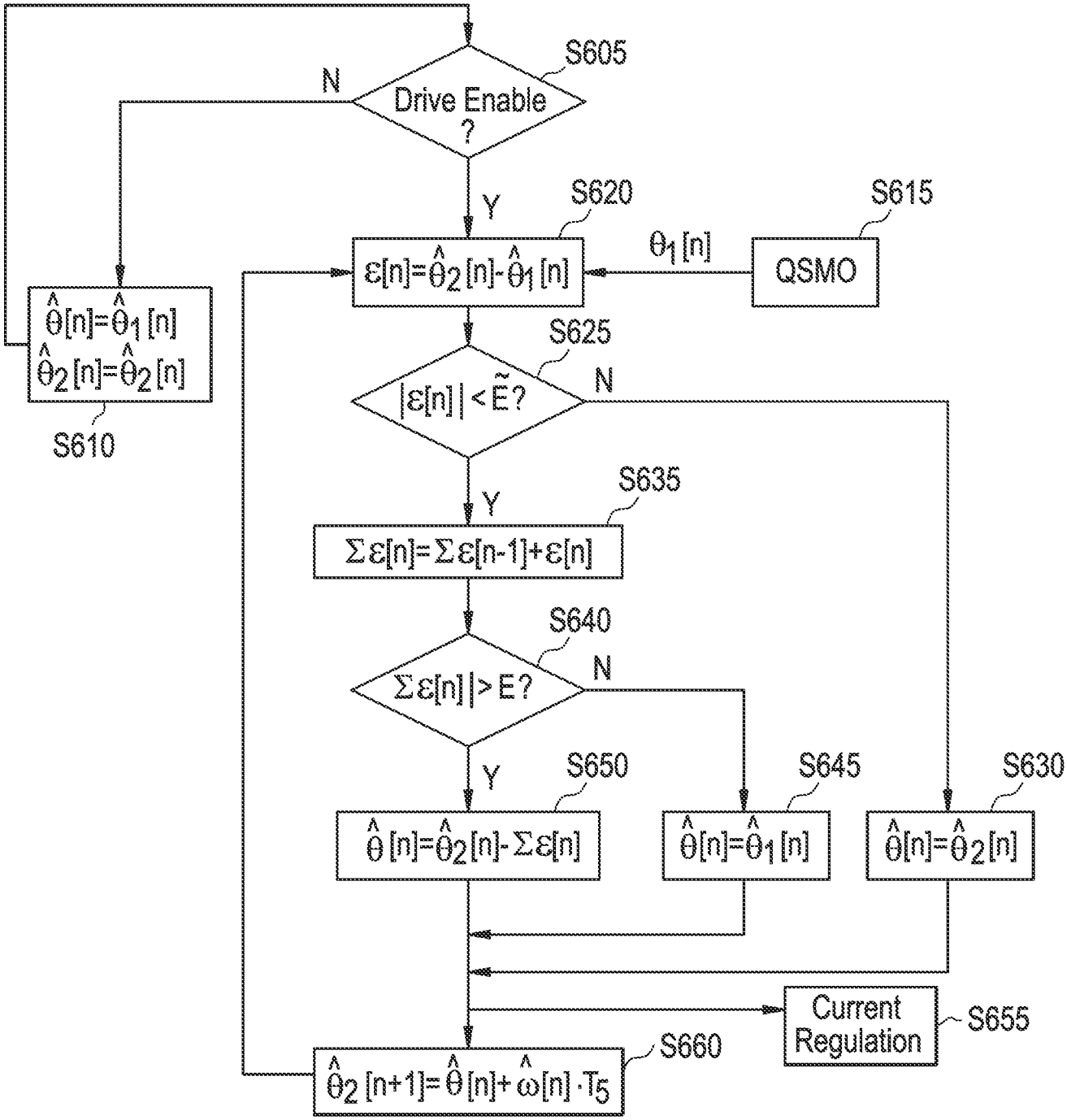

S505
Drive Enable ?
N
Y
S520
S515
$\varepsilon[n]=\left|\hat{\theta}_2[n]-\hat{\theta}_1[n]\right|$
$\hat{\theta}_1[n]$
QSMO
$\hat{\theta}[n]=\hat{\theta}_1[n]$
$\hat{\theta}_2[n]=\hat{\theta}_1[n]$
S510
S525
$|\varepsilon[n]| > E$?
N
Y
S535
S530
$\hat{\theta}[n]=\hat{\theta}_2[n]$
$\hat{\theta}[n]=\hat{\theta}_1[n]$
Current Regulation
S540
S550
$\theta_2[n+1]=\hat{\theta}[n]+\hat{\omega}[n]\cdot T_3$

FIG. 6


S605
Drive Enable ?
N
Y
S620
S615
$\varepsilon[n]=\hat{\theta}_2[n]-\hat{\theta}_1[n]$
$\theta_1[n]$
QSMO
$\hat{\theta}[n]=\hat{\theta}_1[n]$
$\hat{\theta}_2[n]=\hat{\theta}_2[n]$
S610
S625
$|\varepsilon[n]|<\tilde{E}$?
N
Y
S635
$\Sigma\varepsilon[n]=\Sigma\varepsilon[n-1]+\varepsilon[n]$
S640
$\Sigma\varepsilon[n]|>E$?
N
Y
S650
S645
S630
$\hat{\theta}[n]=\hat{\theta}_2[n]-\Sigma\varepsilon[n]$
$\hat{\theta}[n]=\hat{\theta}_1[n]$
$\hat{\theta}[n]=\hat{\theta}_2[n]$
Current Regulation
S655
S660
$\hat{\theta}_2[n+1]=\hat{\theta}[n]+\hat{\omega}[n]\cdot T_5$

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/026276

A. CLASSIFICATION OF SUBJECT MATTER
INV. H02P21/13 H02P23/12
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H02P

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	YU-LONG CUI ET AL: "Improvement of rotor position angle estimation in srd using fuzzy logic based motor model", MACHINE LEARNING AND CYBERNETICS, 2003 INTERNATIONAL CONFERENCE ON NOV. 2-5, 2003, PISCATAWAY, NJ, USA,IEEE, vol. 5, 2 November 2003 (2003-11-02), pages 2726-2730, XP010682127, DOI: 10.1109/ICMLC.2003.1260005 ISBN: 978-0-7803-7865-0	1-3,6-9, 17,18
Y	page 2728 - page 2729; figure 4	4,5
	----- -/--	

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
1 July 2013	10/07/2013

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Gusia, Sorin

2

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	LIANG QI ET AL: "A novel sliding mode observer for PMSM sensorless vector control", MECHATRONICS AND AUTOMATION (ICMA), 2011 INTERNATIONAL CONFERENCE ON, IEEE, 7 August 2011 (2011-08-07), pages 1646-1650, XP032019843, DOI: 10.1109/ICMA.2011.5985961 ISBN: 978-1-4244-8113-2 page 1646 - page 1647; figures 1,2 -----	4,5