Hall Sensor-Based Locking Electric Differential System for BLDC Motor Driven Electric Vehicles

Milad Gougani[1], Mehrdad Chapariha[1], Juri Jatskevich[1], and Ali Davoudi[2]
[1]Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada
{gougani, mehrdadc, jurij}@ece.ubc.ca
[2]Department of Electrical Engineering, University of Texas at Arlington, Arlington, USA
davoudi@uta.edu

***Abstract*—This paper presents a hardware implementation of the previously introduced locking Electric Differential System (EDS) of Electric Vehicle (EV) with two independent Brushless DC (BLDC) machine-based drives. The proposed method locks the active wheels of the vehicle as if they were operating on a common shaft. The locking algorithm is realized by processing the Hall sensor signals of the considered motors and driving them with a single set of "averaged" Hall signals. The proposed Sync-Lock Controller (SLC) is implemented digitally using a programmable integrated circuit microcontroller. First, the Hall signals undergo a layer of filtering to mitigate the errors due to hall sensor misalignment. Then, the locking algorithm is implemented by averaging the filtered Hall sensor signals. The proposed technique locks the motors directly through their corresponding magnetic fields and is shown to achieve better results as compared to a conventional speed control loop. The SLC prototype is implemented in the form of a standalone dongle-circuit that can be easily placed between the original Hall-sensors and the BLDC motor driver.**

Index Terms: Average filtering; brushless dc motors; digital control system; electric differential system; electric vehicle

I. INTRODUCTION

The electric drive train considered in the paper is conceptually illustrated in Fig. 1 [1]. The drive train consists of three main subsystems: electric propulsion, energy source, and auxiliary systems [2]. The electric propulsion subsystem, which is the focus of this paper, is comprised of a vehicle controller, power electronic converters, and electric motors driving the wheels. The control inputs from the accelerator and brake pedal along with the steering are fed to the vehicle controller. The controller in turn operates the power electronic drive which regulates the power flow between the electric motor and the energy source. The energy source subsystem includes energy storage, energy management unit, and a charger. Finally, the auxiliary subsystem manages the power steering units and the interior auxiliary devices. As shown in Fig.1, in the considered configuration there is no common mechanical shaft or transmission that can be used for locking and synchronizing the wheels – which is a mode of operation that is sometimes desirable for improved stability when driving over uneven terrain. This mode of operation is achieved electronically by the proposed Sync-Lock Control (SLC) that enables locking of the electrical differential system (EDS) which lock the driving wheels together.

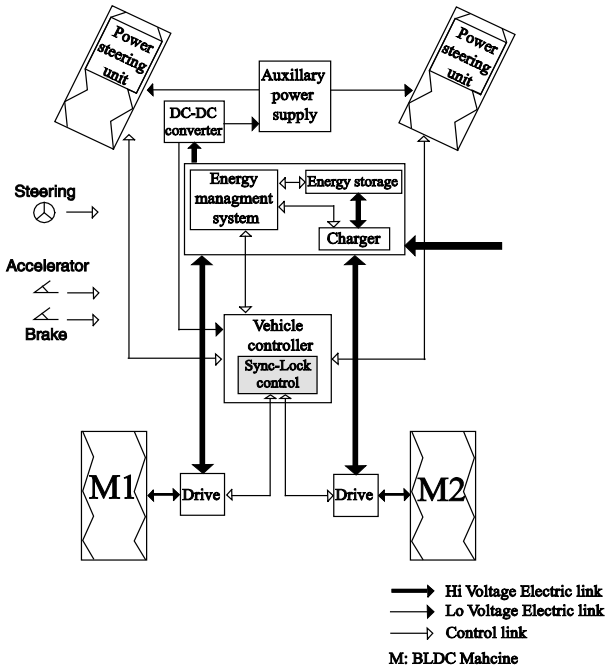

Fig. 1. Considered electric vehicle configuration.

In this paper we consider a vehicle propulsion system that is based on typical brushless dc (BLDC) motors that are voltage-source-inverter (VSI) driven and controlled by the conventional Hall sensors. The details of the electric propulsion system are shown in Fig. 2. Without loss of generality, two BLDC motors are assumed here, whereas the proposed SLC concept is readily extended for any number of driving wheels. To achieve the desired locking of the driving wheels in this paper, an innovative approach is considered that is based on filtering the original Hall-sensor signals to produce a modified set of signals which in turn is used to synchronize-lock both motors/inverters as depicted in Fig. 2. The paper presents a microcontroller implementation of the SLC for the EV and evaluates performance of the proposed technique under various conditions.

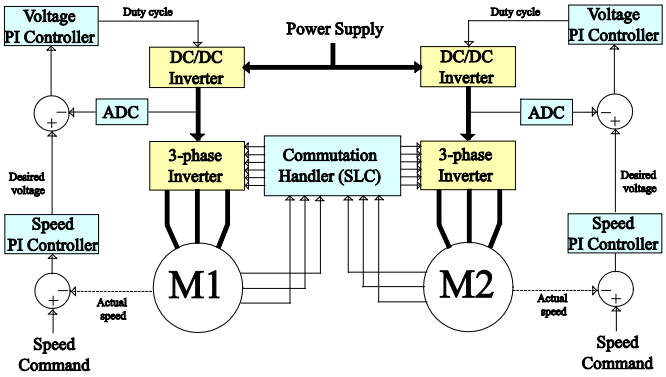

Fig. 2. Proposed two-BLDC motor-drive vehicular propulsion system.

II. BLDC Propulsion System

This paper focuses on a typical three-phase BLDC motor-inverter system, as shown in Fig. 2 where we consider two propulsion motors to demonstrate the proposed concept. The aforementioned filtering and locking algorithm (SLC) is then applied directly to the original Hall sensor signals to produce a modified set of signals that is used to the drive both inverters depicted in Fig. 2.

Conventionally, speed of a BLDC motor is controlled via an external speed control loop. The Hall sensors supply information about the position of the rotor and speed. If the speed of the motor drops because of an increase in load torque, the speed error increases, this in turn results in an increased voltage. The two motors therefore can be coupled with identical reference speeds on the control systems [3]. The mechanical subsystem (i.e. vehicle body and the ground) is the link between the two motors. This conventional approach results in soft locking of the EDS, which does not enforce the same position of the rotors/wheels [1]. In this method, the load experienced by one motor is only slightly felt by the other motor due to the change in the speed of the entire vehicle and the running resistance. The proposed locking strategy is different from a soft lock approach based on a PI controller as the two motors are locked internally by coupling the respective magnetic fields through Hall sensor signals.

In a typical design configuration of BLDC machines the Hall sensors are mounted on a printed circuit board (PCB) attached to the rear end of the motor. For a two pole machine, the Hall sensors must ideally be placed exactly 120 degrees apart to produce the control signals necessary for the standard 120-degree switching logic to control the six-step voltage-source inverter [4], [5], [6]. If this is true, then the Hall sensor signals will be apart by exactly 120 electrical degrees. Although this is a common assumption in most literature sources, this condition is difficult to achieve in practice particularly in many mass-produced low-cost vehicular-type motors due to manufacturing tolerances. Therefore, to utilize the Hall sensor signals for control purpose, these signals have to be filtered first [7].

III. Proposed Sync-Lock Controller

The proposed Sync-Lock Control strategy synchronizes the BLDC motors via the corresponding Hall sensor signals as depicted in Fig. 2. The Sync-Lock technique is based on the idea of driving both motors with a set of "averaged" filtered signals thereby locking speed and relative angle of the motors. However, to apply this locking technique effectively, the errors due to the Hall sensor misalignment must be removed first. The filter [7] is considered here to mitigate inaccurate positioning of the Hall sensors. Following is a detail description of both the filtering and averaging steps involved in the proposed controller.

A. Filtering Hall-Sensor Signals

To illustrate the result of low precision Hall sensor positioning that may be common in low-cost vehicular-type BLDC motors, the measure phase currents our sample motor with the parameters summarized in the Appendix is shown in Fig. 3. Here, the angular- and time-duration between each subsequent Hall sensor transition is denoted by $\theta(n)$ and $\tau(n)$, respectively, and n denotes the interval number. In case of a BLDC motor with ideally-placed Hall sensors, the angular duration $\theta(n)$ between subsequent Hall sensor transitions should be 60 degrees for all n, which is clearly not true in Fig. 3. Also, in steady state operation, the time intervals $\tau(n)$ should be equal.

The frequency content of the sequence $\tau(n)$, introduced in Fig.3, can be evaluated by using the discrete-time Fourier series (DTFS) [8], so that the sequence can be written as:

$$\tau(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \tag{1}$$

where the Fourier coefficients $\{c_k\}$, $k = 0,1,...,N-1$, provide the description of $\tau(n)$ in the frequency domain. In our case, the signal $\tau(n)$ has one zero-frequency component and two components with frequencies of $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ radians per sample.

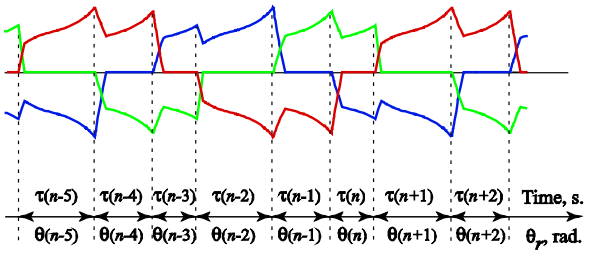

Fig. 3. Stator currents with unequal conduction intervals due to Hall-sensor misalignment.

The purpose of filtering scheme considered in this Section is to make the sequence of the time intervals follow that of a motor with perfectly positioned Hall sensors. Hence, the filter must remove $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ harmonics from the sequence $\tau(n)$. This can be achieved with selection of linear filters that generally have the following form:

$$\bar{\tau}(n)=\sum_{m=1}^{M} b_m \tau(n-m), \tag{2}$$

where M is the order of the filter corresponding to the number of time intervals taken into account, and b_m is the weighting coefficient that depends on a particular filter realization.

Different linear filters were proposed in [7] including 3-step averaging; 6-step averaging; linear extrapolating-plus-averaging; and quadratic extrapolating-plus-averaging. Without loss of generality the basic 3-step averaging is used here, whereas the proposed algorithm is readily extended for any of the higher order filters. The final 3-step filter considered here has the following equation:

$$\bar{\tau}_{a3}(n)=\frac{1}{3}\left(\tau(n-1)+\tau(n-2)+\tau(n-3)\right). \tag{3}$$

B. *Averaging Hall-Sensor Signals*

To better understand how to average the Hall-sensor signals for locking, it is instructive to consider the diagram depicted in Fig. 4. Here we assume that Hall signals coming from each motor have been already filtered and are displaced by exactly 120 electrical degrees relative to each other. Combining all three outputs in each motor produces a square wave (see Fig. 4, top first and second signals) with a period equal to one-third of a Hall-sensor period, which is equal to 60 electrical degrees. Here, the angle φ denotes a possible delay or advance between the rotors of the BLDC Motor 1 and Motor 2 [9].

The SLC is based on constructing one set of Hall signals by appropriately modifying (averaging) the signals from actual

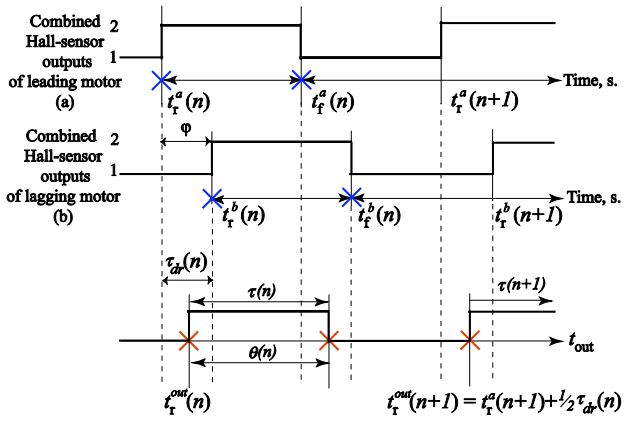

Fig. 4. Proposed averaging of the Hall sensor signals for SLC.

sensors $H_{M1}\{1,2,3\}$ and $H_{M2}\{1,2,3\}$. The averaging is done by first finding the rising edge offset term which is

$$\tau_{dr}(n)=t_r^b(n)-t_r^a(n). \tag{4}$$

Once the offset value $\tau_{dr}(n)$ is established, the actual timing for commutating the inverter transistors for the next cycle can be found as follows:

$$t_r^{out}(n+1)=t_r^a(n+1)+(1/2)\tau_{dr}(n), \tag{5}$$

where $t_r^a(n)$, assumed leading motor switching time, is defined as the reference switching time of the system. A similar approach is used for falling edges of the signals resulting in the period of the average signal:

$$\tau(n+1)=\frac{1}{2}\left(\left(t_f^a(n)-t_r^a(n)\right)+\left(t_f^b(n)-t_r^b(n)\right)\right). \tag{6}$$

IV. Implementation of Sync-Lock Controller

Implementation of the proposed controller is done in two steps. First, the Hall signals are passed through the filter to produce a set of Hall signals placed exactly 120 electrical degrees apart. Next, the filtered Hall signals are averaged and fed to the inverters of both motors thereby locking the speed and angles of the motors.

The computational resources required to implement the controller is dictated by the complexity of the algorithm. Although powerful CPUs and DSPs are readily available and can significantly simplify the development process, it is essential to design a straightforward, robust and computationally efficient implementation that requires minimum computational resources. Therefore, the filtering and averaging algorithms are based on availability of three timers and no external hardware interrupts.

A. *Filtering Hall-Sensor Signals*

The proposed filtering algorithm is implemented using software interrupt service routines (ISR). Using this method, switching of the Hall sensors triggers the input software ISR, at which time all the necessary calculations (instructions) are done inside the microcontroller. With dedicated timers for the rising and falling edges, continuous operation of the drive is enabled by resetting the internal time counter of the microcontroller back to zero at either rising or falling edges of input signals. Hence, the time intervals $\tau(n)$ are readily available simply as the timer counts between the rising and falling edges of the Hall-sensor signals. To switch the transistors when the filter is enabled, the software output ISR has to be invoked at a particular time to provide the inverter with the modified Hall signals. This time of the next switching may be expressed as

$$t_{next\_sw}=\bar{t}(n)+\bar{\tau}(n)=t_{out}(n+1), \tag{7}$$

where $\bar{\tau}(n)$ denotes the averaged time interval as calculated using (3), and $\bar{t}(n)$ is the so-called reference switching time as defined in [7]. Also, here $t_{out}(n+1)$ refers to the time when the modified output Hall signals will be switched.

Denoting the most-recent calling of the input ISR by $t_{in}(n)$, the time of the next output ISR can be expressed as

$$t_{out}(n+1) = t_{in}(n) + \tau^{corr}(n), \tag{8}$$

where $\tau^{corr}(n)$ is the appropriate correction term. To understand how the correction time $\tau^{corr}(n)$ can be calculated, it is instructive to consider Fig. 5. Here, the bottom time axis depicts the input interrupts that are triggered by the actual Hall sensor signals, $t_{in}(n)$. The scheduled output software interrupts for the modified switching signals are depicted on the top axis, $t_{out}(n)$. Assuming a certain reference time $\bar{t}(n)$ and a given $\bar{\tau}(n)$, the correction term may also be calculated using (7)–(8) as

$$\tau^{corr}(n) = t_{out}(n+1) - t_{in}(n) = \bar{t}(n) + \bar{\tau}(n) - t_{in}(n). \tag{9}$$

Computation of (9) requires knowledge of the reference time. This time may be obtained by averaging the switching times of the three phases as depicted in Fig. 5;

$$\bar{t}(n) = \frac{1}{3}\left(t(n) + t'(n) + t''(n)\right). \tag{10}$$

Here, $t(n)$ is time of the last switching of the input Hall signal, and $t'(n)$ and $t''(n)$ are the times extrapolated from the two preceding input Hall signal transition times, as follows:

$$\begin{aligned} t'(n) &= t_{in}(n-1) + \bar{\tau}(n), \\ t''(n) &= t_{in}(n-2) + 2\bar{\tau}(n). \end{aligned} \tag{11}$$

As can be seen in Fig. 5, the most recent input interrupt has occurred at $t_{in}(n)$. The last two input interrupts have occurred at $t_{in}(n-1)$ and $t_{in}(n-2)$ respectively. These times are used to calculate the extrapolated terms according to (11), and then the reference time according to (10). Combining the results, the reference time can be calculated in terms of input interrupts only as:

$$\bar{t}(n) = \frac{1}{3}\left(t_{in}(n) + t_{in}(n-1) + t_{in}(n-2)\right) + \bar{\tau}(n). \tag{12}$$

Since $t_{in}(n-1)$ and $t_{in}(n-2)$ refer to the previous input interrupt times, they can be expressed as

$$t_{in}(n-1) = t_{in}(n) - \tau(n-1), \tag{13}$$

$$t_{in}(n-2) = t_{in}(n) - \tau(n-2) - \tau(n-1). \tag{14}$$

Combining (12)–(14) and (9), the correction term may be represented as

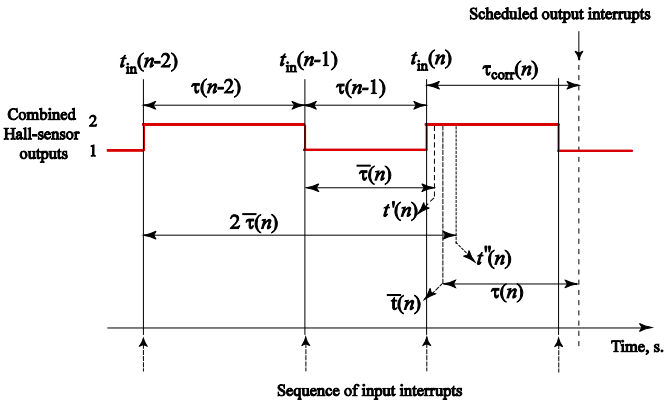

Fig. 5. Timing of the input and output Hall signal transitions using proposed filtering.

$$\tau^{corr}(n) = \frac{1}{3}\left(-2\tau(n-1) - \tau(n-2)\right) + 2\bar{\tau}(n). \tag{15}$$

This correction term $\tau^{corr}(n)$ can now be used for whole range of different filters presented in [7] and [10] by substituting the relevant expression for $\bar{\tau}(n)$. In this case, the average period given by (3) is substituted resulting in

$$\tau_{a3}^{corr}(n) = \frac{1}{3}\left(\tau(n-2) + 2\tau(n-3)\right). \tag{16}$$

Therefore, the third order filter can be implemented by simply using (8) and (16).

B. Averaging Hall-Sensor Signals

The averaging algorithm is also based on the software ISR triggered by rising and falling edges of input signals. As before, each rising and falling edge ISR has a dedicated timer which is used to save the last two time periods between the edges as shown in Fig. 6. With the latest time interval between interrupts denoted by τ_0, and the following one defined as τ_1, the output interrupt is scheduled when $\tau_0 > \tau_1$, which is to be triggered

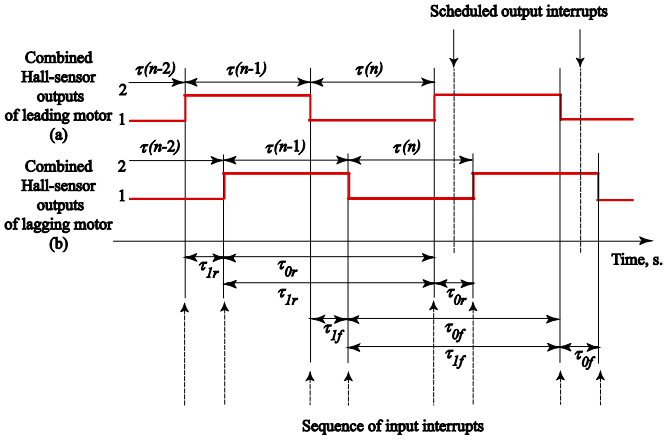

Fig. 6. Timing of the input and output Hall signal transitions using the proposed averaging.

after $\tau_1/2$ seconds. The same logic is used for averaging both rising and falling edges of the signals. It should be noted that in the proposed averaging technique there is no need to differentiate between the leading and lagging motor, which makes this approach different from a master-slave configuration.

C. Integration of Proposed Techniques

It is important to integrate the proposed filtering and averaging techniques efficiently and minimize computational resources as much as possible. Therefore, the proposed techniques are integrated using only two software ISRs with dedicated timers. One ISR and timer is devoted to the rising edges of the input signals, and another ISR and timer pair is used for the falling edges of the input signals. Finally, the third timer is utilized for scheduling the output interrupt for both the rising and falling edges. The proposed algorithm is best understood by considering the rising and falling edges of the Hall signals separately. The output rising edge interrupt is scheduled at the input falling edge when $\tau_{0f} > \tau_{1f}$ to be triggered at $\tau_{corr}(n+2) + \tau_{dr}/2$. Here, τ_{dr} is the offset between two signals defined at rising edge when $\tau_{0r} < \tau_{1r}$ as

$$\tau_{dr} = \tau_{0r} + \left(\tau(n) - \tau_{corr}(n)\right)_a - \left(\tau(n) - \tau_{corr}(n)\right)_b. \quad (17)$$

Subscript "a" in the equation reflects the actual and corrected period difference of leading motor and subscript "b" is used to indicate the period difference of lagging motor. It should be noted that either Motor 1 or Motor 2 could be the leading or lagging motor.

Similar to the rising edge interrupt, output falling edge interrupt is scheduled at the input rising edge when $\tau_{0r} > \tau_{1r}$ to be triggered at $\tau_{corr}(n+3) + \tau_{df}/2$. Here, τ_{df} is the offset between two signals defined at falling edge when $\tau_{0f} < \tau_{1f}$ as

$$\tau_{df} = \tau_{0f} + \left(\tau(n+1) - \tau_{corr}(n+1)\right)_a - \left(\tau(n+1) - \tau_{corr}(n+1)\right)_b. \quad (18)$$

The offset between two filtered signals is implemented by taking into account the fact that the filtered signal could be located before or after the actual input interrupt. The relative position of the filtered and actual signal depends on many conditions, e.g. relative position errors, whether the motor is decelerating or accelerating, etc.

Finally in both the rising and falling edge interrupts when the output ISR is invoked, the next state of the Hall sensors is predicted according to the existing state and the direction of rotation. After that, the output Hall signals will be changed to switch the inverters of the motors into their next topological state. With the proposed algorithm, the closest two phases of the motors are averaged and locked together rather than matching phases. This is desirable as two motors are locked together at any point during operation regardless of their relative positions.

D. Hardware Prototype

The proposed Sync-Lock Controller has been realized on a basic programmable integrated circuit microcontroller (dsPIC30f2020). This and similar microcontrollers are often used in many inexpensive BLDC drive systems. The filter defined by (8) and (16) together with averaging algorithm defined by (4)-(6) were programmed on the microcontroller according to the implementation approach summarized in previous Section and depicted in Figs. 5 and 6. In the implementation considered by the authors, the prototype SLC dongle can be powered either from the dc bus of the BLDC motor drivers or directly from the dc supply that feeds the Hall sensors of the motor. The input and output ports of the microcontroller are also protected against accidental over-voltage. The microcontroller can be re-programmed with different filters through the data programming port, as well as enabled or disabled using the manual switch. The Hall sensors are simply connected with the BLDC driver through the input and output ports provided on the board, thus enabling modification of the Hall sensor signals according to the proposed filtering and averaging methodologies.

V. Case Studies

To demonstrate operation of the proposed SLC dongle, a pair of BLDC motor-drives with the parameters summarized in the Appendix is considered. In the following studies, the motors are supplied with $V_{dc} = 30V$ and operate in an open loop control. To emulate driving different conditions, each motor is loaded by a dc machine dynamometer.

A. Sync-Lock Controller Enabling Transient

To illustrate transient behavior of the proposed controller, the motors are initially assumed to operate in steady state with 0.45 Nm on Motor 1 and 0.60 Nm on Motor 2, running at 1830 rpm and 1800 rpm, respectively. Then, the SLC is enabled. Following this change, both motors undergo a transient in the measured currents and speed as depicted in Fig. 7. This transient is due to the initial angle difference of the motors' shafts, which causes one of the motors to accelerate and the other motor to decelerate to achieve the locking. As can be seen in Fig. 7, after the locking, the currents are automatically adjusted to run both motors at the same identical speed while satisfying their respective loads that are different.

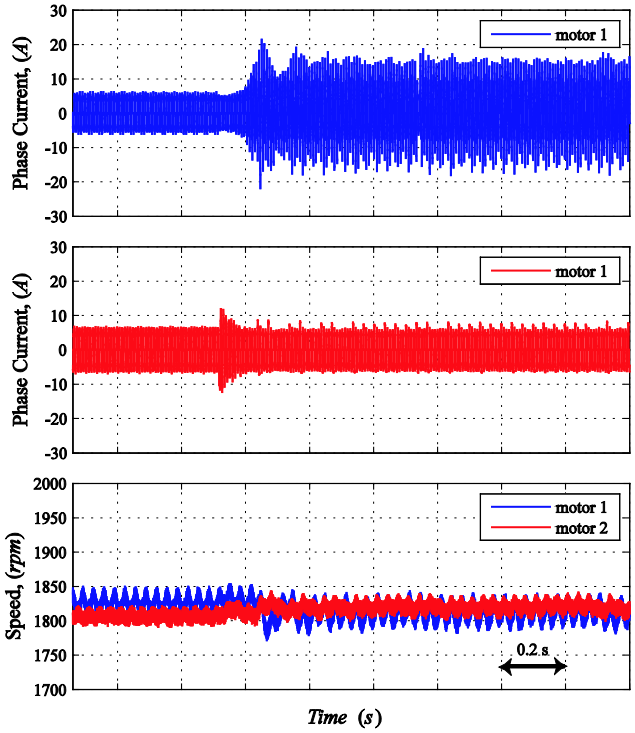

Fig. 7. Measured phase current and speed transient response of both motors due to enabling the proposed SLC.

B. *Transient due to Change in Load*

The performance of the locking mechanism is evaluated further by subjecting the system to a load disturbance. In the following study, both motors are assumed running at 1825 rpm while each driving a load of about 0.5 Nm. Then, the load on Motor 2 is increased as shown in Figs. 8 and 9 (first subplot). The resulting response of the motors' speed with the SLC disabled and enabled have been recorded and are approximately aligned in time and are also shown in Figs. 8 and 9 (second subplot), respectively. As it can be seen in Fig. 8, when the SLC is disabled, the increase in load on one of the motors does not couple to the response of the other motor. However, when the SLC is enabled, the motors remain locked and their speed in affected by changes in the load of any one of the motors.

Continuing from the previous experiment, the load on Motor 2 is stepped down while the motors are locked. As shown in Fig. 10, both motors remain synchronized and speed up to accommodate corresponding loads. The system speed is determined by average of the speeds of two motors with the slower motor carrying higher load as the reference (which is Motor 1 in this case). However, with the proposed controller, there is no one master or slave motor always dictating the system response or speed.

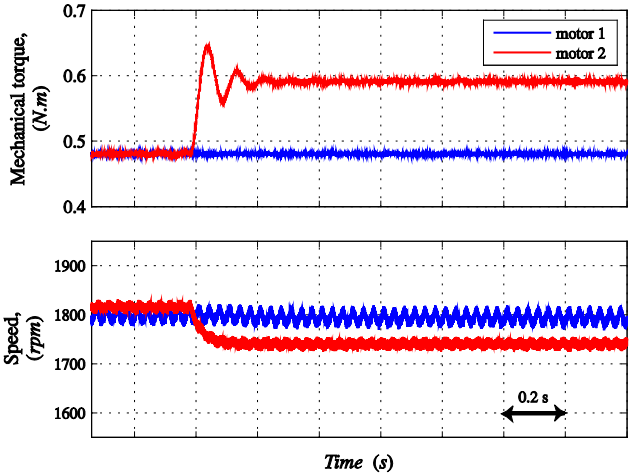

Fig. 8. Measured mechanical torque and speed transient response due to load change with proposed SLC disabled.

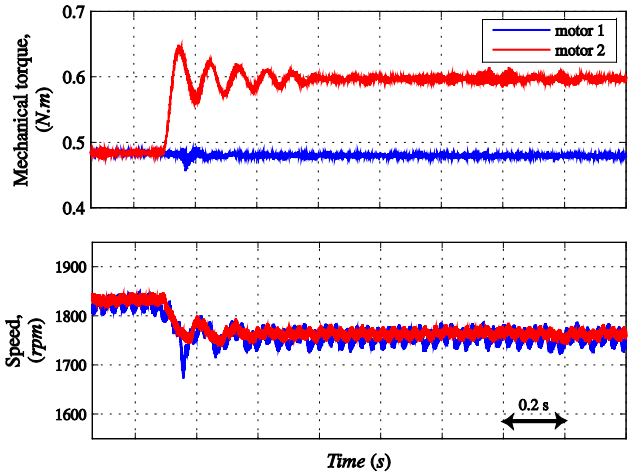

Fig. 9. Measured mechanical torque and speed transient response due to load change with proposed SLC enabled.

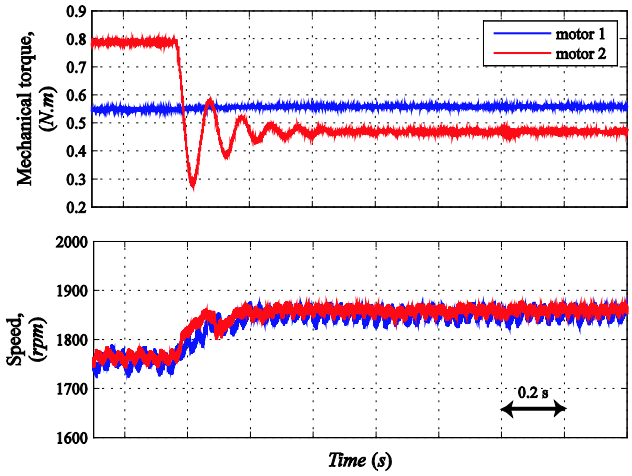

Fig. 10. Measured mechanical torque and speed transient response due to load change with proposed SLC enabled.

VI. CONCLUSION

The stability of vehicles under certain uneven terrain and/or driving conditions can be improved by forcing the wheels to turn at the same speed and angle regardless of the available traction under individual wheels. An innovative approach for synchronizing the vehicular driving wheels, thus emulating the locking electrical differential system (EDS), was recently proposed for electric vehicles (EV) with independent brushless DC (BLDC) machine-based wheel drives. As low-cost BLDC motors commonly have inaccurate Hall sensors, the new approach also filters the Hall sensor signals. Therefore desired locking is achieved by filtering and averaging the Hall sensor signals and applying the averaged switching intervals to both propulsion motors. This paper extends the previous work and proposes a robust and computationally efficient algorithm that can be implemented on a basic microcontroller-based hardware. The implementation is based on input/output interrupts. Due to its simplicity and effective implementation, the proposed algorithm can be readily applied to a variety of EVs that use BLDC motors and drivers. The performance of the proposed locking algorithm has been demonstrated using BLDC motors (with Hall-sensor positioning errors) and is shown to be efficient and robust in steady state and transients studies.

APPENDIX

BLDC PROPULSION MOTOR PARAMETERS:

Arrow Precision Motor Co., LTD., Model 86EMB3S98F, 36 VDC, 210 W, 2000 rpm, 8 poles, $r_s = 0.12\Omega$, $L_s = 0.375mH$, $\lambda_m^{'} = 22mV.s$, combined inertia $J = 12 \cdot 10^{-4} N.m.s^2$, and back EMF harmonics coefficients $K_3 = 0.01$, $K_5 = -0.059$, $K_7 = 0.025$.

REFERENCES

[1] M. Gougani, M. Chapariha, J. Jatskevich, "Locking electric differential for brushless DC machine-based electric vehicle with independent wheel drives," *Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE* , pp.1-6, 6-9 Sept. 2011.

[2] M. Ehsani, Y. Gao, S. E. Gay, and A. Emadi. *Modern Electric Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design*. CRC Press, Boca Raton, FL, 2005.

[3] B. Tabbache, A. Kheloui, N. Hanini, "An electric differential system for a two-wheel mobile plat-form using direct torque control with adaptive flux and speed observers," *Power Electronics, Electrical Drives, Automation and Motion, 2008. SPEEDAM 2008. International Symposium on* , vol., no., pp.550-556, 11-13 June 2008

[4] S. D. Sudhoff, P. C. Krause, "Operation Modes of the Brushless DC Motor with a 120º Inverter," *IEEE Transactions on Energy Conversion,* Vol. 5, No. 3, pp. 558-564, 1990.

[5] P. Pillay, R. Krishnan, "Modeling, simulation, and analysis of permanent-magnet motor drives. Part II. The brushless DC motor drive," *IEEE Trans. Industry Applications,* Vol. 25, Iss. 2, pp. 274–279, Mar.-Apr. 1989.

[6] W. Brown, "Brushless DC Motor Control Made Easy", Microchip Technology Inc., 2002 [Online]. Available: www.microchip.com

[7] N. Samoylenko, H. Qiang, J. Jatskevich, "Dynamic Performance of Brushless DC Motors with Unbalanced Hall Sensors," IEEE Transactions on Energy Conversion, Vol. 23, Iss. 3, pp. 752–763, Sep. 2008.

[8] J. G. Proakis, D. G. Manolakis, *Digital Signal Processing,* Prentice Hall, Upper Saddle River, NJ, 1996, p. 248.

[9] P. C. Krause, O. Wasynczuk, S. D. Sudhoff. *Analysis of Electric Machinery and Drive Systems.* IEEE Press, Piscataway, NJ, 2002.

[10] N. Samoylenko, Q. Han, J. Jatskevich, "Balancing hall-effect signals in low-precision brushless DC motors," *In Proc. IEEE Applied Power Electronics Conference,* Feb. 28 – Mar. 2, 2007, Anaheim CA, USA, pp. 606–611.