An Adaptive Interleaving Algorithm for Multi-converter Systems

Hao Tu, Yuhua Du, Hui Yu, Srdjan Lukic
FREEDM Systems Center
North Carolina State University
Raleigh, NC, USA
{htu, ydu7, hyu11, smlukic}@ncsu.edu

Peter Volgyesi, Mary Metelko, Abhishek Dubey, Gabor Karsai
Institute for Software Integrated Systems
Vanderbilt University
Nashiville, TN, USA
{peter.volgyesi, mary.a.metelko, abhishek.dubey, gabor.karsai}@vanderbilt.edu

***Abstract*—To integrate DC distributed generation (DG) with micro-source into the existing AC grid, a DC distribution bus can be used to couple on-site photovoltaics (PV), battery energy storage systems (BESS), and DC loads. If the converters connected to the DC bus are interleaved, the DC bus capacitor size could be minimized. In this paper, we propose an interleaving algorithm for multi-converter systems to minimize the current harmonics at switching frequency on the DC bus. The proposed algorithm is implemented using Resilient Information Architecture Platform for Smart Grid (RIAPS) platform. Hardware-in-the-Loop (HIL) simulation results based on Opal-RT are presented to validate its performance. The influence of synchronization frequency on the proposed algorithm are also considered.**

I. Introduction

With the increasing attention paid to the environmental issues and the availability of fossil fuels, distributed generation with micro-source becomes an attractive option for the future grid. A lot of DGs, such as PV and BESS, are DC source in nature [1], [2]. Conventionally, PV and BESS are connected to the existing AC grid using power electronics inverters [3], [4]. This could possibly increase the losses and lower the system efficiency since each source requires its own inverter. If all the DC DGs and loads are connected through a common DC distribution bus before they are interfaced to the AC grid, the number of conversion stages could be reduced and the system efficiency could be improved.

As shown in Fig.1, different DC sources and loads are coupled to a single DC distribution bus before they are connected to the AC grid. This is similar to the concept of hybrid AC/DC microgrid as in [5]–[7]. To maintain a stable DC bus voltage, a large capacitor is needed, which increases the cost and volume of the system. To minimize the size of DC bus capacitor, interleaving technique could be applied.

A vast literature has studied the interleaving of different phases in a single converter [8]–[10]. Interleaving is achieved by phase shifting their carriers from each other by $2\pi/N$. This is possible because the operating conditions of different phases are identical and all the phases are controlled by a single central controller. In [11], an interleaving algorithm for multi-terminal DC-DC converter systems working in asymmetric conditions is proposed. Asymmetric condition means different terminals (input stages) of a single converter working in different operating conditions caused by different input voltage, inductor value and current. Compared to $2\pi/N$ phase shift, this algorithm eliminates the current harmonics at switching frequency even when the input stages are not balanced. However, such algorithm considers only the case of three input stages. Also, the implementation in [11] is realized by a central controller. This is not possible for the system in Fig. 1 as each converter is independently controlled by a local controller. In [12], a global interleaving method for inverters are proposed. However, the synchronization of different inverters is based on RS485 communication which has a limited length. When the converters are geographically distant, the method becomes impractical. Also, all the computation in [12] including solving an optimization problem is done by the inverter's local controller, which could overload the local controller.

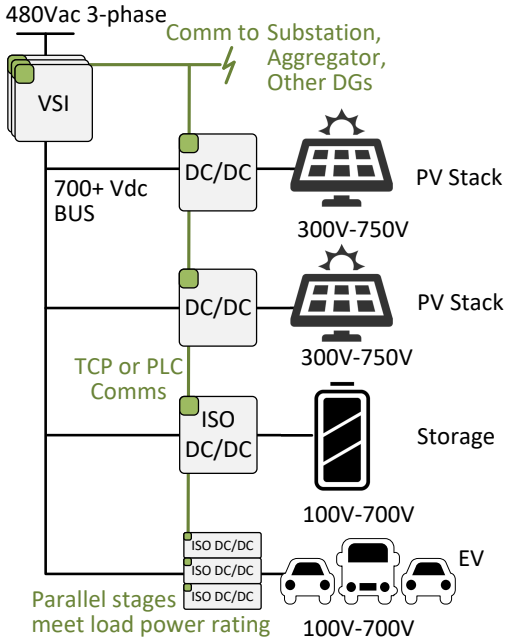


Fig. 1: DC sources coupled with a common DC Bus

To overcome the above issues, in this paper we first extend the existing interleaving algorithm from three input stages of single converter in [11] to a more general case which can be applied to N independent converters. Second, this paper details the implementation of the proposed interleaving algorithm in the RIAPS platform. The implementation requires no central unit to calculate the phase delay for the converters. The resul-

978-1-5386-6705-7/18/$31.00 © 2018 IEEE

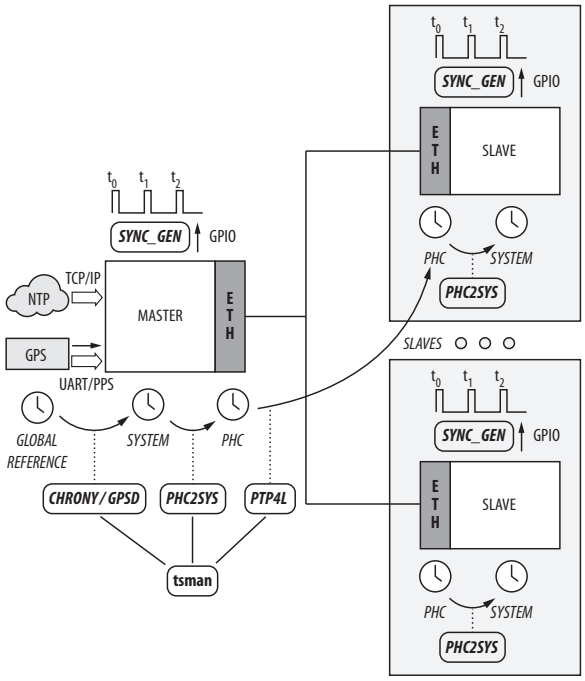


Fig. 2: End-to-end time synchronization in a RIAPS application

tant system is more practical and resilient. The RIAPS platform was introduced in [13] as a distributed control platform. Several distributed control algorithm has been implemented using the platform [14], [15]. While these papers demonstrate the distributed computation capability of the RIAPS platform, none of them require time synchronization. In this paper, we will demonstrate that the synchronization capability of the RIAPS platform is accurate for interleaving multi-converter systems.

The rest of paper is arranged as follow: section II gives an introduction to the RIAPS platform's synchronization mechanism. Section III introduces the proposed interleaving algorithm. Section IV describes the implementation of interleaving algorithm in RIAPS platform. The hardware-in-the-loop(HIL) simulation results are present in Section V. Section VI concludes the paper.

II. RIAPS Platform and its Synchronization Capability

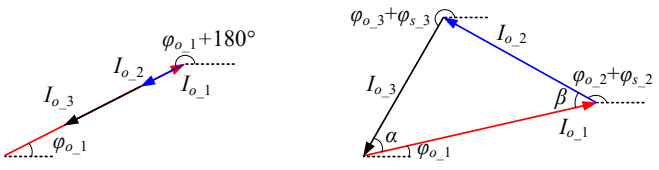
The RIAPS platform implements a complete end-to-end time synchronization architecture with various options for the reference clock(s) and supports a wide range of deployment configurations. The ultimate goal of the time synchronization infrastructure is to align each node's system clock (`CLOCK_REALTIME` POSIX clock) to a common reference in the RIAPS cluster. The platform integrates several key technology components, such as GPS, NTP, PTP—orchestrated with a RIAPS-specific configuration management tool (`tsman`) to achieve this task with the best accuracy for a given deployment configuration. Our recent paper [16] describes the architecture in detail and presents quantitative results achieved in a real-life BeagleBone-Black(BBB) deployment. It demonstrates an end-to-end time synchronization performance within 1 us across all nodes.

The current paper builds on these results and extend the time synchronization boundary beyond the system clock of the BBB nodes. The time synchronization architecture for the multi-converter system is shown in Figure 2. A single time-master node is synchronized to an external time reference. This can be a GPS receiver if unobstructed access is available to the sky, or NTP if the cluster is connected to the Internet, or—in worst case—using the master node's own clock as the absolute reference for the entire cluster. Master to slave time synchronization is achieved by the highly accurate Precision Time Protocol (PTP). This technology relies on LAN (Ethernet) connectivity and hardware time stamping capability at each node. Due to its dependence on a separate hardware timer embedded in the LAN interface, the time synchronization path includes extra steps for aligning the system clock and PTP hardware clocks (PHC) at each node.

While most steps are implemented by third-party open-source software components, we added a custom element to the existing infrastructure (`sync_gen`) for providing easy-to-integrate and robust time synchronization services for external hardware. The tool is capable of generating extremely accurate synchronization pulses on selected BBB GPIO header pins at a wide-range of frequencies (up to 2kHz) with configurable pulse-widths. The most important property of the generated pulses is their well-defined alignment on the globally established time-scale, thus the pulses are generated at the same time instants at all nodes. We made significant effort to minimize the jitter and the bias in the timing of the pulse signals, such as:

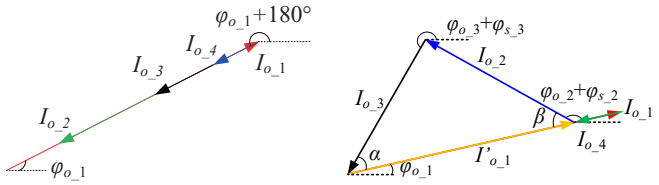
- Instead of relying on user-space libraries (e.g. *libsoc*) and kernel services (*sysfs*), the service uses direct memory-mapped I/O access for driving the GPIO pins. This approach resulted in one magnitude faster response times but requires superuser-level access and ties the current implementation to the BBB hardware platform.
- The RIAPS platform runs on Linux with the RT_PREEMPT patch for minimizing the jitter of user space real-time tasks. The `sync_gen` service is scheduled with real-time policy (`SCHED_FIFO`) at a highest priority than most kernel threads
- For the actual timing of the GPIO signal, we do not rely on the accuracy of sleep services (e.g. $clock\_nanosleep$). Instead, well before the pulse signal is due, the process wakes-up and uses a busy-wait loop (continuously checking the system time) to find the best moment to assert the GPIO signal. This approach can create a significant load on the system—especially if being used at high pulse frequencies—but provides two magnitudes of improvement in the timing of the pulses.

We evaluated the performance of the generated time synchronization pulses using 500 Hz pulse frequency and 10 us



(a) $\vec{I}_{sum} \neq 0$ (b) $\vec{I}_{sum} = 0$

Fig. 3: Three converters interleaving case



(a) $\vec{I}_{sum} \neq 0$ (b) $\vec{I}_{sum} = 0$

Fig. 4: Four converters interleaving case

pulse width. Based on a dataset of 40,000 pulses, we observed a 2.84 us bias (mean error) with a standard deviation of 0.33 us. The maximum error (very rare outlier) was 10.93 us. Note, that we conducted the measurements on a single node by capturing the system time before and after asserting the GPIO pin. Then, we used the mid-point of these two time stamps for estimating the actual time of the rising edge. Our previous work [16] already demonstrated the sub-microsecond alignment of the system clocks across the nodes. The jitter of the external synchronization pulses is in the same range, thus—if the measured 2-3 us bias is compensated for—we can synchronize external components with sub-microsecond accuracy to the system.

III. Improved Interleaving Algorithm for Multi-Converter Systems

The interleaving algorithm for three and four converters are presented in this section. For each DC/DC converter in Fig.1, its output current consists of a desired DC component and undesired high-frequency components. Analysis in the frequency domain reveals that the high-frequency components are mostly fundamental frequency (switching frequency) component and its harmonics [11]. The fundamental frequency component has the largest magnitude as well as the lowest frequency. Thus, it has the greatest influence on DC bus current and voltage ripple. If the overall fundamental component can be minimized, smaller ripple on DC bus can be achieved.

Using Fourier transform, we can extract the fundamental frequency component for each converter's output current as a vector $\vec{I}_{o\_i}$ $(i = 1, 2, 3..., n)$. The overall fundamental component is determined by $\vec{I}_{sum} = \sum_{i=1}^{n} \vec{I}_{o\_i}$.

A. Interleaving algorithm for $n = 3$

For a three-converter system, if the three vectors $\vec{I}_{o\_1}$, $\vec{I}_{o\_2}$ and $\vec{I}_{o\_3}$ could form a triangle with proper interleaving, the overall fundamental frequency component can be eliminated, as $\vec{I}_{sum} = 0$. First, we arrange the current vectors $\vec{I}_{o\_1}$, $\vec{I}_{o\_2}$ and $\vec{I}_{o\_3}$ according to their magnitudes in descending order, as $|\vec{I}_{o\_1}| \geq |\vec{I}_{o\_2}| \geq |\vec{I}_{o\_3}|$. Then, we check whether it is analytically feasible for the three vectors to form a triangle by comparing $|\vec{I}_{o\_1}|$ with $(|\vec{I}_{o\_2}| + |\vec{I}_{o\_3}|)$. If $|\vec{I}_{o\_1}| \geq (|\vec{I}_{o\_2}| + |\vec{I}_{o\_3}|)$, the three vectors can not form a triangle. To minimize the overall fundamental components, $\vec{I}_{o\_2}$ and $\vec{I}_{o\_3}$ should be set in the opposite direction of $\vec{I}_{o\_1}$, as shown in Fig.3a. Otherwise, the three vectors can form a triangle with proper phase shift, as shown in Fig.3b. If we keep $\vec{I}_{o\_1}$ unchanged, optimal interleaving angles for $\vec{I}_{o\_2}$ and $\vec{I}_{o\_3}$ could be calculated as:

$$\varphi_{s\_2} = \pi - \varphi_{o\_1} + \varphi_{o\_2} - \beta \tag{1}$$

$$\varphi_{s\_3} = \pi - \varphi_{o\_1} + \varphi_{o\_3} + \alpha \tag{2}$$

where ϕ_{o_i} $(i = 1, 2, 3)$ is the original phase angle of each fundamental component. Angles α and β can be calculated by $|\vec{I}_{o\_1}|$, $|\vec{I}_{o\_2}|$ and $|\vec{I}_{o\_3}|$ using *Law of Cosines*,

$$\cos\beta = \frac{|\vec{I}_{o\_1}|^2 + |\vec{I}_{o\_2}|^2 - |\vec{I}_{o\_3}|^2}{2|\vec{I}_{o\_1}||\vec{I}_{o\_2}|} \tag{3}$$

$$\cos\alpha = \frac{|\vec{I}_{o\_1}|^2 + |\vec{I}_{o\_3}|^2 - |\vec{I}_{o\_2}|^2}{2|\vec{I}_{o\_1}||\vec{I}_{o\_3}|} \tag{4}$$

B. Interleaving algorithm for $n = 4$

Above technique can be extended to a four-converter system. Geometrically speaking, a quadrilateral case is more complex compared to a triangle case and does not have a unique solution. Triangle is still favored for the four vectors case. Current vectors of four converters $\vec{I}_{o\_1}$, $\vec{I}_{o\_2}$, $\vec{I}_{o\_3}$ and $\vec{I}_{o\_4}$ are arranged according to their magnitudes in descending order, as $|\vec{I}_{o\_1}| \geq |\vec{I}_{o\_2}| \geq |\vec{I}_{o\_3}| \geq |\vec{I}_{o\_4}|$. First we check whether it is feasible for the four vectors to form a triangle by comparing $|\vec{I}_{o\_1}|$ with $\sum_{i=2}^{4} |\vec{I}_{o\_i}|$. If $|\vec{I}_{o\_1}| \geq \sum_{i=2}^{4} |\vec{I}_{o\_i}|$, the four vectors can not form a triangle. $\vec{I}_{o\_2}$, $\vec{I}_{o\_3}$ and $\vec{I}_{o\_4}$ should be set in the opposite direction of $\vec{I}_{o\_1}$, as shown in Fig.4a. Otherwise, the four vectors can form a triangle. We set $\vec{I}_{o\_4}$ in the opposite direction of $\vec{I}_{o\_1}$ and a new vector is $\vec{I}'_{o\_1}$ generated whose magnitude is $|\vec{I}'_{o\_1}| = |\vec{I}_{o\_1}| - |\vec{I}_{o\_4}|$. A triangle is guaranteed to be formed by $\vec{I}'_{o\_1}$, $\vec{I}_{o\_2}$ and $\vec{I}_{o\_3}$ because $|\vec{I}_{o\_1}| \geq \sum_{i=2}^{4} |\vec{I}_{o\_i}|$. If we apply the discussed three converters interleaving algorithm, proper phase shift can be found as shown in Fig.4b.

C. Interleaving algorithm for $n = N (N \geq 4)$

For N converters, the same reduction process as four converter case can be done. Current vectors of N converters are arranged according to their magnitudes in descending order, ie. $|\vec{I}_{o\_1}| \geq |\vec{I}_{o\_2}| \geq ... \geq |\vec{I}_{o\_N}|$. Again we first check whether it is feasible for the N vectors to form a triangle by comparing $|\vec{I}_{o\_1}|$ with $\sum_{i=2}^{N} |\vec{I}_{o\_i}|$. If $|\vec{I}_{o\_1}| \geq \sum_{i=2}^{N} |\vec{I}_{o\_i}|$, the N

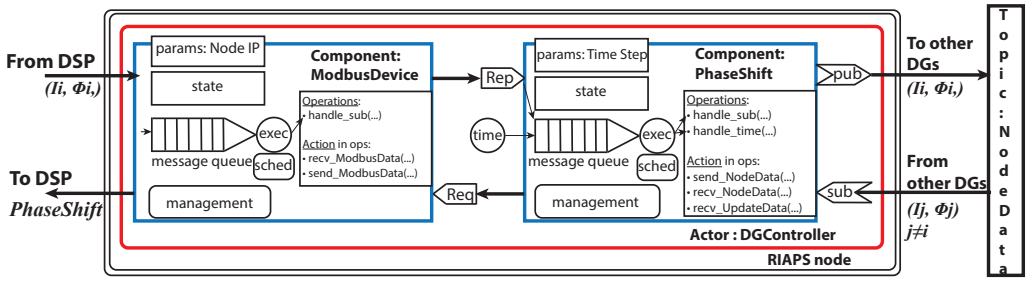


Fig. 5: RIAPS actor containing two components for interleaving algorithm

vectors can not form a triangle. $\vec{I}_{o\_2}, ..., \vec{I}_{o\_4}$ should be set in the opposite direction of $\vec{I}_{o\_1}$.

If $|\vec{I}_{o\_1}| \leq \sum_{i=2}^{N} |\vec{I}_{o\_i}|$, the same as before, we form a new vector $\vec{I}'_{o\_1}$ by setting the smallest vector in the opposite direction of the largest vector whose magnitude is $|\vec{I}'_{o\_1}| = |\vec{I}_{o\_1}| - |\vec{I}_{o\_N}|$. The resulting vectors now are $|\vec{I}'_{o\_1}|, |\vec{I}_{o\_2}|, ..., |\vec{I}_{o\_N-1}|$ for which two magnitude relationships are possible.

1) $|\vec{I}'_{o\_1}| \geq |\vec{I}_{o\_2}| \geq ... \geq |\vec{I}_{o\_N-1}|$*:* it is obvious that

$$|\vec{I}'_{o\_1}| \leq \sum_{i=2}^{N-1} |\vec{I}_{o\_i}| \tag{5}$$

2) $|\vec{I}_{o\_2}| \geq ... \geq |\vec{I}'_{o\_1}| \geq ... \geq |\vec{I}_{o\_N-1}|$*:* As $|\vec{I}_{o\_2}| \leq |\vec{I}_{o\_1}|$ and $N \geq 4$, we can have

$$|\vec{I}_{o\_2}| \leq (|\vec{I}_{o\_1}| - |\vec{I}_{o\_N}| + (N-3)|\vec{I}_{o\_N}|) \tag{6}$$

Considering $|\vec{I}_{o\_i}| \geq |\vec{I}_{o\_N}|$, it is true that,

$$|\vec{I}_{o\_2}| \leq (|\vec{I}_{o\_1}| - |\vec{I}_{o\_N}| + \sum_{i=3}^{N-1} |\vec{I}_{o\_i}|) \tag{7}$$

Substituting $\vec{I}'_{o\_1} = \vec{I}_{o\_1} - \vec{I}_{o\_N}$,

$$|\vec{I}_{o\_2}| \leq (|\vec{I}'_{o\_1}| + \sum_{i=3}^{N-1} |\vec{I}_{o\_i}|) \tag{8}$$

Eq.(5) and Eq.(8) show that the remaining $N-1$ vectors still satisfy the condition of forming a triangle. The same process is repeated until the number of vectors is reduced to three.

IV. Implementation of Improved Interleaving Algorithm in the RIAPS Platform

In the proposed architecture in Fig. 1, each converter is controlled by its local controller. On top of the local controller, a RIAPS node is associated with it providing computation and communication capabilities. The RIAPS nodes can communicate with the local controller using Modbus protocol. The communication among RIAPS nodes are facilitated by ZeroMQ middleware. Typical communication patterns supported by the RIAPS platform include request-reply and publish-subscribe.

In a RIAPS node, specific functions are realized by actors. An actor can be comprised of several components. Fig. 5 shows the RIAPS actor for controlling the phase shift of

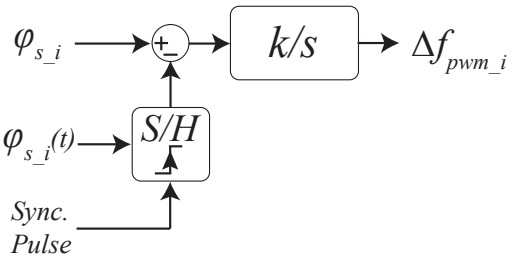


Fig. 6: Control block digram for phase shift in converter i

the converter. The actor has two components. One is a Modbus component communicating with the converter's local controller. The other is a phase shift component where the proposed algorithm is computed.

Every time step, the Modbus component sends out a request to pull out the current vector $\vec{I}_{o\_i}$ which is calculated by the converter's local controller. This vector is then passed to the phase shift component to calculate the phase shift. The calculated phase shift is then sent back to the converter through another Modbus message. The vector $\vec{I}_{o\_i}$ is also published across the RIAPS platform by the phase shift component. Therefore, each RIAPS node will publish its vector while receiving vectors from the others nodes.

After the local converter controller receives the phase shift, it is stored in a register. With the synchronization pulses from the RIAPS nodes to the local controllers, there are two methods to achieve desired phase shift,

1) Regulating the phase directly by loading the phase shift to the counter register of the triangle wave carrier generator when it receives the synchronization pulse from its RIAPS node.
2) Regulating the phase indirectly by regulating the PWM frequency of each converter.

The first method offsets the PWM phase shift immediately upon receiving the synchronization pulse. However, it introduces an abrupt change of the PWM duty cycle at the synchronization moment. This can degrade the output current quality and thus is not preferred.

In this paper the second method is selected and implemented in the local controllers. The control block diagram is shown in Fig.6. Upon receiving the synchronization pulse, the local

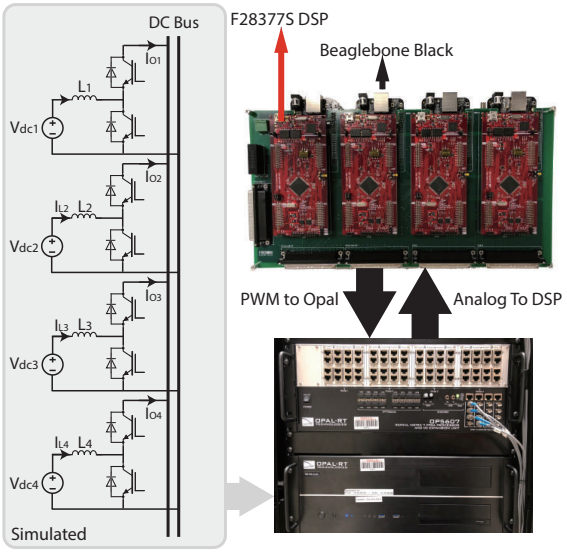


Fig. 7: HIL real-time simulation setup

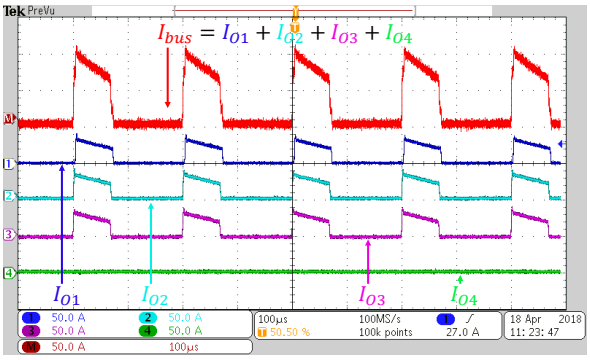


Fig. 8: Test result of case 1: converter currents and bus current

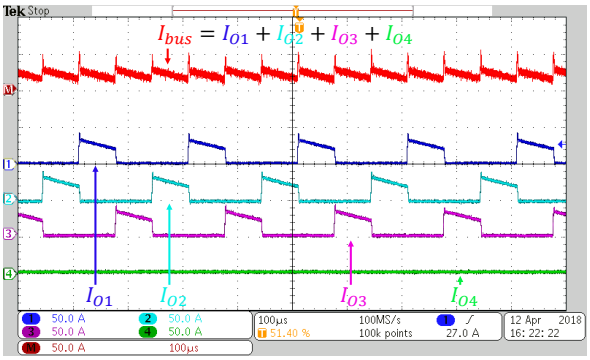


Fig. 9: Test result of case 2: converter currents and bus current

controller samples and holds the phase at that moment and compares it with the desired phase shift from the RIAPS node. The error is passed to an integrator whose output is used to offset the switching frequency of the converter. The integrator gain k is selected small enough so that the switching frequency does not deviates from its normal value too much. In steady state, the adjusted switching frequency $f_{pwm\_i} + \Delta f_{pwm\_i}$ is the same for all converters. Δf_{pwm_i} compensates for the differences in crystal frequency drift, temperature and etc. The correct phase shift is also guaranteed by accurate synchronization pulses across the entire RIAPS platform.

V. Simulation Results

To validate the performance of the proposed algorithm and the synchronization capability of the RIAPS platform, HIL real-time simulations are conducted.

A converter system with four boost converters are simulated in Opal-RT as shown in Fig. 7. The real-time simulation time step is 0.5 μs. In [17] it is shown that the fast trasient dynamics of power eletronics converter can be captured by such as small time step. The four converters are controlled by four DSPs (F28377S from Texas Instruments), respectively. There is no direct information exchange between DSPs. Each DSP is connected to its RIAPS node by two channels. One is a bidirectional serial communication channel which runs on Modbus. The other is a unidirectional channel sending synchronization pulse from the RIAPS node to the DSP. The hardware for RIAPS node is a BBB single board computer.

Five cases are simulated. In the first case no synchronization or interleaving is implemented. Then, in the second and third case three converters are working in symmetric condition with the proposed interleaving algorithm implemented. The influence of the synchronization frequency are tested by varying the frequency of the synchronization pulse. The last two cases show the performance of the proposed interleaving algorithm for three and four converters working in asymmetric condition, respectively. For all cases, the inductors are identical $L_1 = L_2 = L_3 = L_3 = 1.75$ mH. The DC bus voltage is 400 V. The switching frequency of all converters is 5 kHz. All the converters are working in current mode to control the inductor current. Their current reference is set to $I = 25A$.

A. Case 1: three converters in symmetric condition without interleaving

In the first simulation, only the first three converters are operating. The fourth converter is shut down. The input voltages of the converters are set to $V_{\rm dc1} = V_{\rm dc2} = V_{\rm dc3} = 140$ V. When no synchronization or interleaving is implemented for the converters, the phase difference between different converters becomes time-varying. This is because their internal clocks can not be exactly the same due to crystal frequency drift, different ambient temperature and other operation conditions. The worse case happens when the three converters have the same phase as shown in Fig. 8. The large pulse current can increase system losses and decrease the DC-link capacitor life time.

B. Case 2: three converters in symmetric condition with synchronization pulse every 10 ms

The parameters of the second simulation are the same as the first one except the interleaving algorithm is implemented in the RIAPS platform with synchronization pulse generated by

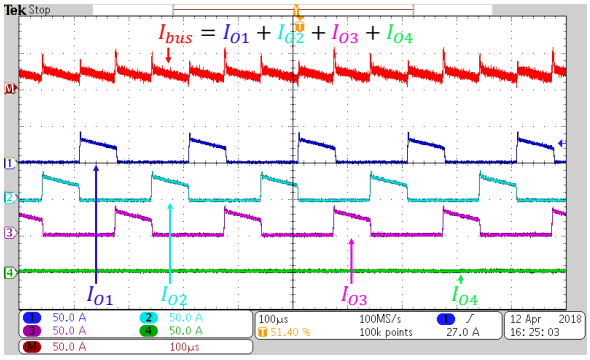


Fig. 10: Test result of case 3: converter currents and bus current

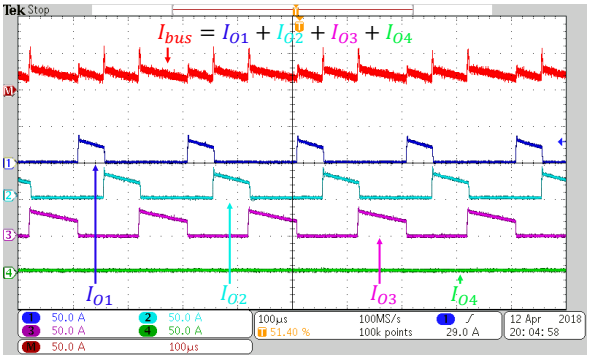


Fig. 11: Test result of case 4: converter currents and bus current

the RIAPS nodes every 10 ms. Fig. 9 shows the output current I_{Oi} $(i = 1, 2, 3, 4)$ of the converters and the sum of them. Because the three converters are working in the symmetric condition, the proposed algorithm calculates phase shift $\frac{4\pi}{3}$ and $\frac{2\pi}{3}$ for the second and third converter, respectively. The sum of the output currents shows a much smaller switching frequency ripple.

C. Case 3: three converters in symmetric condition with synchronization pulse every 100 ms

In this case the synchronization pulse is generated at every 100 ms. The simulation results are shown in Fig. 10. Although the same phase shifts are calculated as in case 2, the performance of interleaving is not as good. The converter current phase shifts constantly fluctuate around $\frac{4\pi}{3}$ and $\frac{2\pi}{3}$, respectively. The bus current also varies around the operating point in Fig. 9. The explanation is provided as following. The DSPs are synchronized by synchronization pulses. After receiving the synchronization pulse, the control loop in Fig. 6 will drive the frequency to eliminate the sampled phase difference. If the synchronization frequency is low, the phase error might be overcompensated, resulting in degraded performance. If the synchronization frequency is lowered further, the system can become unstable. Test results show a synchronization pulse generated every 10 ms is accurate enough for interleaving converters with 5 kHz switching frequency.

D. Case 4: three converters in asymmetric condition with synchronization pulse every 10 ms

In this simulation, the converters are working in asymmetric condition with different DC voltages. The voltages are set to $V_{\mathrm{dc1}} = 100$ V, $V_{\mathrm{dc2}} = 140$ V, $V_{\mathrm{dc3}} = 180$ V. Fig. 11 shows the simulation result. The output currents from the three converters are interleaved to minimize the switching frequency component although they have different duty cycles. Close observation reveals that the switching frequency components is not completely removed by the interleaving algorithm. This is because the proposed interleaving algorithm is essentially an open loop control. The phase shift calculated by phase shift component in Fig. 5 is based on the measurement information. There is no feedback to guarantee the triangle formation in Fig. 3b. However, the bus current has a much smaller switching frequency ripple compared to the case without interleaving.

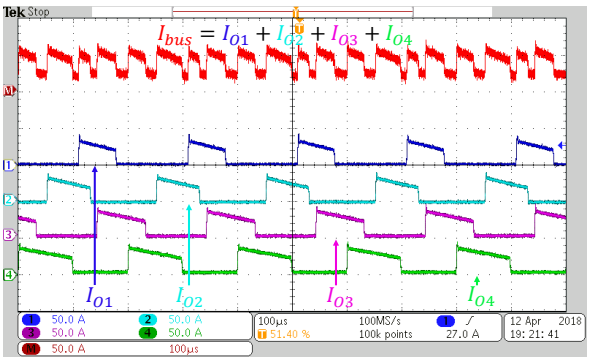


Fig. 12: Test result of case 5: converter currents and bus current

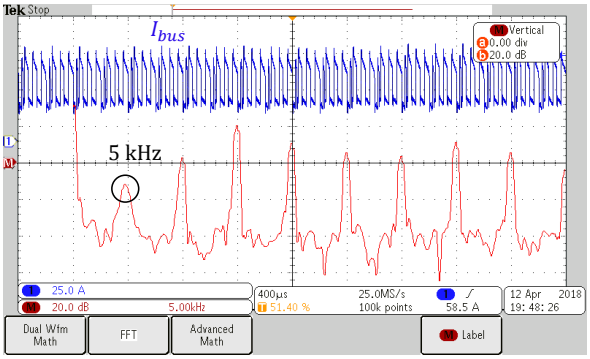


Fig. 13: Test result of case 5: bus current and its spectrum

E. Case 5: four converters in asymmetric condition with synchronization pulse every 10 ms

Now four converters are operating in asymmetric conditions. The fourth converter's DC voltage is set to 200 V. The current waveforms are shown in Fig. 12. The four converter currents are interleaved with the proposed algorithm. In this case, the magnitude of the converter output current switching frequency components satisfies $|\vec{I}_{o\_1}| \leq |\vec{I}_{o\_2}| \leq |\vec{I}_{o\_3}| \leq |\vec{I}_{o\_4}|$. Based on the proposed algorithm, $\vec{I}_{o\_1}$ and $\vec{I}_{o\_4}$ are out of phase. The

sum of of $\vec{I}_{o\_1}$ and $\vec{I}_{o\_4}$ forms a triangle together with $\vec{I}_{o\_2}$ and $\vec{I}_{o\_3}$.

The bus current and its spectrum is shown in Fig. 13. The 5 kHz switching frequency component is reduced below 0 dB. The most prominent component becomes the third order component.

VI. Conclusion

In this paper we propose an adaptive interleaving algorithm for multi-converter systems. The proposed algorithm is implemented on the RIAPS platform. The time synchronization capability of the RIAPS platform and the performance of the proposed algorithm is validated through HIL simulation. Simulation results show the RIAPS platform time synchronization accuracy is high enough to interleave different converters. The proposed algorithm can greatly reduce the switching frequency component of the bus current.

Acknowledgment

This work was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000666. The views and opinions of authors expressed herein do not necessarily state or reflect those of the US Government or any agency thereof.

References

[1] M. G. Villalva, J. R. Gazoli, and E. Ruppert Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," *IEEE Transactions on power electronics*, vol. 24, no. 5, pp. 1198–1208, 2009.

[2] Y. Du, R. Jain, and S. M. Lukic, "A novel approach towards energy storage system sizing considering battery degradation," in *Energy Conversion Congress and Exposition (ECCE), 2016 IEEE*. IEEE, 2016, pp. 1–8.

[3] T. Chen, R. Yu, Q. Huang, and A. Q. Huang, "A single-stage bidirectional dual-active-bridge ac-dc converter based on enhancement mode gan power transistor," in *2018 IEEE Applied Power Electronics Conference and Exposition (APEC)*, March 2018, pp. 723–728.

[4] F. Xue, R. Yu, and A. Huang, "Fractional converter for high efficiency high power battery energy storage system," in *Energy Conversion Congress and Exposition (ECCE), 2017 IEEE*. IEEE, 2017, pp. 5144–5150.

[5] N. Eghtedarpour and E. Farjah, "Power control and management in a hybrid ac/dc microgrid," *IEEE transactions on smart grid*, vol. 5, no. 3, pp. 1494–1505, 2014.

[6] P. C. Loh, D. Li, Y. K. Chai, and F. Blaabjerg, "Autonomous control of interlinking converter with energy storage in hybrid ac–dc microgrid," *IEEE Transactions on Industry Applications*, vol. 49, no. 3, pp. 1374–1382, 2013.

[7] D. Chen, Y. Xu, and A. Q. Huang, "Integration of dc microgrids as virtual synchronous machines into the ac grid," *IEEE Transactions on Industrial Electronics*, 2017.

[8] C. Chang and M. A. Knights, "Interleaving technique in distributed power conversion systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, no. 5, pp. 245–251, 1995.

[9] X. Liang, C. Zhang, S. Srdic, and S. Lukic, "Predictive control of a series-interleaved multi-cell three-level boost power factor correction converter," *IEEE Transactions on Power Electronics*, 2017.

[10] X. Xu, W. Liu, and A. Q. Huang, "Two-phase interleaved critical mode pfc boost converter with closed loop interleaving strategy," *IEEE Transactions on Power Electronics*, vol. 24, no. 12, pp. 3003–3013, 2009.

[11] J. Shen, K. Rigbers, and R. W. De Doncker, "A novel phase-interleaving algorithm for multiterminal systems," *IEEE Transactions on Power Electronics*, vol. 25, no. 3, pp. 741–750, 2010.

[12] T. Xu and F. Gao, "Global synchronous pulse width modulation of distributed inverters," *IEEE Transactions on Power Electronics*, vol. 31, no. 9, pp. 6237–6253, 2016.

[13] S. Eisele, I. Madari, A. Dubey, and G. Karsai, "Riaps: resilient information architecture platform for decentralized smart systems," in *20th IEEE International Symposium On Real-Time Computing, IEEE. Toronto, Canada: IEEE*, vol. 5, 2017, p. 2017.

[14] S. Eisele, A. Dubey, G. Karsai, and S. Lukic, "Transactive energy demo with riaps platform," in *Proceedings of the 8th International Conference on Cyber-Physical Systems*. ACM, 2017, pp. 91–91.

[15] Y. Du, H. Tu, S. Lukic, D. Lubkeman, A. Dubey, and G. Karsai, "Implementation of a distributed microgrid controller on the resilient information architecture platform for smart systems (riaps)," in *Power Symposium (NAPS), 2017 North American*. IEEE, 2017, pp. 1–6.

[16] P. Volgyesi, A. Dubey, T. Krentz, I. Madari, M. Metelko, and G. Karsai, "Time synchronization services for low-cost fog computing applications," in *Rapid Systems Prototyping (RSP)*. ACM/IEEE, 10 2017.

[17] H. Yu, H. Tu, and S. Lukic, "A passivity-based decentralized control strategy for current-controlled inverters in ac microgrids," in *2018 IEEE Applied Power Electronics Conference and Exposition (APEC)*, March 2018, pp. 1399–1406.