Optimization of System Identification for Multi-Rail DC-DC Power Converters

JIN XU

B.Sc, M.Sc.

A Thesis submitted for the degree of

Doctor of Philosophy

July 2021

School of Engineering

Newcastle University

United Kingdom

Abstract

There have been many recursive algorithms investigated and introduced in real time parameter estimation of Switch Mode Power Converters (SMPCs) to improve estimation performance in terms of faster convergence speed, lower computational cost and higher estimation accuracy. These algorithms, including Dichotomous Coordinate Descent (DCD) - Recursive Least Square (RLS), Kalman Filter (KF) and Fast Affine Projection (FAP), etc., are commonly applied for performance comparison of system identification of single-rail power converters. When they need to be used in multi-rail architectures with a single centralized controller, the computational burden on the processor becomes significant. Typically, the computational effort is directly proportional to the number of converters/rails. This thesis presents an iterative decimation approach to significantly alleviate the computational burden of centralized controllers applying real-time recursive system identification algorithms in multi-rail power converters. The proposed approach uses a flexible and adjustable update rate rather than a fixed rate, as opposed to conventional adaptive filters. In addition, the step size/forgetting factors are varied, as well, corresponding to different iteration stages. As a result, reduced computational burden and faster model update can be achieved. Recursive algorithms, such as Recursive Least Square (RLS), Affine Projection (AP) and Kalman Filter (KF), contain two important updates per iteration cycle. Covariance Matrix Approximation (CMA) update and the Gradient Vector (GV) update. Usually, the computational effort of updating Covariance Matrix Approximation (CMA) requires greater computational effort than that of updating Gradient Vector (GV). Therefore, in circumstances where the sampled data in the regressor does not experience significant fluctuations, re-using the Covariance Matrix Approximation (CMA), calculated from the last iteration cycle for the current update can result in computational cost savings for real- time system identification. In this thesis, both iteration rate adjustment and Covariance Matrix Approximation (CMA) re-cycling are combined and applied to simultaneously identify the power converter model in a three-rail power conversion architecture.

Besides, in multi-rail architectures, due to the high likelihood of the at-the-same-time need for real time system identification of more than one rail, it is necessary to prioritize each rail to guarantee rails with higher priority being identified first and avoid jam. In the thesis, a workflow, which comprises sequencing rails and allocating system identification task into selected rails, was proposed. The multi-respect workflow, featured of being dynamic, selectively pre-emptive, cost saving, is able to flexibly change ranks of each rail based on the application importance of

rails and the severity of abrupt changes that rails are suffering to optimize waiting time and make-span of rails with higher priorities.

Dedication

Dedicated to my mother Yinglan Xie, my father Daqing Xu, my brother Fan Xu, and to the memory of my uncle Dexian Xie.

Acknowledgment

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. Matthew Armstrong for the continuous support during my doctoral research, for his great patience, immense knowledge, tolerance, motivation and suggestions. His guidance helped me in all the time of research and writing of this thesis.

I am also thankful to the other supervisor: Dr. Maher Al-Greer, for his insightful comments on every draft I wrote, and also the persuasive advices and hard questions which incited me to widen my research. His dedication and focus on researches also encouraged me to work.

I am also hugely appreciative to my friends and colleagues in EP group. Special mention goes to Dr. Mohamed Ahmeid for sharing his circuit design and hardware, Dr. Ruisheng Li, Dr. Haimeng Wu, Dr. Weichi Zhang, Dr. Chen Wang, Dr. Petros Missailidis, Dr. Binh Vu for their valuable comments on my research, and Paul Killan, Xiang Wang and Dr. Cuili Chen for their guidance of experimental equipment usage.

I warmly appreciate my friends, Huiqing Xu, Minglian Zhao, Junchao Li, Jiaxin Hou, Jiaxin Chen, Jiajie Fan, Yu Tang and Diya Bhattachaya, for their help, suggestions, supports and company in various forms during these years. The time spent with them is memorable. During the thesis writing, my cousin, Li Xie, also expressed a great concern on the progression and we encouraged each other. Specially, I would like to express my deep thankfulness to Xuelan Wang, a great friend who gave me enormous mental support, useful suggestions and help throughout my entire PhD life.

I dedicate this PhD thesis to my families, who kept the financial support throughout this research even if they suffered a hard time. Without their help, I could not conduct this work. Specially, I would dedicate this PhD thesis to my father (Daqing Xu): Your remote support, encouragement and great patience were the bedrock upon since the PhD project started in 2016. Your positive and fearless attitude towards difficulties inspired and will keep infecting me to perseveringly pursue my goals. I also dedicate this work to the memory of my uncle, Dexian Xie, his concern always warmed me in these years.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ABC	Activity Based Costing
AC	Analogue Control
ACC	Average Current Control
ADC	Analogue-to-Digital Conversion
ADRS	Adaptive Double Ring Scheduling
AE	Average Error
ALU	Arithmetic Logic Unit
AP	Affine Projection
ARMA	Autoregressive Moving Average
ARMAX	Autoregressive Moving Average with Extra Input
ARX	Autoregressive with Extra Input
BBO	Biogeographical Based Optimization
BLS	Batch Least Square
BJ	Box-Jenkins
CCM	Continuous Conduction Mode
CCS	Code Composer Studio
CMA	Covariance Matrix Approximation
CPU	Central Processor Unit
CR	Cognitive Radio
CT	Correction Term
DCD	Dichotomous Coordinate Descent
DCM	Discontinuous Conduction Mode
DFB	Describing-Function-Based
DFT	Discrete Fourier Transform
DMA	Direct Access Memory
DSPs	Digital Signal Processors
eCAN	Enhanced Controller Area Network
eCAP	Capture Input
EDF	Earliest Deadline First
ERLS	Exponential RLS
ESR	Equivalent Series Resistance
FAP	Fast Affine Projection

FCFS	First Come First Serve
FFT	Fast Fourier Transform
FLC	Feedback Linearizing Controller
FPGA	Field-Programmable Gate Array
FPU	Floating Point Unites
FRIT	Fictitious Reference Iterative Tuning
GPIO	General Purpose Input/output
GPU	Graphics Processing Unit
GV	Gradient Vector
HSS	Harmonic State-Space model
IBA	Intermediate Bus Architecture
IC	Integrated Circuits
IDE	Integrated Development Environment
IF	Iteration Frequency
IIR	Infinite Impulse Response
IMC	Internal Model Control
IRBS	Inverse Repeat Binary Sequence
KF	Kalman Filter
LAC	Linear Averaged Controller
LCO	Limit Cycle Oscillation
LMS	Least Mean Square
MAF	Moving Average Filters
MIMO	Multi-Input Multi-Output
MISO	Multi-Input Single-Output
MLBS	Maximum Length Binary Sequence
MSPS	Mega Samples per Second
NG	New Guess
NLMS	Normalized Least Mean Square
OE	Output-Error
OPA	Operational Amplifier
PBC	Passivity-Based Controller
PCC	Peak Current Control
PCM	Priority Coefficient Matrix

PE	Process Error
PID	Proportional-Integral-Derivative
POL	Point of Load
PRBS	Pseudo-Random Binary Sequences
PSO	Particle Swarm Optimization
PU	Primary-User
PWM	Pulse-Width Modulation
RLS	Recursive Least Square
SALS	Step-adaptive Approximation Least Squares
SAM	Simple Additive Method
SCI	Serial Communication Interface
SF	Sampling Frequency
SI	System Identification
SISO	Single-Input Single-Output Systems
SMC	Sliding Mode Control
SMPCs	Switch Mode Power Converters
SPI	Serial Peripheral Interface
SSA	Small-Signal Average
SSE	Steady State Error
SSI	Steady State Indicator
SWP	Simple Weighted Product
TBPRD	Time-Base Period Register
TI	Texas Instruments
UCV	Urgency Coefficient Vector
UPS	Uninterruptable Power Supplies
VMT	Virtual Machine Tree

List of Symbols

k_p	Proportional Gain
k_i	Integral Gain
k_d	Derivative Gain
L	the inductor
C	the output capacitor
R_C	the parasitic resistances of the capacitor
R_L	the parasitic resistances of the inductor
f_{sw}	the switching frequency
D	duty cycle
T_{sw}	switching period
V_L	the inductor voltage
V_{in}	the DC voltage input
i_L	the inductor current
V_{out}	the voltage output
i_{load}	the output current
R_{load}	the resistance load
v_C	the capacitor voltage
G_{DC}	the DC gain
ω_n	the natural frequency
ξ	the damping ratio
ω_z	zero frequency
M_p	overshoot
t_p	peaking time
t_s	the settling time
ΔV_{out}	the ripple magnitudes of the output voltage
Δi_L	the ripple magnitudes of the inductor current
e	control error signals
d	outputs control signals
f_b	the loop bandwidth
ω_i	the updated estimation vector
μ	the step size of AP

y	the scalar-valued output
R_{yu}	the cross-covariance vector of y
u	regressor
R_u	a positive-definite covariance matrix of u
$\widehat{R_u}$	the instantaneous values of R_u
$\widehat{R_{yu}}$	the instantaneous values of R_{yu}
f_c	the corner frequency
f_s	the sampling frequency
a_1 and a_2	denominators coefficients of transfer function
b_1 and b_2	numerators coefficients of transfer function
ϵ	the constant regularization sequence
λ	the forgetting factor of RLS
G_i	Kalman Gain
ω_i	new parameter guess
P_i	the estimation dispersion of KF
P_i^+	covariance matrix project of KF

Chapter 1 Introduction

1.1 Introduction

Multi-rail power converter architectures are commonly used in distributed power supply applications to convert a single voltage supply rail to multiple regulated voltage levels via several Point of Load (POL) converters [1, 2]. Multi-rail DC-DC power converters have been employed in computing and communication equipment [3, 4], electric vehicles [5], and DC micro-grids [6], their main application fields are to provide low voltages with high power density to downstream devices including microprocessors, FPGAs and their peripherals [7]. A typical multi-rail power supply product applied to power microcontrollers and their peripherals is TPS653850-Q1 from Texas Instruments. A similar type of product can be seen in Analog Devices, such as LT8602 [8]. For carrying out these applications in various working conditions of Switch Mode Power Converters (SMPCs), robustness to system changes is important [9-12]. The operation of SMPCs may suffer unexpected or periodic load changes, abrupt disturbances, gradual capacitor degradations, sudden malfunction of circuit components and occasional additions of paralleled output capacitors [13], etc. As these cases may happen concurrently, it is difficult to homogenize parameter change rates to specific numbers and the odds of system parameter changes become highly random. Therefore, a robust control loop, used to cope with these randomly happened system variations, is required. In such cases, control parameters may need to be adjusted in real-time (adaptive control) to minimize the impact of these system variations and achieve optimal regulation. Such controller tuning – adjusting controller parameters – based on information received from a real-time model of the plant, is normally based on online system identification [14].

Compared with conventional control techniques, adaptive control is more advanced as it can adjust controller parameters on-site based on the monitored working conditions of objectives. System identification, as a commonly used tool to realise adaptive control technique, can be achieved by estimating the model parameters of the power converters (parametric methods) or analyzing the system frequency response (non-parametric methods) [15-20]. Non-parametric methods usually need open-loop control, transient response acquisition, and off-line analyses [21-23], while parametric methods, which use algorithms with particular application in areas such as adaptive control, may be achieved during closed-loop operation. The performance of algorithms can be judged by convergence time, computational cost, and estimation accuracy [24]. Literature shows that variants of the RLS algorithm [15, 25-27] are widely used in power converter applications [28, 29]. For instance, the Dichotomous Coordinate Descent (DCD)-

RLS is shown to be more computationally efficient than classical RLS [28, 30]. In [31], a variable-forgetting factor method, computationally heavier than the classical RLS though, is proposed to improve tracking of real-time parameter variations. A Kalman Filter (KF) approach has also been used in single-rail converter applications [31]. The KF is demonstrated to have advantages in dealing with abrupt load changes, but again computational effort is an issue. Besides, a Fast Affine Projection (FAP) algorithm was proposed [15]. Results show that FAP performs better than RLS in terms of convergence time, estimation accuracy, and computational cost. To further alleviate the computational burden of parameter estimation, recently the authors in [26] proposed a Step-adaptive Approximation Least Squares (SALS) for high-frequency estimation of a single-rail multi-phase buck converter.

According to pieces of literature, the recursive algorithms perform well in real-time parameter estimation for single power converters. However, in multi-rail architectures with a centralized single controller, the computational burden will become heavy, increasing proportionately with the addition of rails. For example, if the available computation time is 50μs, the employed processor should finish 64 additions, 109 multiplications, and 1 division in 50μs for single-rail parameter estimation by using RLS. If three rails are simultaneously identified, the computational burden in the 50μs will be increased to 192 additions, 327 multiplications, and 3 divisions. The significant increase in the computational burden could cause the need for advanced processors more computationally capable particularly, resulting in extra investments. As such, finding the clever approaches that may efficiently alleviate the computational burden during each sampling event is what this project is based on.

1.2 Scope and Contribution of the Thesis

This thesis focuses on the problem of the tremendous computational costs of parameter estimation of multi-rail power conversion architectures, proportionally increased with the number of rails. Approaches would be proposed to try to save the computational efforts of centralised parameter estimation of multi-rails, and guarantee there is not any compromise in other identification performances, such as the convergence time and estimation accuracy.

Besides, centralised parameter estimation requires prioritization of System Identification (SI) requests as well to avoid request jam in rush hours. Consequently, a ranking and task allocation approach is also proposed in this paper.

In Summary, the main objectives, and contributions to knowledge of this research are:

1. To derive an iterative decimation approach with adaptive forgetting factors/step sizes of conventional algorithms to reduce computational complexity of real time parameter estimation
2. To reuse CMA/CT in algorithms during iteration processes to reduce computational burden
3. To develop a hard-real-time prioritization approach featured of being dynamic, multi-respects to rank SI requests in multi-rails.
4. Although all the three proposed methods are applied for multi-rail power converters, they can be also used into areas where algorithms can be applied as they are mainly for application optimization and simplification.

1.3 Publications Arising from this Research

1. **J. Xu**, M. Armstrong and M. Al-Greer, "Parameter Estimation of DC-DC Converters Using Recursive Algorithms with Adjustable Iteration Frequency," 2018 IEEE 19th Workshop on Control and Modelling for Power Electronics (COMPEL), 2018, pp. 1-8.
2. **J. Xu**, M. Armstrong and M. Al-Greer, "Computational Burden Reduction in Real-Time System Identification of Multi-Rail Power Converter by Re-using Covariance Matrix Approximation," 2020 IEEE Applied Power Electronics Conference and Exposition (APEC), 2020, pp. 2150-2157.
3. **J. Xu**, M. Armstrong and M. Al-Greer, "Centralized System Identification of Multi-Rail Power Converter Systems Using an Iterative Decimation Approach," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 8, pp. 3520-3533, Aug. 2021.
4. **J. Xu**, M. Armstrong and M. Al-Greer, "A Real-Time Prioritization Approach Applied for System Identification Request Coordination in Multi-Rail Power Conversion Architecture," 2021 56th International Universities Power Engineering Conference (UPEC), 2021, pp. 1-6.

1.4 Layout of the Thesis

This thesis is organised into 7 chapters. This chapter introduces the background and the brief introduction of the project.

Chapter 2 presents the topology, the operation, the SSA (Small Signal Average) model and several commonly used control techniques of Buck Converters. Besides, multi-rail power conversion architectures comprising several Buck Converters, their applications and the control challenges caused by various system variations are introduced. Then, the process of the pole-

zero cancellation approach for a digital PID controller for a simulated Buck Converter is demonstrated in a detailed way. The transfer function of the buck converter and the controller parameters acquired in this chapter is used for parameter estimation later.

Chapter 3 shows an overview of system identification techniques applied in SMPCs, including the introductions of non-parametric estimation and parametric estimation methods. Iterative parameter estimation, mainly applied in this project, is reviewed in a detailed way. Besides, to build a prioritization workflow for system identification of multi-rail power conversion architectures, various task ranking/sequencing approaches proposed in other application areas are investigated. The optimization approaches of the parameter estimation process proposed in Chapter 4 and 5, and the prioritization workflow designed in Chapter 6 are both based on the background knowledge introduced in this chapter.

Chapter 4 proposes an iteration decimation approach and a Covariance Matrix Approximation (CMA)/Correction Term (CT) recycling approach to reduce computational burden of simultaneous system identification of multi-rails. The first approach utilizes a combination of adjustable forgetting factor/step size and a reduced iteration frequency. The two approaches are validated in both simulation and experiments by applying RLS and AP. They significantly relief burdens without compromises of accuracy and convergence speed.

Chapter 5 presents the experimental validation of the iteration decimation approach and the CMA/CT reusing approach introduced in Chapter 4. Through conducting a complete real-time parameter estimation platform, the two proposed approaches are implemented and verified on three algorithms (RLS, AP and KF). The experimental results of the two computational cost-saving approaches are in very good accordance with the simulation results presented in Chapter 4, suggesting the applicability of the two approaches in a broad kind of adaptive filters.

Chapter 6 presents a proposed workflow, containing updating request priorities in hard-real-time and task allocating, to coordinate SI requests. This workflow is validated in simulation, which suggests that it can comprehensively consider many elements, dynamically and flexibly update scheduling strategy and be cost saving and highly efficient.

Finally, Chapter 7 presents the conclusion of the thesis and suggests the future work.

Chapter 2 Modelling and Digital Control of Multi-Rail DC-DC Switch-Mode Power Conversion Architectures

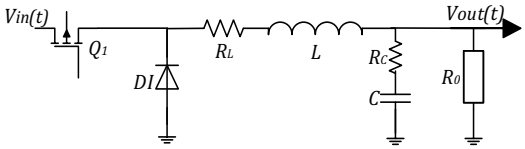
2.1 Introduction

DC-DC Switch Mode Power Converters (SMPCs) are typical power sources providing regulated voltages or currents to various electric equipment. The power level that they manage can be from micro to million watts. Based on applications, their structure can be divided into isolated and non-isolated conversion architectures. To be used in downstream devices (lower power applications) such as computers, telecommunication devices, motors, and electric vehicles, etc., non-isolated ones, which do not contain costly and large-size magnetic transformers, are preferred due to their low power conversion losses, ease of control, small sizes and simple structures. Apart from SMPCs, a linear regulator is another kind of popular power source, however, it suffers serious heat dissipation and low power conversion efficiency, while SMPCs may reach a conversion efficiency of about 97%. As an analogue power processing device, SMPCs comprise passive components (capacitors and inductors) and semiconductor devices (diodes and switches). The outputs and operation requirements of SMPCs may refer to power density, switching frequency, voltage or current levels, high stabilities and robustness in system variation rejection, etc., concluded as high performance in static and dynamic operations. The inputs of non-isolated power converters are typically outputs from AC-DC power rectifiers in DC micro-grids, and the outputs are usually various integrated circuits. As these circuit loads are becoming highly homogenized, the required power levels (output voltages) are increasingly homogenised which can be 1.8V, 3.3V and 5V, etc. As such, converters are inclined to be packaged and the corresponding products can be found in Texas Instruments, Analog Devices, etc.

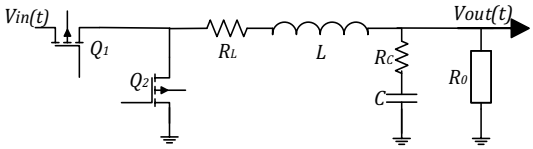
2.2 Circuit Topologies and Operation of DC-DC Converters

Based on different configurations of passive components and switches, different topologies will be designed. The basic function of these topologies is to increase/decrease voltage levels. As such, the most basic topologies are single-phase Buck Converter, Boost Converter and Buck/Boost Converter, shown in Figure 2.1. All the three topologies contain an energy storage device (inductor L), one voltage stabilization device (output capacitor C), one diode (DI) and one semiconductor switch (S_1) (see non-synchronous converters in Figure 2.1). With different arrangements of these devices in circuits, they are respectively used for voltage lowering, rising and both lowering/rising. As converters are increasingly used in low voltage applications, particularly under 5V, the voltage drop passing through diodes is noticeable, diodes were

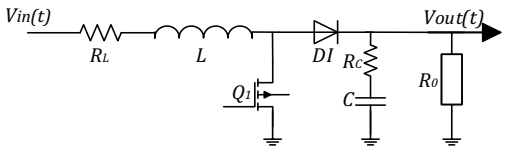
therefore replaced by switches (S_2) of which On/Off states are opposed with S_1 for power loss reduction (see synchronous converters in Figure 2.1). The On/Off state of a couple of switches is controlled by Pulse-Width Modulation (PWM) signal which includes information of switching frequency (f_{sw}) and duty cycle (D). If deadtime is not considered and D is the duty cycle of S_1, the duty cycle of S_2 would be $1 - D$.



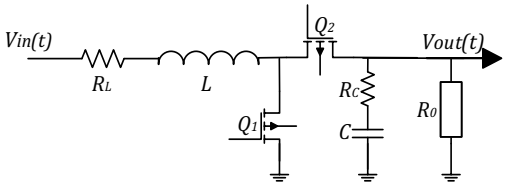
(a) Non-Synchronous Buck Converter



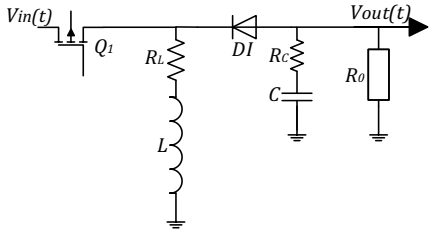
(b) Synchronous Buck Converter



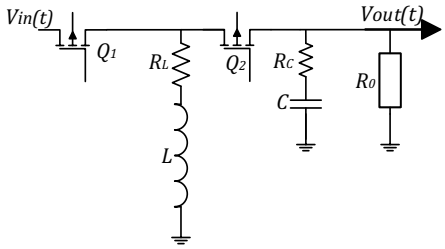
(c) Non-Synchronous Boost Converter



(d) Synchronous Boost Converter

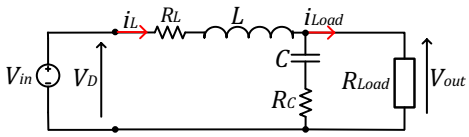


(e) Non-Synchronous Buck/Boost Converter

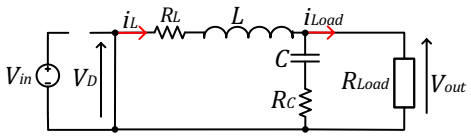


(f) Synchronous Buck/Boost Converter

Figure 2.1 The Topologies of Buck Converter [(a) and (b)], Boost Converter [(c) and (d)] and Buck/Boost Converters [(e) and (f)]



(a) Mode A



(b) Mode B

Figure 2.2 The Two Modes of Buck Converter in Continuous Conduction Operation

Based on different parameter configurations and the output voltage demands, the operation of the aforementioned converters is categorised into two formations. Continuous Conduction Mode (CCM) and Discontinuous Conduction Mode (DCM). In Buck Converter, for example, under continuous operation, there are two current modes (Mode A and B in Figure 2.2): when S_1 in Figure 2.1 is on, the diode (DI) is reverse biased and the inductor (L) stores energy. the

corresponding current loop is shown as Figure 2.2(a). Then, with the off-switching mode, DI would be conducted and the inductor becomes a voltage source, as shown in Figure 2.2(b).

V_{in} is the DC voltage input, V_D is the voltage passing through the diode (DI), i_L is the inductor current, V_{out} is the voltage output, i_{load} is the output current and R_{load} is the resistance load.

During a steady-state, the net increase in the inductor current, or the stored energy, over a switching period (T_{sw}) is zero. As shown in Figure 2.3, the inductor voltage (V_L) equals $V_{in} - V_{out}$ in Mode A, and $-V_{out}$ in Mode B.

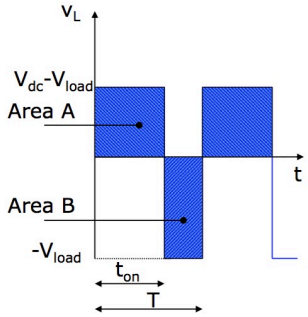


Figure 2.3 The Inductor Voltage (V_L) over a Switching Period (T_{sw})

$$\int_0^{T_{sw}} di_L = \frac{1}{L}\int_0^{T_{sw}} V_L\, dt = \frac{1}{L}\int_0^{t_{on}} (V_{in} - V_{out})\, dt + \frac{1}{L}\int_{t_{on}}^{T_{sw}} (-V_{out})\, dt = 0 \quad (2\text{-}1)$$

$$-(V_{in} - V_{out})t_{on} = -V_{out}(T_{sw} - t_{on}) \quad (2\text{-}2)$$

$$-V_{in}t_{on} = -V_{out}T_{sw} \quad (2\text{-}3)$$

$$\frac{V_{out}}{V_{in}} = \frac{t_{on}}{T_{sw}} = D \quad (2\text{-}4)$$

t_{on} is the on-switching time of S_1.

From (2-4), it is known that the ratio between the output and input voltages of Buck Converter operating in CCM equals the value of duty cycle (D). Similar to Buck Converter, the output-input voltage ratio of the other two converters in CCM is also related to the duty cycle (see Table 2-1).

Table 2-1 The Relation Between Input and Output Voltages of the Three Converters in CCM

Converters	**The Relation between Input and Output Voltages**
Buck Converter	$\frac{V_{out}}{V_{in}} = D$
Boost Converter	$\frac{V_{out}}{V_{in}} = \frac{1}{1-D}$
Buck-Boost Converter	$\frac{V_{out}}{V_{in}} = -\frac{D}{1-D}$

Sometimes, when the inductor (L) finishes energy release, S_1 will not turn on, and the diode (DI) cannot be conducted since there is no voltage drop between its two terminals. Thus, the third mode occurs until S_1 turns on and the inductor (L) starts to store energy again. This operation, which includes three modes, is named Discontinuous Conduction Mode (DCM). The first two modes are the same as those in CCM shown in Figure 2.2, and the third mode is in Figure 2.4.

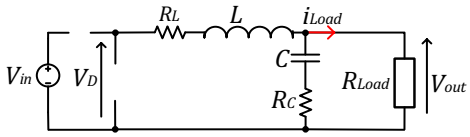


Figure 2.4 The Mode C of the Operation of Buck Converter in DCM

The causes of discontinuous operation are various. Firstly, a small inductor, which symbols poor energy-storage performance, will produce higher-magnitude ripples in current responses, and the discontinuous mode will possibly occur. Secondly, a large resistance load or a significantly low demand of output voltage will lead to a small DC component in the inductor current (i_L), then discontinuous mode (Mode C) occurs. The input-output voltage ratios of converters operating in DCM are different from those in CCM, which not only depend on the duty cycle but the inductor size (L), switching frequency (f_{sw}) and output current (i_{load}), etc.

Apart from these simplest and conventional topologies, there are many other topologies, multi-phase converters typically [32], proposed and applied for efficiency increase, inductor size reduction, output current ripple alleviation, heat dissipation reduction of individual devices, etc. In this project, Buck Converter working in CCM is used to build multi-rail power conversion architectures for experimental validation of proposed approaches.

2.3 Multi-Rail Power Conversion Architecture

2.3.1 The Topology and Control of Multi-Rail Power Converter Architecture

Compared with centralised power supplies, known as being bulky, inefficient and expensive due to the wide use of large-size transformers and long wires [1, 33], distributed ones have gained their popularity in the last three decades by low energy losses, highly flexible structures and high power density with the small size of passive components [34], etc. A typical system of distributed power supplies is Intermediate Bus Architecture (IBA) [1, 33], shown in Figure 2.5 [4, 35, 36], which comprises an isolated intermediate (bus) converter and a multi-rail power conversion architecture [37]. As voltage levels, inputs and outputs, for the bus converters have been highly homogenised, the demand of bus converters to adapt wide input ranges would not

be as strict as it used to be, which brings the simplicity of bus converter design and control [35, 38]. The second conversion step (multi-rail structures) of IBA converts a single power rail to multiple regulated voltage levels via several Point of Load (PoL) converters [1, 2]. As tiny sizes of PoL converters allow them to be implemented directly adjacent to loads, they have been widely applied to power various Integrated Circuits (IC), of which the space of boards is limited [34]. Figure 2.5 shows typical loads, including memories, Central Processor Unit (CPU), Graphics Processing Unit (GPU) and logic circuits, etc., powered by multi-rails in computing and communication equipment [3, 4, 39]. Other areas with similar loads and power supplies include electric vehicles [5] and DC micro-grids [6], etc. PoL converters can be Buck/Boost converters and multi-phase ones [40], while for low-current-load demands or simple-topology inclination, rails are preferred to be single phases. A typical commercial product of power supplies with multi-rail architectures is LT8602 from Analog Devices, which contains four rails (Buck Converters) applied to the power microcontroller and its peripherals [8], similar to TPS62770 from Texas Instruments [41].

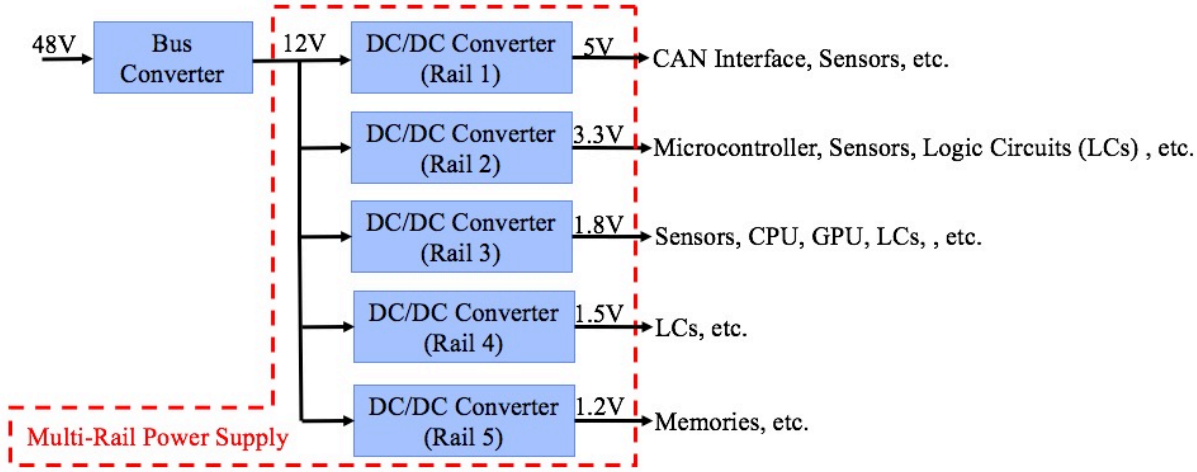


Figure 2.5 The Structures of an Intermediate Bus Converters including a Multi-Rail Power Conversion Architecture with its Loads [Central Processor Unit (CPU), Graphics Processing Unit (GPU)] [42]

2.3.2 System Variations

The operation of SMPCs may suffer abrupt disturbances, unexpected or periodic load changes, gradual capacitor degradations, sudden malfunction of circuit components and occasional additions of paralleled output capacitors [13], etc. Some of them (severe faults) may stop SMPCs from properly working while some others only result in changes in coefficients of mathematical models of systems under normal operation. According to a comprehensive

investigation from industry, [43], the two components most likely to get faults are capacitors and switches. The events that commonly cause system variations are listed as follows:

1. Failure of switches: Switch problems are closely related to operating temperature, while only long-term thermal degradation will cause a breakdown of switches [44-53]. Other problems, such as on-state resistance increase and threshold voltage shift, etc., affect the lifetime of switches rather than the circuit function of converters. Therefore, the influence of switch faults on parameters of mathematical models of converters is limited.

2. Capacitor degradation: As of [54-56], open/short circuits and wearing out are the three main problems existing in capacitors. They are often related to temperatures, operating voltage and ripple current, etc. Compared with open/short circuits that may instantly stop conversion circuits from properly working, wearing out, in terms of capacitance decrease and Equivalent Series Resistance (ESR) increase [57-59], is more likely to gradually change electrical parameters of converters. Therefore, parameter estimation, as a useful tool, can be applied to update coefficients of mathematical models of converters corresponding to capacitor degradation for adaptive control [9, 10].

3. Addition of capacitors: Sometimes extra capacitors will be added in parallel with an output capacitor for various voltage ripple demands, which could also cause system parameter changes. As such, real-time system identification can be adopted for circuit parameter re-estimation.

4. Load changes: During converter operation, load changes, periodic, frequent, expected or unexpected, etc., can be the most possible reason for applying system identification based adaptive control [13]. It is essential to guarantee the robustness of control systems under different circuit parameters which would be significantly influenced by resistance values of loads.

As the above-listed system variations and faults may happen concurrently, it is difficult to homogenize parameter change rates to specific numbers: System operation would suffer frequency-and-severity-unexpected changes, which requires high robustness in SMPC control.

2.4 Modelling of Buck Converter

System modelling is always an essential step to build closed-loop control systems, as based on its results (mathematic expressions of systems), the general information, such as operation, sensitivity and stability, etc., of the system, can be acquired. The small-signal (state-space) average modelling technique, as the most common and ideal method for controller design, is widely used to linearize nonlinear effects (switching events) of SMPCs at an equilibrium point [22, 60-68]. However, a review about SMPC modelling, [69], points out that the accuracy of

small-signal average modelling is doubtable since that ripples of output signals (voltages and currents) are always ignored [70-74]. Papers, [75-78], also suggest that this modelling way limits expressions of instability and unusual operation of DC-DC converters. This review work later compared a Small-Signal Average (SSA) model with two other ones, the Describing-Function-Based (DFB) model and Harmonic State-Space (HSS) model. It turns out that taking ripples into account the DFB model can describe the converter's operation in a more detailed way, such as expressing sideband effects [79]. However, among the three models, only the HSS model can describe beat frequency oscillation caused by coupling interactions of frequencies in multi-input multi-output (MIMO) systems, despite its high orders [62, 80-89]. Other discrete modelling methods, which pursue higher accuracy and nonlinear effect prevention, include Cayley-Hamilton theorem [90] and the spectral decomposition theorem-based [91-93]. In this project, as multi-rail power converters comprise individual sub-converters independent of each other, which means rails are single-input single-output (SISO) converters, the HSS model is unnecessarily applied. Besides, the DFB model is rather to be used in current model control than voltage control due to its capability of subharmonic oscillation evaluation [94-96]. More importantly, more accurate and comprehensive models result in a higher computational burden of hardware implementation. In summary, the SSA model is adopted in this project as it is computational cost-saving and sufficiently precise for controller design purposes. Basic SMPCs topologies (Buck, Boost, Buck/Boost Converters) can be expressed by a linear second-order model [97].

2.4.1 State Space Average Model of Buck Converter

As introduced in **Section 2.2**, in CCM, the operation of Buck Converter is divided into Mode A and B depending on S_1 is On or Off. In a switching interval of S_1, On state (Mode A) keeps $D \cdot T_{sw}$ seconds and Off state (Mode B) $(1-D) \cdot T_{sw}$ seconds.

A. SSA Model of Buck Converter in Ideal Condition (Without R_L and R_C)

As Figure 2.2 shows, if R_L and R_C are neglected, the inductor (L) is charging (i_L is linearly increasing) in Mode A (S_1 is on and S_2 off) while in Mode B (S_1 is off and S_2 on) is discharging. According to Kirchhoff Law, the differential equations expressing the circuit operation can be acquired as (2-5)-(2-8) show.

Mode A:

$$V_{in} = v_{out} + L\frac{di_L}{dt} \tag{2-5}$$

$$i_L = C\frac{dv_C}{dt} + \frac{v_{out}}{R} \tag{2-6}$$

Mode B:

$$0 = v_{out} + L\frac{di_L}{dt} \tag{2-7}$$

$$i_L = C\frac{dv_C}{dt} + \frac{v_{out}}{R} \tag{2-8}$$

where $v_{out} = v_C$ without consideration of Equivalent Series Resistances (ESR) of the inductor (L) and output capacitor (C).

By averaging differential equations of the two operation modes, the SSA model in State Space would be acquired as listed below:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{r}(t) = d(\boldsymbol{A_{on}}\boldsymbol{x} + \boldsymbol{B_{on}}\boldsymbol{r}) + (1-d)(\boldsymbol{A_{off}}\boldsymbol{x} + \boldsymbol{B_{off}}\boldsymbol{r}) \tag{2-9}$$

$$g = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{E}\boldsymbol{r}(t) = d(\boldsymbol{C_{on}}\boldsymbol{x} + \boldsymbol{E_{on}}\boldsymbol{r}) + (1-d)(\boldsymbol{C_{off}}\boldsymbol{x} + \boldsymbol{E_{off}}\boldsymbol{r}) \tag{2-10}$$

where $\dot{\boldsymbol{x}} = \begin{bmatrix}\dot{\boldsymbol{x_1}}\\ \dot{\boldsymbol{x_2}}\end{bmatrix} = \begin{bmatrix}\dot{i_L}\\ \dot{v_C}\end{bmatrix}, \boldsymbol{r} = V_{in}, g = v_{out}, \boldsymbol{A_{on}} = \boldsymbol{A_{off}} = \begin{bmatrix}0 & -\frac{1}{L}\\ \frac{1}{C} & -\frac{1}{RC}\end{bmatrix}, \boldsymbol{B_{on}} = \begin{bmatrix}\frac{1}{L}\\ 0\end{bmatrix}, \boldsymbol{B_{off}} = 0, \boldsymbol{C_{on}} = \boldsymbol{C_{off}} = [0 \quad 1], \boldsymbol{E_{on}} = \boldsymbol{E_{off}} = 0.$

It is known that the inductor current (i_L), duty cycle (d) and output voltage (v_{out}) signals contain two parts: DC component (in the capital) and the small signals with tilde notations under switching action.

$$i_L = I_L + \widetilde{i_L} \tag{2-11}$$

$$d = D + \tilde{d} \tag{2-12}$$

$$v_{out} = V_{out} + \tilde{v}_{out} \tag{2-13}$$

Then, by replacing the above-listed signals in (2-9) and (2-10) with their small-signal expressions (2-11), (2-12) and (2-13), the averaged differential equations representing the combination of the two operation modes can be acquired:

$$\frac{d\widetilde{i_L}}{dt} = -\frac{V_{out}+\tilde{v}_{out}}{L} + \frac{D+\tilde{d}}{L}V_{in} \tag{2-14}$$

$$\frac{d\widetilde{v_C}}{dt} = \frac{I_L+\widetilde{i_L}}{C} - \frac{V_{out}+\tilde{v}_{out}}{RC} \tag{2-15}$$

Through Laplace Transform, (2-14) and (2-15) are transferred into the s domain, shown as follows:

$$s\cdot\widetilde{i_L}(s) = -\frac{\tilde{v}_{out}(s)}{L} + \frac{\tilde{d}(s)}{L}V_{in} \tag{2-16}$$

$$s\cdot\tilde{v}_{out}(s) = \frac{1}{C}\widetilde{i_L}(s) - \frac{1}{RC}\tilde{v}_{out}(s) \tag{2-17}$$

By combining (2-16) and (2-17), the transfer function of a Buck Converter operating in ideal condition is shown as (2-18).

$$\frac{\tilde{v}_{out}(s)}{\tilde{d}(s)} = \frac{V_{in}}{LCs^2+\frac{L}{R}s+1} \tag{2-18}$$

B. SSA Model of Buck Converter with Consideration of ESR

In ideal mode, capacitor voltage equals output voltage, i.e. $v_C = v_{out}$, while when ESR of inductors and capacitors (R_L and R_c) are considered, the circuit operation changes. The corresponding differential equations of Mode A and B in Figure 2.2 are shown as follows.

Mode A:

$$V_{in} - L\frac{di_L}{dt} - R_L \cdot i_L - R_C(i_L - i_0) - v_C = 0 \tag{2-19}$$

$$i_L = i_0 + i_C = \frac{v_{out}}{R} + C\frac{dv_C}{dt} \tag{2-20}$$

$$v_c = v_{out} - R_C \cdot i_C = v_{out} - R_C \cdot C\frac{dv_C}{dt} \tag{2-21}$$

Mode B:

$$-L\frac{di_L}{dt} - R_L \cdot i_L = v_{out} \tag{2-22}$$

$$i_L = \frac{v_{out}}{R} + C\frac{dv_C}{dt} \tag{2-23}$$

$$v_C = v_{out} - R_C \cdot C\frac{dv_C}{dt} \tag{2-24}$$

Then the State-Space matrices in the SSA model [(2-9) and (2-10)] would be:

$$\boldsymbol{A_{on}} = \boldsymbol{A_{off}} = \begin{bmatrix} -\frac{1}{L}\left(R_L + \frac{RR_C}{R+R_C}\right) & -\frac{R}{L(R+R_C)} \\ \frac{R}{C(R+R_C)} & -\frac{1}{C(R+R_C)} \end{bmatrix} \tag{2-25}$$

$$\boldsymbol{B_{on}} = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \tag{2-26}$$

$$\boldsymbol{B_{off}} = 0 \tag{2-27}$$

$$\boldsymbol{C_{on}} = \boldsymbol{C_{off}} = \begin{bmatrix} \frac{RR_C}{R+R_C} & \frac{R}{R+R_C} \end{bmatrix} \tag{2-28}$$

$$\boldsymbol{E_{on}} = \boldsymbol{E_{off}} = 0 \tag{2-29}$$

After averaging and combining these matrices in the two switching modes and separating inductor current (i_L), capacitor voltage (v_C), duty cycle (d) and output voltage (v_{out}) into their DC components and AC small signals, the transfer function (output voltage-to-duty cycle) describing the operation of a buck converter would be acquired:

$$G_{vd}(s) = \frac{\tilde{v}_{out}(s)}{\tilde{d}(s)} = \frac{V_{in}(CR_Cs+1)}{s^2LC\left(\frac{R+R_C}{R+R_L}\right)+s\left[R_CC+C\left(\frac{RR_L}{R+R_L}\right)+\frac{L}{R+R_L}\right]+1} \tag{2-30}$$

By comparing (2-18) and (2-30), it is known that the transfer function of a buck converter contains two poles, while there would be a zero when the ESR of the output capacitor is considered. This zero might bring instability and severer transient responses of the system, which could be neutralised through appropriately locating a pole in its control loop. The general form of a second-order transfer function, shown as (2-31), is introduced in which the DC gain

(G_{DC}), natural frequency (ω_n), damping ratio (ξ) and zero frequency (ω_z) can be expressed by circuit component values [see (2-32), (2-33) and (2-34)]. According to these equations listed below, the values of natural frequency (ω_n) and damping ratio (ξ) could influence the locations of the two poles of (2-30).

$$G_{vd}(s) = G_{DC} \cdot \frac{1+\frac{s}{\omega_z}}{\left(\frac{s}{\omega_n}\right)^2+\frac{2\xi s}{\omega_n}+1} \tag{2-31}$$

$$\omega_n = \sqrt{\frac{R+R_L}{LC(R+R_C)}} \tag{2-32}$$

$$\xi = \frac{\omega_n}{2}\left[R_C C + C\left(\frac{RR_L}{R+R_L}\right) + \frac{L}{R+R_L}\right] \tag{2-33}$$

$$\omega_z = \frac{1}{CR_C} \tag{2-34}$$

$$G_{DC} = V_{in} = \frac{V_{out}}{D} \tag{2-35}$$

Moreover, the two values also could predict transient behaviours such as overshoot (M_p), peaking time (t_p) and settling time (t_s), etc., of the corresponding systems.

$$M_p \approx e^{-\frac{\pi\xi}{\sqrt{1-\xi^2}}} \tag{2-36}$$

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{2-37}$$

$$t_s = \frac{4}{\zeta\omega_n} \tag{2-38}$$

2.4.2 Simulated Comparison of Outputs between Buck Converter and Its SSA Model

To validate that SSA models are accurate enough to express the necessary features of a Buck Converter, the voltage outputs of a Buck converter and of its SSA model would be compared. Firstly, for Buck Converter design and ESR evaluation, di_L and dt in (2-39) are approximated as Δi_L(the magnitude of output voltage ripples) and DT_{sw}(the switching-on time of S_1) as ripples are small enough. As such, the values of the inductor and capacitor can be calculated:

$$V_{in} - V_{out} = L\frac{di_L}{dt} \approx L\frac{\Delta i_L}{DT_{sw}} \implies L = \frac{(V_{in}-V_{out})\cdot D\cdot T_{sw}}{\Delta i_L} = \frac{(1-D)\cdot R\cdot I_L\cdot T_{sw}}{\Delta i_L} \tag{2-39}$$

$$C = \frac{\Delta i_L}{8\cdot f_s\cdot \Delta V_{out}} \tag{2-40}$$

$$ESR = \frac{\Delta V_{out}}{\Delta i_L} \tag{2-41}$$

where $V_{out} = Ri_L$.

As shown in (2-39)-(2-41), after setting up output voltages and (resistance) loads, when the ESR of the inductor is neglected, selection of circuit component values (inductors, capacitors and their ESRs, etc.) of the converter typically depends on the demands of ripple magnitudes

of inductor currents (Δi_L) and output voltages (ΔV_{out}). The typical inductor current ripples are required within 10% of its DC components (I_L) while output voltage ripples within 0.3% of V_{out} [98, 99]. Consequently, the circuit component values are listed as: $V_{in} = 10\text{V}, L = 220\mu\text{H}, C = 330\mu\text{F}, R = 5\Omega$, $R_C = 25\text{m}\Omega, R_L = 68\text{m}\Omega$, $f_S = 20\text{kHz}$ (f_S is the switching frequency). The output voltage is set up as 3.3V so that duty cycle in steady state, a constant without perturbations, is 33%.

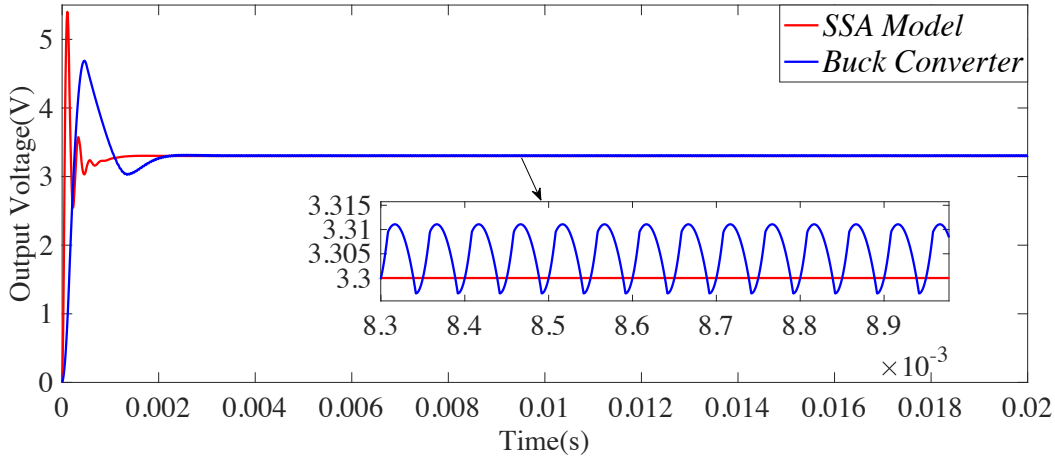


Figure 2.6 The Output Voltage Signal of a Buck Converter

Simulation Results of the Buck Converter and its SSA model, acquired through MATLAB/Simulink, are shown in Figure 2.6. In this thesis, the sampled data used for system identification would not include ripples as the sampling frequency equals the switching frequency. This means the sampled signal under the Zero-Order-Hold effects would be the same with the output from the SSA model. As such, SSA modelling way is detailed enough for adaptive controller design.

2.4.3 Discrete-Time Model of Buck Converter

To build a digital control system, dominantly applied in research experiments and industries nowadays due to its ease of hardware implementation, of Buck Converter, a discrete transfer function (in z domain) of Buck Converter should be developed based on its counterpart in Laplace domain, (2-30). There are many approaches transforming transfer functions from Laplace (s) domain to digital (z) domain, including some simple and easily-applicable methods (backward difference and bilinear transformation) and other ones for particular requirements (impulse invariant is to acquire impulse responses and matched pole/zero mapping for the precise transformation of poles/zeros, etc.). Despite that almost all of these approaches would suffer from aliasing, an appropriate sampling frequency could alleviate this problem.

The sampled signals used for control loop design and system identification are output voltage and duty cycle control signals. For hardware implementation, in a sampling interval of the A/D operation, the instant value of a signal at the beginning of the interval will be recorded and held until the next sampling interval starts. Correspondingly, the frequency of updating duty cycle control signal is the same as sampling frequency. Therefore, Zero-Order-Hold (ZOH) transformation approach shown as (2-42), on which the A/D conversion process is based, would be applied in this work.

$$G_{vd}(z) = (1 - z^{-1})Z\left\{\frac{G_{vd}(s)}{s}\right\} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (2\text{-}42)$$

The digital model of a buck converter is a linear second-order Infinite Impulse Response(IIR) filter, containing four coefficients (a_1, a_2, b_1 and b_2), presented as (2-42). The four coefficients should be identified by parameter estimation.

2.5 Control of DC-DC Buck Converters

2.5.1 Current Mode Control and Voltage Mode Control

Robust control of Buck Converters is necessary to cope with external disturbances and system variations (e.g. load changes, reference voltage variations and abnormal operation of circuit components, etc.). Therefore, closed-loop control, containing negative feedback, is used here. There are two ways, selected depending on applications, of output controlling of Buck Converter: current control and voltage control. Peak Current Control (PCC) and Average Current Control (ACC) are the two most common ways applied in current mode control. As shown in Figure 2.7, S_1, the upper-bridge switch of Buck Converter, would be turned on by a clock signal, and off by an impulse produced when the detected inductor current signal equals a pre-determined limit. This limit can be a constant signal or a saw-tooth one. An R-S Flip-Flop is used to deal with the clock signal and the impulse signal from a comparator, and output switching signals to switches of Buck Converter. The most noticeable merits of PCC is its fast transient responses, same with Sliding Mode Control (SMC). Differently, however, PCC is also featured with fixed switching frequencies, which makes it able to be realised in hardware and its controller design much easier [100].

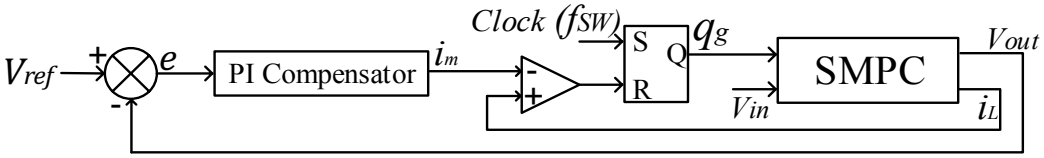


Figure 2.7 Peak Current Control Loop

ACC structure is similar to that of voltage control, apart from the reference signal being current rather than voltage (see Figure 2.7). It is known that the reaction time of the current control loop is faster than the voltage loop in converters so that it is more used as the inner loop of cascade control [101]. The voltage loop, however, is more widely chosen for three reasons: Firstly, it is simple to be designed as it can be a single loop while the current loop is often designed as the inner loop of the cascade control technique. Secondly, it may achieve high stability in virtue of plenty of noise margins. Thirdly, the impedance of its output is small. Nevertheless, voltage control also has drawbacks such as the slower response speed and input-voltage-dependent loop gains, etc. The selection of current or voltage control is mainly based on load applications. For digital electronics, regulated voltage outputs are required so that voltage control is preferred, while current control will be applied for LED lights that need the regulated current. In some cases, e.g. battery chargers, etc., both voltage and current outputs should be regulated [102]. Therefore, cascade control, in which the inner loop is current control and the outer one is voltage control, has been considered for robust control (see Figure 2.8) [103]. For example, in [100], PCC was used to build the inner current control loop with a PI controller for voltage control. As such, a current sensor, which can be viewed as extra resistances, should be added and the design complexity of control loops will increase. Therefore, in this project, only voltage control loop will be configured for further studies (see Figure 2.9).

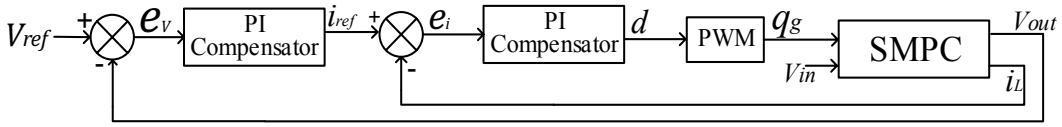


Figure 2.8 The Cascade Control with the Average Current Control (the Inner Loop)

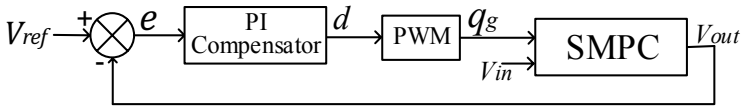


Figure 2.9 The Single Voltage Control Loop

2.5.2 Control Techniques Applied in SMPCs

There have been many control techniques introduced in SMPCs:

1. Dependent on the types of controllers, it is categorised as linear or nonlinear control.

2. Dependent on different modulation ways, it includes fixed switching frequency techniques, variable switching frequency approaches and hysteresis controls [102, 104-106].

3. Dependent on types of clock signals: autonomous or non-autonomous [101], etc.

The performance of these methods are often evaluated based on four main respects: 1. transient and steady-state response optimization under different input references (step, sine waves, etc.), 2. the ability of disturbance rejection, 3. sensitivities to parameter changes, and 4. Complexity/possibilities of hardware realisation [107, 108].

PWM, a typical example of fixed switching frequency techniques featured with ease of hardware implementation and high flexibility, has been predominantly applied in both academics and industries. The combination of the PID controller (the classical linear controller) and the PWM technique is the most commonly used technique for SMPCs control, due to its straightforward hardware realization: PWM is generated by a comparison between a ramp signal and a control signal produced from various kinds of compensators, e.g. PI controller, PID controller, etc. As the frequency and magnitude of the ramp signal are always preconfigured and fixed, the gist of the control system design becomes the selection of controller parameters. The benefits of applying PID controller on SMPCs was concluded in [109]:

1. Controller parameters can be tuned with 'trial and error'.
2. It is suitable for various topologies operating in both CCM and DCM.
3. Various tuning methods of PID controllers have been well explored.
4. The programmable hardware realisation of the PID controller becomes 'effortless' and flexible with the development of digital control techniques.
5. It may guarantee qualified transient behaviours in terms of fast responses, the eliminated Steady State Error (SSE) and the suppressed oscillations.

Variable-switching-frequency control techniques have also been deeply explored. Sliding Mode Control (SMC), for example, is a typical one particularly for high switching frequency applications. It can also be viewed as an autonomous control way as its switching decisions are based on pre-settled principles (or functions). This idea was originally proposed by Bilalovic in 1983. The technique equivalence between PWM and SMC was then proved by Sira-Ramirez in 1988 [110]. Compared with PID controllers which output linear control signals for PWM generation, SMC will directly produce nonlinear switching on/off demands [110]. As such, its theoretical structure is much simpler and processing speed is significantly faster than those of PID controllers, which makes SMC being commonly applied on the inner loop of multi-loop control techniques [109]. However, SMC also suffers some problems such as chattering [111-113], difficulties of being applied on complicated circuit topologies [109], the hardness of practical instalments [114-116] and insensitivity to parameter variation and uncertainties [117], etc. For dealing with these issues, SMC has been tried to be combined with other control approaches. For example, the combination between SMC and a PI controller proposed in [118]

was proved to be able to alleviate sensitivities of control systems to parameter changes and disturbances. SMC was also combined with neural network techniques in [114] which turn out to be highly adaptive for significant system uncertainties.

There are many comparative studies of various control techniques. In [108], Linear Averaged Controller (LAC) was compared with four different nonlinear methods, separately Feedback Linearizing Controller (FLC) based on linearization [119], Passivity-Based Controller (PBC) derived on passivation [120], SMC [121] and a combination of SMC and PBC [122]. It turns out that FLC is highly sensitive to parameter uncertainties. In [107], through comparing a PI controller, SMC and fuzzy logic techniques, it is proved that the sensitivities of SMC and fuzzy logic to voltage inputs and load variations are at the same level.

Conventional control strategies of SMPCs mainly focus on transient response improvements in normal operation circumstances that systems do not significantly or frequently vary [123-126]. However, as these wide applications of multi-rail power supplies may refer to some challenging working conditions, such as the occurrence of abrupt disturbances, frequent and (or) periodic load changes [13], malfunction of circuit components and system degradation over time and temperature fluctuations [9, 10], etc., the robustness of control to system changes is essential. Adaptive control, able to adjust controller parameters in real-time, can be a useful technique to minimize the impacts of system variations.

2.5.3 Digital Voltage Mode Control and its Hardware Implementation

Before digital control techniques, operations of SMPCs were used to be controlled through individually designed circuits, i.e., Analogue Control (AC). The high dependence on the circuit design of AC often limits it to achieve sophisticated control methods, as advanced control techniques would be practically realised by complicated circuits, which may lead to large sizes and high investments of control systems. As such, the efforts of system revisions or improvements would also be significant. Compared with analogue control techniques, digital control systems have been dominant. Because it can be achieved by programmable devices which may achieve various complicated control methods with low costs and results in high flexibility for parameter configurations and adjustments, hardware implementations and revisions, and control method changes [127], etc. Also, programming all compensation decisions in processors makes control parts being more insensitive to external disturbances. However, digital control still commonly suffers some problems: firstly, the digital control loop takes longer reaction time due to delays during program processing, and its narrower loop bandwidth (than that of AC) prevents it from coping with high-frequency elements in control error signals. Secondly, aliasing and quantization effects, separately caused by limited sampling

rates and finite resolution of ADC, may bring inaccurate compensations of error signals. Also, the limited code processing capability of processors may prohibit digital control from achieving highly centralised control approaches, which requires independent loops for many individuals realised by a single processor. Nevertheless, as processing performances of practical equipment are rapidly improving, digital control techniques are still preferred and used in this project.

Figure 2.10 shows the process of a voltage control loop of Buck Converter, divided into an analogue part (Buck Converter, output sensors and gate drivers, etc.) and a digital part processing sampled voltage outputs and providing DPWM control signals. The two parts are linked with an A/D conversion block and a DPWM module (D/A conversion). The digital part is typically realised in programmable devices, such as Digital Signal Processors (DSPs) and Field-Programmable Gate Array (FPGA). DSP is applied in this project as all necessary modules, such as DPWM and ADC, etc., already exist for personalised setups. Apart from the project-related configurations, such as sampling frequency, controller parameters and system identification algorithm programming, etc., there would be some pre-settled configurations for topology and processor protections. For example, a 1μs deadtime between two switches (S_1 and S_2) of a Buck Converter is prepared, and a voltage-divider circuit is added before sampling in case the input of ADC extends its highest voltage receiving tolerance (3.3V of the TMS320F28335 DSP). Besides, there are some requirements for the ADC module of DSPs: Firstly, following in the Nyquist-Shannon sampling theorem, the cut-off frequency and anti-aliasing filter of ADC should be lower than half of the sampling frequency. Secondly, its frequencies of A/D conversion and sampling should be high enough to achieve qualified control performance [102].

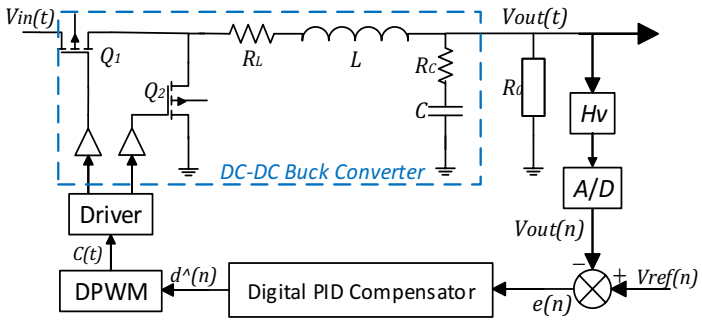


Figure 2.10 The Digital Control Loop of Buck Converter

As shown in Figure 2.10, the sampled output voltage signal [$V_{out}(n)$] is compared/subtracted with the reference signal [$V_{ref}(n)$] to acquire control error signals [$e(n)$] which are then fed to the compensator, carefully tuned to minimise the error signal through producing control signal [$d(n)$] with duty cycle information. The control signal would be

compared with carrier waves, of which magnitude and frequency (switching frequency) are preconfigured, in the DPWM module to generate PWM signal controlling switching actions of the two switches (S_1 and S_2) of the Buck Converter. The magnitude of PWM outputted from PWM module [$c(n)$] of DSP is normally lower than gates voltage of S_1 and S_2, so that a gate driver is used for amplifying PWM signals. The frequency of the clock signal producing DPWM or saw-tooth wave could be 100MHz when the switching frequency of SMPCs (or PWM frequency) equals 100kHz to guarantee enough resolutions of PWM, while the higher the frequency is, the more hardware investments would be taken [102]. In this project, the clock frequency of the applied DSP (TMS320F28335) is 150MHz and the switching frequency is 20kHz. In every switching cycle, there are 7500 clock pulse generated, which can well balance the needs of fast fault corrections and hardware cost-saving.

2.5.4 Digital PID Controller Tuning through Pole-Zero Cancellation Approach

Due to the afore-mentioned merits of PID controllers, it will be applied for SMPCs control in this project. However, as it is also viewed as 'not being able to achieve more precise control or faster response speed' [109], various auto-tuning/self-tuning approaches for PID controllers have been fully derived, e.g. Ziegler-Nichols tuning [128], Kappa-Tau method [129], Internal Model Control (IMC)-PID auto-tuning [130] and data-based Fictitious Reference Iterative Tuning (FRIT) method [131, 132], etc. Almost all of the tuning methods/adaptive control techniques of PID controllers (automatically updating K_p, K_i and K_d) are based on transfer functions of open-loop plants (Buck Converters here). As such, applying a PID controller for Buck Converter control in this project would result in system-identification-based adaptive control realisation being easier.

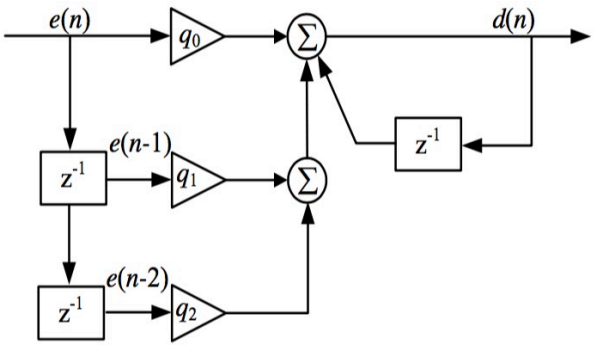


Figure 2.11 The Structure of PID Controller in the Digital Domain [133]

Figure 2.11 shows the structure of a PID controller, which contains three individual control signal generation paths (Proportional, Integral and Derivative) superposed, in Laplace and digital domains. Inputs of the controller are typically error signals [$e(t)$ or $e(n)$], and outputs

control signals $[d(t)$ or $d(n)]$ (see Figure 2.9 and 2.10). The transfer functions of a PID controller in Laplace and digital domains, $G_C(s)$ and $G_C(z)$, are respectively expressed as (2-42) and (2-43).

$$G_C(s) = \frac{D(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad (2\text{-}42)$$

$$G_C(z) = \frac{D(z)}{E(z)} = K_P + \frac{K_I}{1-z^{-1}} + K_D(1-z^{-1}) = \frac{(K_P+K_I+K_D)-(K_P+2K_D)z^{-1}+K_D z^{-2}}{1-z^{-1}} \quad (2\text{-}43)$$

According to (2-43), the time series of a PID controller can be presented as follows:

$$d(n) = (K_P + K_I + K_D)e(n) - (K_P + 2K_D)e(n-1) + K_D e(n-2) + d(n-1) \quad (2\text{-}44)$$

where, E in (2-43), or e in (2-44), are the error signal produced by reference signal subtracting feedback signal, $V_{ref} - V_{sensed}$ in Figure 2.9 if voltage control is applied. D in (2-43), or d in (2-44), are the output of the controller. K_p, K_i and K_d (K_P, K_I and K_D) are respectively proportional, integral and derivative gains of PID controller. According to (2-44), q_0, q_1, and q_2 in Figure 2.11 equal $K_P + K_I + K_D$, $K_P + 2K_D$ and K_D respectively.

The values of the three parameters of a PID controller, K_p, K_i and K_d, would affect stability and transient responses of control systems in terms of phase margins and gain margins in Bode Plot, or overshoot, settling time and steady-state error, etc. in the time domain. The proportional control path would proportionally amplify or attenuate the magnitude of error signals $[e(n)]$. Therefore, a larger K_p could boost system transient responses, which result in shorter peaking time and severer overshoot, etc. The most noticeable merit of the integral control path is that it can eliminate Steady State Error (SSE) which the proportional controller, at most, can only reduce to a small range. Besides, it also can undermine the effects of high-frequency noises on systems because it has a similar structure to low-pass filters. However, as (2-42) shows, the integrator would bring a pole to its corresponding open-loop transfer function so that the stability of entire control systems should be re-evaluated. In the frequency domain (Bode Plot), the integral part could increase the magnitude of attenuation slope in the low-frequency section, a decreased phase margin, therefore, can be viewed. This phase-lag property is typically reflected as a prolonged settling time (or severer oscillations) in transient responses. The wind-up phenomenon is also a probable issue brought by large values of K_i. As such, an integral path often ends with a saturation block to limit output control signals. As opposed to the integral part, the derivative control path is a lead controller that could improve system stabilities, as it may be predictive to error magnitude by generating derivations of control error signals. The facilitated stability would be expressed as suppressed overshoot and shortened settling time in transient responses. As shown in Bode Plot, the derivative controller could make its magnitude

attenuation curve plainer. Consequently, a widened phase margin can be seen, as well as the increased magnitude in the high-frequency area, which means control systems with derivative controllers would be very sensitive to high-frequency noises. Therefore, the use of a derivative control path is often carefully considered, or it can be turned on only for partial periods of control processes. Overall, the PID controller, as a kind of Lead-Lag compensator, could achieve acceptable transient responses and efficiently minimize control error signals, as long as the three parameters are reasonably selected, so can the PI controller.

If digital control is required, the PID controller can be tuned in the digital domain directly, or in the frequency domain firstly and then transformed into the digital domain. Direct methods are viewed to be more advanced in resulting in better transient responses and stronger stability because approximations during s-to-z transformation could lead to impreciseness in the parameter tuning results of digital controllers [134-138]. However, the advantages of direct tuning ways are typically reflected when system variations or parameters are accurately known [135, 139, 140]. Directly tuning digital PID controllers allows controller parameter updates to be based on real-time system identification results [141-145]. As such, the entire control system could keep high stability and robustness to system variations in real-time. While, without updates of controller parameters during system operations, both direct and indirect tuning ways are based on linearized models of SMPCs. As the inaccuracy caused by s-to-z transformation and quantization, etc. is insignificant, compared with that brought by linearization, there might not be noticeable differences between the two tuning ways. Pole-zero cancellation [146-149] and pole placement [144, 150-152] are the most two methods commonly used, and the former one is proven in [102] to be highly reliable. There have been many comparative studies investigating the performance of commonly used s-to-z transformation methods [138, 153-156], including Backwards Difference, Bilinear, Impulse Invariant, Step Invariant and Zero-Pole Matching, etc. It turns out that transformation distortion can be significantly alleviated as long as these methods are properly selected. For example, Backwards Difference and Bilinear are preferred at high sampling frequency when application simplicity is required, while Impulse Invariant won't be chosen for step input tests and vice versa. Overall, in this project, the PID controller will be tuned by pole-zero cancellation approach in frequency domain first and Zero-Pole Matching method applied for s-to-z discretization due to its precise transformation of poles and zeros.

The process of pole-zero cancellation has been introduced in many papers [147, 148, 154]. It can be seen that the transfer function of buck converters (see (2-30)) contains two poles, which would be cancelled by two zeros in PID controllers, seen in (2-45) and (2-46). The

cancellation is to set up the two zeros at the same locations with the two poles. the frequency of the two zeros, ω_{n_c}, should equal natural frequency, ω_n. To simplify the tuning process, the effects of the zero of buck converters caused by ESR of the output capacitor, R_C, won't be considered. As such, the open-loop transfer function can be expressed as (2-46) and the frequency of zeros can be calculated as (2-47) shows.

$$G_c(s) = \frac{D(s)}{E(s)} = \frac{K_d s^2 + K_p s + K_i}{s} = G_{co} \cdot \frac{\left(\frac{s}{\omega_{n_c}}\right)^2 + \frac{2\xi_c s}{\omega_{n_c}} + 1}{s} \tag{2-45}$$

$$L(s) = G_c(s) \cdot G_{vd}(s) \cdot H_s(s) = G_{co} \cdot G_{DC} \frac{\left(\frac{s}{\omega_{n_c}}\right)^2 + \frac{2\xi_c s}{\omega_{n_c}} + 1}{s\left[\left(\frac{s}{\omega_n}\right)^2 + \frac{2\xi s}{\omega_n} + 1\right]} \cdot H_s(s) = \frac{G_{co} \cdot G_{DC} \cdot H_s(s)}{s} \tag{2-46}$$

$$\omega_{n_c} \approx \omega_n = \sqrt{\frac{R + R_L}{LC(R + R_C)}} \approx 3727.19 rad/s \tag{2-47}$$

Here, G_{co}, ξ_c and ω_{n_c} are respectively DC gain, damping ratio and natural frequency of PID controller, $H_s(s)$ is the transfer function or loop gain of feedback. The DC gain, which could directly influence phase margin (stability) in frequency responses, can be acquired based on the DC gain of open-loop systems, G_{DC}, and the required loop bandwidth, f_b, which is typically one-tenth of sampling frequency (f_s) [147, 148, 157]:

$$G_{co} = \frac{2\pi f_b}{G_{DC} \cdot H_s(s)} = \frac{2\pi f_s}{10 G_{DC} \cdot H_s(s)} \tag{2-48}$$

According to the above-introduced process, the rigid pole-zero cancellation should be based on accurate system awareness, such as circuit component values and appropriate modelling, etc. As such, controller parameters might not provide optimized control performance if noticeable coefficient changes of transfer functions of systems occur, as these changes would lead to variations of pole locations which won't be matched with zeros of controllers anymore. Without adaptive adjustment of controller parameters corresponding to system variations, this problem can be solved by selecting a fixed value of damping ratio, ξ. damping ratio is normally chosen from 0.6 to 1 to guarantee a good balance between overshoot magnitude and response speed. In this project, $\xi_c = 0.7$.

If the circuit component values, sampling frequency and switching frequency configured in **2.4.2** are still used here, $H_s(s) = 0.5$, $\omega_{n_c} \approx \omega_n \approx 3727.19 rad/s$, and $G_{co} = 800\pi$. Hence, the transfer function of the PID controller in the frequency domain is calculated in (2-49). After discretization with zero-order-hold (ZOH) effects, its digital version, shown in (2-50), is acquired.

$$G_c(s) = \frac{D(s)}{E(s)} = G_{co} \cdot \frac{\left(\frac{s}{\omega_{n_c}}\right)^2 + \frac{2\xi_c s}{\omega_{n_c}} + 1}{s} = \frac{1.809 \times 10^{-4} s^2 + 0.944 s + 2513}{s} \tag{2-49}$$

$$G_c(z) = \frac{D(z)}{E(z)} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} = \frac{4.121 - 7.169 z^{-1} + 3.174 z^{-2}}{1 - z^{-1}} \tag{2-50}$$

$$\begin{cases} q_0 = K_P + K_I + K_D \\ q_1 = -(K_P + 2K_D) \\ q_2 = K_D \end{cases} \Rightarrow \begin{cases} K_P = 0.821 \\ K_I = 0.126 \\ K_D = 3.174 \end{cases} \tag{2-51}$$

Then the closed-loop transfer function of the control system, in the digital domain, is:

$$G_{CL}(z) = \frac{G_c(z) G_{vd}(z)}{1 + G_c(z)} = \frac{0.9304 - 1.158 z^{-1} - 0.08494 z^{-2} + 0.3549 z^{-3}}{1.93 - 4.074 z^{-1} - 2.781 z^{-2} - 0.5952 z^{-3}} \tag{2-52}$$

Here, $G_{vd}(z)$ is the digital transfer function of the buck converter shown in (2-42).

Figure 2.12 is the bode plot transfer functions of PID controller, buck converter and the closed-loop control system. It shows that, after applying the PID controller tuned as the above-introduced process, the control system could achieve a phase margin of 72.2º and gain margin of Infinity that indicate qualified stability.

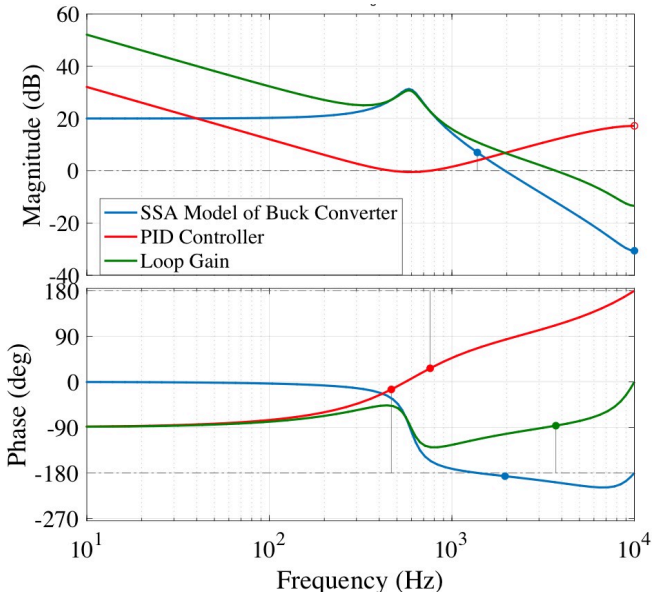
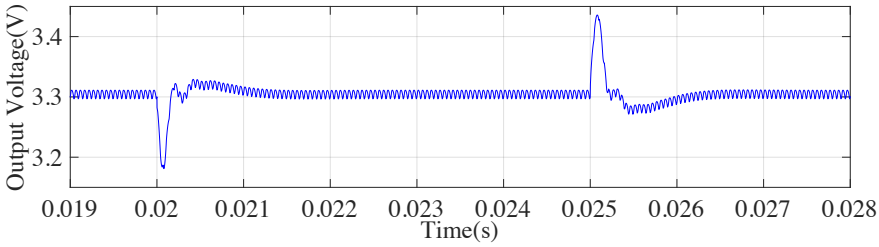
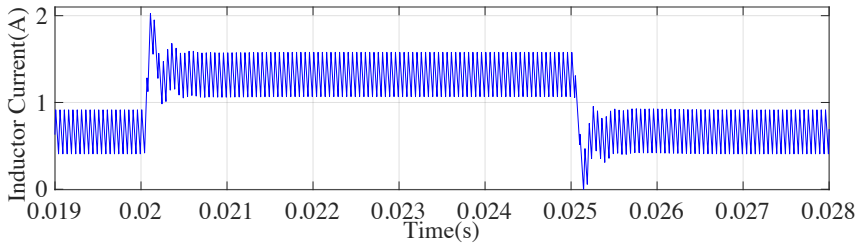


Figure 2.12 The Bode Plot of PID Controller, Buck Converter and the Corresponding Closed-Loop Control System

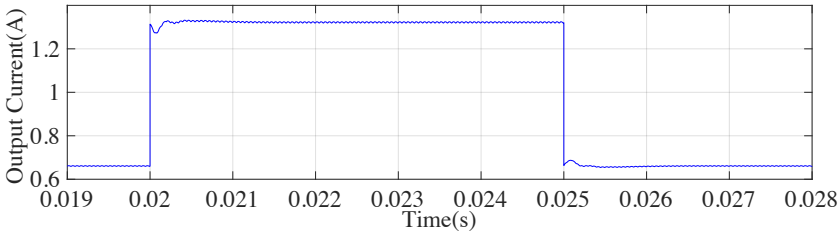
The controller is then applied to deal with sudden load changes for transient behaviour tests. When the output of the buck converter is still set up as 3.3V while the resistance load will change from 5Ω to 2.5Ω at 0.02s, and back to 5Ω at 0.025s (see Figure 2.13). This indicates the digital compensator is well designed, as a small overshoot, approximately 3%, in voltage output is produced under a 50% load change and the output of the buck converter is quickly recovered to a steady-state.



(a)



(b)



(c)

Figure 2.13 The Transient Behaviour Test for the Tuned PID Controller

2.6 Chapter Summary

This chapter presents the topology, the operation, the SSA model and several commonly-used control techniques of Buck Converters. Besides, multi-rail power conversion architectures comprising several Buck Converters, their applications and the control challenges caused by various system variations are introduced. For the balance between the needs of model accuracy and low model complexity, SSA modelling way is selected and the corresponding transfer function is validated to be accurate enough for controller tuning. Then, the process of the pole-zero cancellation approach for a digital PID controller for a simulated Buck Converter is demonstrated in a detailed way. Through the Bode Plot of the control system and operating it under load changes, the designed controller is proven to be robust enough to cope with abrupt system variations.

Chapter 3 System Identification of Switch Mode Power Converters

3.1 Introduction

This chapter presents a comprehensive review, divided into three parts, of recent studies about System Identification (SI) of Switch Mode Power Converters (SMPCs). Firstly, the currently dominant approaches of linear SI are categorised into non-parametric, parametric and hybrid techniques, and individually introduced. Through comparison among these techniques, the reasonability of applying parametric approaches on adaptive control is proofed. Therefore, in the second section, the four steps, typically followed to achieve iterative parameter estimation techniques, are listed and separately described. Finally, to solve the commonly suffered problem (priority conflict of rails for being identified) during real-time parameter estimation of multi-rail power converters, several efficient task prioritization approaches adopted in other areas are investigated, their applicability into simultaneous identification of multiple rails is also discussed.

System identification, as a useful tool to get a mathematic model of an identified system through system information collection (inputs and outputs) [158, 159], has been widely applied to improve control performance [160-162], manage energy consumption [163] and diagnose system operation conditions [164-167], etc. Based on the processes of SI, it can be divided into off-line SI and online SI [141, 168-170]. Online SI, as it requires results updated during system operation, is commonly used in real-time applications, e.g. adaptive control, health operation monitoring, fault detections [15, 164-167], etc. As offline SI can be achieved when systems are not working (the collection and storage of necessary information will complete before SI), its results, e.g. transfer function, etc., will provide dynamic behaviours of open-loop objectives for controller tuning [169], to match some transient behaviour requirements of closed-loop control systems (overshoot, rising time, etc.) [141, 171, 172]. According to the pre-knowledge of identified systems, they can be viewed as a black box and grey box [159]. Black box indicates there is nothing known before identification, such as the transfer function order of the model, the number of parameters, circuit components used in the topologies, etc., while for grey boxes, the model has been selected through system modelling [173, 174]. The model order of the identified system has been known, and the number of its transfer function coefficients is known. Therefore, grey box methods usually refer to circuit analysis, linearization of nonlinear systems [175, 176] and model fitting, etc. SI also could complete as long as appropriate guesses of coefficient values are acquired, unlike the black box which first needs to figure out the most

fitted system models with sampled data [177, 178]. Consequently, the grey box may get a faster identification speed [15].

3.2 Approaches of SI

According to two reviews, [15] and [158], which introduced SI techniques adopted in SMPCs in a detailed way, these techniques can be divided into two categories: parametric approaches and non-parametric approaches. Both need frequency-rich perturbation signals for system excitation. The specific categories of SI are presented in Figure 3.1. Parametric estimation, including iterative or non-iterative schemes, can be taken places when systems are operating and the corresponding results can update in real-time, while non-parametric one may require open-loop control, transient response acquisition and off-line analysis [15, 21-23, 159, 160]. Hybrid approaches, combining parametric and nonparametric schemes, have also been developed for comprehensive system investigation.

$$System\ Identification \begin{cases} Parametric\ Estimation \begin{cases} Parametric\ Based\ Modelling\ Scheme \\ Iterative/Recursive\ \ Scheme \\ Non-Iterative\ Scheme \end{cases} \\ Non-Parametric\ Estimation \begin{cases} Correlation\ Analysis\ Scheme \\ Network\ Analyser\ Scheme \\ Power\ Spectrum\ Desity\ Scheme \end{cases} \end{cases}$$

Figure 3.1 Categories of System Identification Techniques

3.2.1 Non-parametric Approaches

Non-parametric approaches, for a long period, were always viewed as 'only can be applied on off-line fields'. However, nowadays, although parametric approaches are still preferred for many online applications, such as health monitoring and fault detection of DC-DC power converters, non-parametric approaches are also tried to be used in these areas [161-165]. For example, for health monitoring and acquiring circuit component values, a frequency response-based pole-zero mapping way is derived in [161]. Other examples are in [166, 167], where non-parametric approaches are investigated to realise adaptive control and auto-tuning techniques. The pursuits of recent studies on improving non-parametric approaches are as same as that of parametric ones: alleviating computational complexity of identification processes [22, 162], enhancing disturbance rejection ability and noise insensitivity [159, 168]. Besides, improvements of perturbation signals are necessary, as the large magnitude of them, which could fully excite systems though, would disturb system outputs (strictly regulated voltages). To minimise the negative effects, new perturbation signals, such as blue noises, have been examined to substitute the conventionally used one [Pseudo-Random Binary Sequences

(PRBS)] [15, 168, 169]. Other signals proposed to be used in non-parametric identification include Inverse Repeat Binary Sequence (IRBS) and Maximum Length Binary Sequence (MLBS) [170, 171].

Compared with parametric SI which owns the typical merits of being able to acquire each value of passive components, its non-parametric counterpart is more advanced in the easy design of the control loop without knowing exact plant information. There are three techniques typically involved in non-parametric approaches: correlation analysis, frequency response analysis [172, 173] and Fourier or spectrum analysis [158]. The basic process of non-parametric SI can be concluded as two steps. The first step is to derive impulse responses [22] through applying cross-correlation analysis on sampled input and output signals [168]. The second one is to acquire frequency responses by computing Fourier Transform (FFT in [22, 162]) based on the impulse responses. Typical examples of the application of the two-step process are in [22], where a forward converter is identified. Another scheme is to use Discrete Fourier Transform (DFT) with sine wave excitation signal to directly get frequency responses, which makes correlation analysis unnecessary [161, 174]. However, this method may be further improved before being applied in real time occasions as it suffers from severe computational burden than the first does. As reviews, [158] and [15], suggest, non-parametric approaches have the merits of ease of implementation and being unnecessary to acquire prior information about identified systems, therefore, they have been widely adopted for operation supervision and fault detection [163-165]. However, they suffer long convergence time (as most of them are black-box techniques [97]), identification inaccuracies [caused by quantization in Analogue-to-Digital Conversion (ADC) [141] during low-resolution occasions, domain transformation from Laplace to Digital, etc.] and high noise/disturbance sensitivities, etc. For example, compared with real-time parameter estimation using Recursive Least Square (RLS) which may only spend 3ms for getting real values under a 100kHz sampling frequency, non-parametric approaches would take 180ms [22]. Besides, their realization process needs a significant amount of data saved for correlation analysis [97]. Moreover, as the result update of non-parametric approaches is not iterative, it is difficult to find a highly flexible pace of updating control parameters, necessary in adaptive control. Therefore, there are many papers looking for improvements. For higher estimation accuracy, both pre-emphasis and de-emphasis filters were added in the identification process in [162], and the author of [168] proposed a windowing method, which can set aside a phase delay in data sampling for perturbation signal revision. Furthermore, [162] adopts a hardware-efficient approach to attenuate the influence of quantization in ADC and a fractional decade spectral window to smooth frequency responses. In [168], signals, typically

duty cycle and output voltages/currents, are sampled in a high density to circumvent discontinuity in collected data. In [23], the impreciseness of sampling and sensing is derived through a fuzzy density method. For disturbance effect reduction, [23] suggests applying circular correlation analysis on transfer function acquirement of SMPCs. The same idea of developing correlation techniques on SI can be seen in [172] and [162].

3.2.2 Parametric Approaches

Parametric SI is also called parameter estimation. It can be accomplished by two schemes: iterative and non-iterative schemes [175]. SI which does not need iterations is called non-iterative parameter estimation. According to [147, 176, 177], the application of non-iterative approaches in SMPCs contains converter identification and the corresponding controller tuning. Results acquired from this scheme typically are values of corner frequency and quality factor of systems, rather than new guesses of candidate model coefficients typical results of iterative approaches. Systems could be excited by adding oscillations with certainly configured frequencies into voltage outputs of SMPCs during the steady-state period [Limit Cycle Oscillation (LCO) method] [28, 136, 147] or start-up period (forward relay-feedback method) [136, 176, 177]. Non-iterative approaches are often featured of easy implementation, though their identification accuracy is questionable [178].

Iterative parametric estimation aims to figure out the optimal model parameters (or coefficients of transfer functions) describing systems. Challenges that this scheme often encounters are concluded as: Firstly, before carrying out estimation procedures, prior information about identified systems, such as the order of models (transfer functions), the number of parameters and even circuit component values, etc., should be known. Secondly, the identification process is sensitive to noises so that sampled data should be carefully filtered. Thirdly, algorithms currently used for parameter estimation of SMPCs are computationally costly as most of them involve multiplications, inversions, or even divisions of matrixes. When orders of candidate models become higher, the issue of significant computational burden would be severer, which often brings difficulty or extra investments in the corresponding hardware implementation. Nevertheless, this approach can achieve fast identification speed and acquire coefficient values of system models which can be directly used for controller auto-tuning. Moreover, parametric approaches using iterative schemes may iteratively update estimation results at sampling frequency during system operation. These merits make parameter estimation ideal tools for real-time applications such as controller auto-tuning, operation monitoring and fault detection [162], etc. Adaptive filters, using steepest-descent methods to minimize cost function, $J(\omega)$ in (3-1), to iterate an optimal parameter vector, have been diffusely utilized in

online SI (iterative parameter estimation) [179] of linear systems [15, 180-183]. Convergence time, computational cost and estimation accuracy are three main considerations to evaluate the performance of adaptive filters (recursive algorithms).

$$J(\omega) = E|y - u\omega|^2 \tag{3-1}$$

y is the output signal. u is regression vector (regressor). ω is the updated vector guess of parameters.

The simplest algorithm used for SI might be Least Mean Square (LMS) derived from steepest-descent methods of which correction term comprises of step-size parameter and direction vector, shown in (3-2).

$$\omega_i = \omega_{i-1} + \mu\left[R_{yu} - R_u\omega_{i-1}\right], \omega_{-1} = initial\ guess \tag{3-2}$$

Here, ω_i is the updated estimation vector at the ith iteration event, ω_{i-1} is the one at the $(i-1)th$ iteration event. μ is a constant step size. $R_{yu} = \mathrm{E}yu^*$, which is the cross-covariance vector of y (scalar-valued output) and u (regressor). $R_u = \mathrm{E}uu^*$, which is a positive-definite covariance matrix of u. The updated direction vector, $R_{yu} - R_u\omega_{i-1}$ in (3-2), equals the negative conjugate-transpose of the gradient vector of the cost function at ω_{i-1}, i.e.,

$$R_{yu} - R_u\omega_{i-1} = -[\nabla\omega \cdot J(\omega_{i-1})]^* \tag{3-3}$$

By replacing the random variable with observations and approximating R_{yu} and R_u with instantaneous values, i.e. $\widehat{R_{yu}} = y(i)u_i{}^*, \widehat{R_u} = u_i{}^*u_i$, the gradient vector $\nabla\omega \cdot J(\omega_{i-1})$ can be viewed as:

$$-[\nabla\omega \cdot J(\omega_{i-1})]^* = y(i)u_i{}^* - u_i{}^*u_i\omega_{i-1} = u_i{}^*[y(i) - u_i\omega_{i-1}] \tag{3-4}$$

Then, the LMS algorithm can be described as:

$$\omega_i = \omega_{i-1} + \mu u_i{}^*[y(i) - u_i\omega_{i-1}], i \geq 0, \omega_{-1} = initial\ guess \tag{3-5}$$

μ is a positive step-size.

Another algorithm in the LMS family is Step Adaptive Least Square (SALS), applied for SI of a single-phase buck converter in [184] and a two-phase converter in [26]. The two papers also selected Dichotomous Coordinate Descent (DCD)-RLS and Batch Least Square (BLS) for performance comparisons. It shows that SLAS features high noise insensitivity despite a longer convergence time than DCD-RLS [15]. However, SALS was applied in an open-loop and extra memory is needed for pre-processing sampled data.

$$f_c = \frac{f_s\sqrt{\log b_2^2 + 4\cos\left(\frac{-2b_1}{2\sqrt{b_2}}\right)^{-1}}}{4\pi} \tag{3-6}$$

$$\omega_i = \omega_{i-1} + 2\mu u_i^*[y(i) - u_i\omega_{i-1}] \tag{3-7}$$

f_c is the corner frequency, f_s is sampling frequency, b_1 and b_2 are numerator coefficients of transfer function representing buck converters.

For improved performance, the regularized Newton's recursion with a constant regularization sequence, i.e.$\epsilon(i) = \epsilon$, and a constant step-size sequence, i.e.$\mu(i) = \mu$, can be applied in steepest-descent algorithms, shown as (3-8).

$$\omega_i = \omega_{i-1} + \mu[\epsilon I + R_u]^{-1}\left[R_{yu} - R_u\omega_{i-1}\right] \tag{3-8}$$

By respectively replacing the quantities of $(\epsilon I + R_u)$ and $(R_{yu} - R_u\omega_{i-1})$ with the instantaneous approximations, $(\epsilon I + u_i{}^* u_i)$ and $u_i{}^*[y(i) - u_i\omega_{i-1}]$, the corresponding stochastic-gradient recursion would be written as:

$$\omega_i = \omega_{i-1} + \mu[\epsilon I + u_i{}^* u_i]^{-1} u_i{}^*[y(i) - u_i\omega_{i-1}] \tag{3-9}$$

According to [185], $[\epsilon I + u_i{}^* u_i]^{-1} u_i{}^* = \frac{u_i{}^*}{\epsilon + \|u_i\|^2}$, Normalized Least Mean Square (NLMS) recursion is then described as

$$\omega_i = \omega_{i-1} + \frac{\mu}{\epsilon + \|u_i\|^2} u_i{}^*[y(i) - u_i\omega_{i-1}], i \geq 0, \omega_{-1} = intial\ guess \tag{3-10}$$

ϵ is a small positive parameter.

NLMS, using a stochastic-gradient approximation of Newton's method, is proofed to be more efficient in terms of the shorter convergence time in real-time SI than LMS [186]. However, as models/transfer functions of the commonly used SMPCs, such as buck/boost converters, are second-order systems that may contain four coefficients for identification, the afore-mentioned algorithms will take ages to update estimation results. Moreover, there is a risk of instability when identification processes encounter disturbances if the step size is not appropriately set up [184]. As such, other adaptive filters are investigated and introduced in parameter estimation of SMPCs, including Fast Affine Projection (FAP), Kalman Filter (KF), Recursive Least Square (RLS, the most classical one) and its different variants [15, 25, 26, 28, 29], etc.

In Affine Projection (AP), R_u and R_{yu}, shown in (3-8), are replaced by the mean of several previous instantaneous values rather than the current one only, i.e.,

$$\widehat{R_u} = \frac{1}{N}\left(\sum_{j=i-N+1}^{i} u_j{}^* u_j\right),\ \widehat{R_{yu}} = \frac{1}{N}\left(\sum_{j=i-N+1}^{i} y(j) u_j{}^*\right) \tag{3-11}$$

Thus, $\widehat{R_u} = \frac{1}{N} U_i{}^* U_i$, $\widehat{R_{du}} = \frac{1}{N} U_i{}^* Y_i$, $U_i \triangleq \begin{bmatrix} u_i \\ u_{i-1} \\ \vdots \\ u_{i-N+1} \end{bmatrix}$, $Y_i \triangleq \begin{bmatrix} y(i) \\ y(i-1) \\ \vdots \\ y(i-N+1) \end{bmatrix}$, where N is a positive integer that defines the number of historical steps considered in the current iteration,

M is the number of coefficients that need to be identified. Therefore, $U_i \in \boldsymbol{R}^{N\times M}$, $Y_i \in \boldsymbol{R}^{N\times 1}$, $\omega \in \boldsymbol{R}^{M\times 1}$. Then Newton's recursion (3-8) becomes

$$\omega_i = \omega_{i-1} + \mu[\epsilon I + U_i{}^* U_i]^{-1} U_i{}^* [Y_i - U_i \omega_{i-1}] \tag{3-12}$$

Given that $[\epsilon I + U_i{}^* U_i]^{-1} U_i{}^* = U_i{}^* [\epsilon I + U_i U_i{}^*]^{-1}$ [185], AP is described as follows:

$$\omega_i = \omega_{i-1} + \mu U_i{}^* [\epsilon I + U_i U_i{}^*]^{-1} [Y_i - U_i \omega_{i-1}] \tag{3-13}$$

FAP, as an improved variant of AP for computational burden alleviation, can be described as follows:

$$e_i = Y_i - U_i \omega_{i-1} \tag{3-14}$$

$$G_i = U_i U_{i-1}{}^* \tag{3-15}$$

$$\gamma_i = z_i + G_i \varepsilon_i = U_i \omega_{i-1} \tag{3-16}$$

$$z_i = U_i \omega_{i-2} \tag{3-17}$$

$$R_i = [\epsilon I + U_i U_i{}^*] \tag{3-18}$$

$$\omega_i = \omega_{i-1} + \mu U_i{}^* R_i{}^{-1} e_i \tag{3-19}$$

G_i, R_i and z_i are intermediate matrix variables, derived for being repetitively used because in (3-13) some elements have been calculated in previous iteration events. As a result, computational burden per iteration is relieved and time spent on finishing per iteration process is shortened. In [179] and [187], FAP is adopted in adaptive control for performance comparison with RLS, in which a less convergence time of using FAP is presented.

KF family mainly contains three versions: classical KF for linear systems, extended KF for nonlinear systems and unscented KF for improved performance. Due to its advantages of being efficient and robust anti-interference, this family has been applied in target tracking, dynamic positioning, digital image processing, pattern recognition and navigation [188], etc. In [189] the classical KF was proposed to estimate coefficients of a linear second-order transfer function expressing a synchronise buck converter. It turns out that KF can achieve a shorter convergence time and higher estimation accuracy than exponential RLS (ERLS) does. As another typical adaptive filter with the least-mean-square criteria, its iterative update process includes four steps: Kalman gain (G_i), new parameter guess (ω_i), estimation dispersion (P_i) and covariance matrix project (P_i^+). The corresponding equations are listed below:

$$G_i = P_{i-1}^+ u_i^* \left[r + u_i P_{i-1}^+ u_i{}^*\right]^{-1} \tag{3-20}$$

$$\omega_i = \omega_{i-1} + G_i [y(i) - u_i \omega_{i-1}] \tag{3-21}$$

$$P_i = P_{i-1}^+ (I - G_i u_i) \tag{3-22}$$

$$P_i^+ = P_i + T \tag{3-23}$$

$P(0) = g \times \boldsymbol{I}$ and $\omega_{-1} = 0$,where $\boldsymbol{I}$ is an $M \times M$ identity matrix, g is large number, r is scalar > 0, T is $diag[T_{11}, T_{22}, .., T_{MM}]$ added in covariance matrix project (P_i^+). As [190] suggests, these diagonal elements in T are beneficial to random changes of numbers in P_i^+. In addition, an attribute of KF, the linear increase in P, may result in KF operating without perturbation effects for longer time than ERLS [189].

Apart from the afore-listed adaptive filters, other algorithms that have been introduced in SI of buck converter include DCD-RLS [28, 141], Biogeographical Based Optimization (BBO) combined with RLS [25]. In summary, finding an ideal algorithm that can achieve one or more requirements (less convergence time, higher estimation accuracy, lower computational burden, stronger disturbance rejection ability and lower noise sensitivity, etc.) has been always one of the populist studies in parametric SI.

3.2.3 Hybrid Approaches

Hybrid identification approaches, combining parametric and non-parametric ways, may acquire both frequency responses and transfer function coefficients although open-loop tests are needed [167]. The differences within hybrid approaches are mainly the sequence of carrying out parametric and non-parametric processes. In [167], frequency response acquirement is firstly done while it starts after iterative parameter estimation completes in [16]. The application of hybrid techniques is the same as that of parametric and non-parametric approaches (health monitoring, adaptive control and system aging detections, etc.), however, they always suffer high computational complexity due to their combination instinct.

3.2.4 Summary and Discussion

This section reviews different SI approaches applied in SMPCs. As many studies investigated, although more comprehensive knowledge can be acquired through using non-parametric approaches, these techniques require an open loop, extra data memories and cannot be well cooperated with real-time applications. Parametric approaches have the advantage of being instinctively iterative, ideally used for real-time applications, including adaptive control, operation fault detection, etc. Firstly, sampling and calculation capabilities of hardware equipment (FPGA, Digital Signal Processors, etc.) being upgrading are alleviating high computational cost issues suffered during algorithm implementation. Secondly, techniques that acquiring circuit component values according to estimated parameters of the transfer function are rapidly improving, which makes parametric SI an efficient tool for health monitoring. Besides, only on some particular occasions, hybrid ways can be used due to their tremendous computational complexity produced by compositing two identification approaches. Due to the

increased popularity of adaptive control techniques coping with frequent system variations during system operations, parametric SI has been widely used and the relative studies have become more comprehensive and deeper. Researchers have always urged to introduce or derive more efficient algorithms in terms of shorter convergence time, lower computational complexity and higher estimation accuracy. Algorithms that can be used for parameter estimation have not been fully explored. Correspondingly, RLS, as the most classic algorithm used in SMPCs due to its high stability and qualified performance, has been dominantly used in parameter estimation of linear models describing SWPCs. Besides, almost all new-algorithm investigation studies would compare their proposed algorithms with RLS to demonstrate improvements [15, 26, 179, 189]. Therefore, in this project, RLS will be selected for parameter estimation of multi-rail power converters to validate the proposed two computational reduction approaches.

3.3 Steps of Iterative Parameter Estimation

Due to the use of iterative parameter estimation in this project, its process would be specifically introduced in this section, which normally includes four steps: data collection, model selection, approach selection and hardware implementation [15].

3.3.1 Data collection

Data collection refers to system excitation, sampling rate selection, sampled data filtering and storage. Before collecting/sampling data, typically the input and output of identified systems, systems would be excited by injecting a set of frequency-rich perturbation signals into the control loop of the identified system. Perturbation signals commonly used include pink/blue noise signals [10, 191], chirp signals [161], multi-tone sinusoid signals [174], inverse repeat binary sequence (IRS) [158, 170] and pseudo-random-binary-sequence (PRBS). Due to its spectral resemblance to white noises, implemental ease, less computational cost and rich frequency, PRBS has been widely applied in [15, 26, 179, 184, 189] and will be used in this project either.

Sampling rate selection in SMPCs is often relevant to applications. the rate of SI, used for fault detection or lifetime prediction in which circuit component values (capacitors) should be estimated, is normally configured with high sampling frequency. For example, in [192-197], a sampling frequency, 10 times higher than switching frequency is selected for life-cycle monitoring, on-line fault detection, condition monitoring of power converters. This would bring a problem: Switching frequency is increasing recently as the power density requirement is becoming higher. If the switching frequency is as high as 100kHz in [193], it is difficult or

costly to set up the sampling frequency 10 times higher. In this case, [25] proposed a low sampling frequency fault detection method, BBO, which allows circuit components to be evaluated under a low sampling frequency. While for applications such as adaptive control, where only coefficients of mathematical models are required, low sampling frequency, typically equalling switching frequency, can be selected [22, 28, 179, 189]. However, if the sampling frequency is further lowered, effects of PRBS might be removed given that excitation signal frequencies are commonly equal to switching frequency as well.

The amount of data sampled and saved depends on the convergence time of SI approaches affected by intrinsic properties of the applied algorithms, the complexity of identified systems and noise effects, etc. For example, systems described with higher-order transfer functions will spend a longer time to be identified and consequently, a larger number of sampled signals is needed. Or RLS, a classical algorithm used for SI, will be more time-consuming than FAP is, proved in [179].

As SI is only based on the relation between the input and output signals, it is sensitive to noises that might be caused by poor connections, quantization in hardware implementation (ADC), the transformation between Laplace and digital domains, etc. Thus, filters should be carefully selected to remove unwanted noises. In [28, 179, 189], a 4-tap Moving Average Filter (MAF) is tested and applied while in [158], a low pass filter was mentioned to reduce measurement noises. Then filtered data should be stored in memories if it is necessary for some techniques, e.g. off-line or non-parametric, which may cause extra memory investments.

3.3.2 Model Selection

To realise typical grey box System Identification (SI), the mathematical models (transfer functions) of systems should be selected at first. Model selection of SI can be nonlinear, linear and hybrid. According to the small-signal modelling of buck converter described in Chapter 2, its model would be a linear one. The structure of a general-linear polynomial model is shown in Figure 3.2, in which A, B, C, D and F are polynomials that can be in both frequency domain or digital domain. As real-time parameter estimation is usually based on digital control techniques, discrete system transfer functions are required and expressed as (3-24).

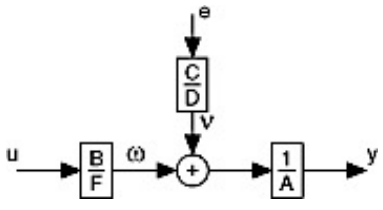


Figure 3.2 The Structure of a General-Linear Polynomial Model

$$A(z)y(k) = \frac{B(z)}{F(z)}u(k-n) + \frac{C(z)}{D(z)}e(k) \quad (3\text{-}24)$$

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_{k_a} z^{-k_a} \quad (3\text{-}25)$$

$$B(z) = b_0 + b_1 z^{-1} + \cdots + b_{k_b-1} z^{-(k_b-1)} \quad (3\text{-}26)$$

$$C(z) = 1 + c_1 z^{-1} + \cdots + c_{k_c} z^{-k_c} \quad (3\text{-}27)$$

$$D(z) = 1 + d_1 z^{-1} + \cdots + d_{k_d} z^{-k_d} \quad (3\text{-}28)$$

$$F(z) = 1 + f_1 z^{-1} + \cdots + f_{k_f} z^{-k_f} \quad (3\text{-}29)$$

where $u(k), e(k)$ and $y(k)$ respectively are system inputs, disturbances and system outputs in the discrete domain. n indicates the number of sampling delays. $A(z), B(z), C(z), D(z)$ and $F(z)$ are polynomials in the digital domain built to express ratio relation among inputs, outputs and disturbances. Then SI can be achieved by estimating coefficients of these polynomials [a, b, c, d and f, with subscripts (k), represented from (3-25) to (3-29)]. Based on this general linear model, many variations are derived. If $C(z), D(z)$ and $F(z)$ are 1, the model becomes Autoregressive with Extra Input (ARX). If $D(z)$ and $F(z)$ are 1, Autoregressive Moving Average with Extra Input (ARMAX) is acquired. Output-Error (OE) model is the version that $A(z), C(z)$ and $D(z)$ are 1 [69, 198], and Box-Jenkins (BJ) model is the one when $A(z)$ equals 1. These linear models, however, are not typically used for SI of SMPCs, particularly for buck/boost converters. Small signal average models of buck/boost converters are Single-Input Single-Output (SISO) but the afore-mentioned linear models all include disturbance terms, which are suitable for Multi-Input Single-Output (MISO). Thus, a model named Infinite Impulse Response (IIR) filter [or Autoregressive Moving Average (ARMA) model] is widely selected in SISO systems identification, for example in [15, 26, 97, 142, 179, 189], as well as applied in this project, of which structure is shown below:

$$A(z)y(k) = B(z)u(k-n) \quad (3\text{-}30)$$

Nonlinear models are usually combined with linear models to build hybrid models, which is also the classical model used in black box identification. For example, the Hammerstein model, which comprises a nonlinear static part and ARX part [158, 199], is applied to identify second-order systems such as boost converters [200, 201] and a fourth-order system in [202]. Specifically, only in the linear part, PRBS was injected while the nonlinear part is identified based on varied duty cycles. In [203], a DC-DC converter with remote control is expressed by a hybrid (Wiener-Hammerstein) model, which can take large-signal behaviour such as inrush current and starting up into account and assess power consumption and system stability.

3.3.3 Algorithm Selection

According to [15, 24-26, 189], algorithms investigated for being used in parameter estimation of SMPCs include FAP, DCD-RLS, KF, SALS, Biogeographical based optimization (BBO)-RLS, etc. They are all compared with the most classical one, RLS, and their performance can be evaluated in terms of convergence time, computational cost, estimation accuracy and robustness of disturbance rejection.

Table 3-1 Computational Complexities of the Mostly Used Algorithms in Parameter Estimation

Algorithms	**Computational Complexity**		
	+	×	/
SALS	$2M+1,$ (9)	$3M+2,$ (14)	
RLS	$4M^2,$ (64)	$6M^2+3M+1,$ (109)	1
FAP	$(1+N)M+N^3+2N^2+2N+1,$ (68)	$(1+N)M+N^3+2N^2+2N+2,$ (69)	
AP	$(N^2+2N)M+N^3+N^2-N,$ (93)	$(N^2+2N)M+N^3+N^2,$ (96)	
KF	$M^3+2M^2+2M,$ (104)	$M^3+2M^2+4M,$ (112)	1

^{a}M is the number of transfer function coefficients. ^{b}N is selected as 3 in this project, a positive integer that defines the number of historical steps considered in the current iteration.

Table 3-1 lists computational complexity per iteration of the commonly used algorithms from the easiest to the most complicated. SALS, slightly different from LMS, features the lowest calculation cost, strong noise-proof ability but significantly slow identification speed [184]. It was adopted in the high-frequency estimation of a single-rail-multi-phases buck converter in [26]. RLS, in virtue of low computational burden and qualified estimation accuracy, have been dominantly applied in parameter estimation of power electronics. Moreover, its convergence time can be adjusted through manipulating the forgetting factor (specifically described in Chapter 4. Nowadays, RLS with variable-forgetting factor is proposed to improve tracking of real-time parameter variations, although the algorithm is computationally heavier than the classical version which owns a fixed forgetting factor [31]. FAP, as a computational

complexity alleviation version of AP, was proposed for on-line SI of power converters [15]. Results show that FAP performs better than RLS in terms of convergence time and estimation accuracy. KF, despite the highest computational burden shown in Table 3-1, is experimentally demonstrated to be superior in dealing with abrupt load changes. Besides, KF may keep normal operation for a longer time without excitation signal injection, which could minimize the perturbation effects brought to voltage/current outputs of power electronics. Methods not listed in Table 3-1 are DCD-RLS and BBO-RLS. DCD-RLS in which there is not multiplication is proofed to be more computationally efficient than the classical RLS [28, 30]. BBO, the most computationally costly one among all these mentioned methods though, could achieve the highly precise real-time evaluation of circuit component values of a buck converter, even including Equivalent Series Resistance (ESR) of inductors and output capacitors as well, based on transfer function coefficients [25].

Given that individual configurations of these algorithms and the setup and environmental situations of their experimental validation, such as value selections of forgetting factor and step size, filters for pre-processing sampled signals, the magnitude of perturbation signals and noise severities, etc., are different, it is difficult to rank the performances of these algorithms based on their estimation speed presented in the above-mentioned papers. However, it is known that most recursive algorithms normally contain a numerical factor that would affect convergence time, such as constant step size, μ, in FAP, AP and KF and forgetting factor, λ, in RLS. Increasing step size or decreasing forgetting factor to a reasonable extent may speed the identification process up while a large step size factor, or a very small forgetting factor, may also result in instability or severe fluctuations in recursive curves.

Acceptable estimation accuracy, ±5% of real values suggested in [189], is achieved in all experimental validation sections in papers investigating the above-listed algorithms. Specifically, the identification results of applying KF are more accurate than those from RLS. There is no noticeable difference in converged values of AP, FAP and RLS.

In summary, in virtue of its less computational consumption, less convergence time and qualified estimation accuracy, RLS is still dominantly applied in parameter estimation of SMPCs. Other adaptive filters are more likely to be used to satisfy one particular requirement. For example, KF might be selected due to a strict demand for estimation accuracy while computational efforts could be compromised.

3.3.4 Hardware implementation of SI

Iterative algorithms, typically combined with closed-loop control, are commonly implemented for real-time SI through digital signal processors (DSPs) and Field Programmable

Gate Arrays (FPGAs). In [26], to realise SALS estimation, an Altera Cyclone IV E series FPGA (EP4CE115) was utilised and the switching frequency of the identified converter reached 1MHz. Others (FAP, DCD-RLS, KF and BBO-RLS) are performed on a Texas Instruments DSP (TMS320F28335) to estimate transfer function coefficients of a buck converter with 20kHz switching frequency [24, 28, 189]. In [189], an adaptive control loop was built based on real-time identification results through using KF, which indicates that the DSP (TMS320F28335) is computationally tolerable for the high complexity of KF, not to mention others suffering less calculation burden than KF. Meanwhile, with the continuous development of hardware equipment, their processing abilities have been upgraded and powerful enough to achieve a recursive algorithm-based online estimation of high order systems.

3.4 Prioritization of Multi-Rail Architecture

In multi-rail power conversion architectures, the number of rails can be varying on different occasions, and the importance of each rail may depend on the application, the magnitude of error signals, etc. of the rail. For instance, the rail providing regulated voltage to memory devices is often operationally superior to those for monitors, or rails bearing more frequent or periodic load changes should be paid more attention than those barely suffering abrupt system variations. Otherwise, when two rails are encountering disturbances at the same time, the one with a larger magnitude of its error signal, indicating a severer system change, should be coped with firstly. If these application-decided or error-signal-level-decided importance are considered in an adaptive control loop, some essential rails should be prioritized for real-time parameter estimation. As it typically takes a few milliseconds to get estimation results, when parameter estimation should take place in several rails at the same time, rails with low priority will wait for estimation-result update until ones with higher priority finish.

For real-time parameter estimation of a multi-rail power converter, there is a high likelihood that the number of rails sending SI requests would be above the maximum capability of simultaneous parameter estimation. As such, without prioritization, execution systems would encounter SI request jam due to not being able to decide which request should be responded to first. Accordingly, Maher Al-Greer proposed a predetermined prioritization way, followed by sampled data storage, in [204]. All the necessary data from rails sending SI requests will be sampled. After these requests being ranked based on their predefined priorities, rails with higher priorities will be identified first while sampled data from lower priority rails are saved for later use. This approach features ease of implementation and low ranking cost, it, however, requires users' rich experiences about multi-rail architectures they applied to conclude a well-predefined

ranking queue. Besides, the large amount of sampled data from rails whose SI requests are pending needs investments in storage equipment. Even worse, if system variations occur after sampled data storage in rails waiting for SI, the consequent estimation result would still be the one corresponding to the change which happened before data storage, instead of the newest one. As such, this paper would propose a hard-real-time SI allocation way allowing sampling only happens when SI is undergoing, i.e., there is not storage of sampled data.

While, task/request prioritization is not only discussed in the power electronic area, but has now been considered in various fields in which as long as multiple tasks should be dealt with under limited time/financial/equipment resources. These areas include fault discovery of source codes [205], multi-case testing, management of robotic production cells, response optimization of sensors [206], of network packets and of memory requests in chips [207], fault severity estimation [208] and data packet scheduling in wireless sensor networks [206], etc. To raise operation efficiency and avoid information jams, all these applications would contain an optimized decision-making workflow to schedule the sequence of requests/tasks. A typical example is a heterogeneous prioritization framework, applied in chip multiprocessor fields [207], which includes a novel way proposed to queue network packets and memory requests particularly for the increase in workload diversity. When tasks which need to be ranked are SI of multi-rail power converters, since that there can be, as demonstrated in Chapter 4, three rails identified at the same time, it is rails ranked ‘first’, ‘second’ and ‘third’ that would be selected for SI instead of only the first one. Consequently, the ranking system can be more tolerant as even if one rail is ranked higher than another, they may still be identified together. However, this would increase the complexity of prioritization: a comprehensively-designed and efficient workflow is necessary, such as the execution system proposed in [209], featured of high operation efficiency, and the framework in [207] considering various circumstances, etc.

There can be many ways of categorising scheduling approaches. Based on if tasks are interruptible, common scheduling approaches include pre-emptive (interruptible), Earliest Deadline First (EDF) (interruptible) and ‘First Come First Serve (FCFS)’ (uninterruptible) [210], etc. Apart from prioritizing simultaneous requests, execution system also need to deal with requests with high priorities sent when SI of rails with low priorities have already started. The solution can be to wait (FCFS) or to interrupt (Pre-emptive, EDF). A typical example of pre-emptive task assignment schemes, named ‘uC/OS-2’, is introduced in [211], under which new arrival tasks may interrupt ones being processed. Besides, a round-robin scheduling way to rank tasks with the same priorities is also mentioned in [211]. For comparison, the author of [210] specifically pointed out the features of the afore-mentioned three approaches: Non-pre-

emptive ways, such as First Come First Serve (FCFS), are more likely to result in later-requested-high-priority tasks pending for ages until earlier-requested-low-priority tasks finish. While using the pre-emptive method may lead to some tasks with lower priorities never can be dealt with during rush hours of request sending. As to EDF, which marks tasks, required by earlier completion deadlines, with higher priorities, it is, therefore, particularly in limited resources, possible that some tasks which share the same deadlines with others miss their opportunities of being coped with. Accordingly, to guarantee that more important rails can be identified earlier, this paper would apply the pre-emptive scheduling way. However, to avoid some low priority rails wait unreasonably long, there will be a window time set aside, only during which high-priority requests will be allowed to interrupt SI of low priority rails.

Based on if priorities of tasks are editable during system operation, scheduling approaches can be distinguished as ‘dynamic’ (editable) and ‘predetermined’ (invariable) [212]. For wise decision-making, a predetermined scheduling system, in which each task is pre-ranked by existing knowledge [212], requires a large amount of historical information including experts’ experienced opinions and reference database, etc. Papers referring to typical predetermined ranking ways include [208, 213], which describe fuzzy logic, and [214], which introduces a pre-decided priority rule, etc. Accordingly, predetermined scheduling approaches gain their popularity due to the unnecessity of complicated mathematic calculation [215], cost-saving and ease of hardware configuration. However, the unchangeable priorities of tasks during system operation makes these approaches difficult to be applied in circumstances that task importance varies [216]. Oppositely, dynamic prioritization methods are often introduced as ‘particularly suitable for real-time systems’ [206, 211] by their high flexibility [215]. For example, to minimize recording age (time elapsed since a request occurs until its corresponding task has finished), dynamic prioritization schemes are both applied on surveillance task assignment in [206] and integrated service coordination in [217]. The author of [217] pointed out, the applied scheduler features typical merits of most dynamic prioritization methods. Being FPGA-based and being iterative make the applied scheduler highly flexible, readily programmable and customizable. Editable scheduling ideas also can be seen in [218], in which newly arrived data segments (low priority) can be allocated into the high-priority queue for being handled earlier. Likewise, there is a well-defined revisable workflow being able to revise priorities of tasks in [209]. Based on updated priorities, tasks which have been, or asking for being, managed will be re-ranked for making decisions that if the undergoing process should be interrupted. In the case of SI of multi-rail power conversion architectures, as the magnitude of control error signals

of rails will be compared in real-time for defining priorities, the dynamic approach, therefore, is particularly suitable.

For an optimized scheme, there are various algorithms introduced to make ranking decisions. For example, in the cloud computing area, algorithms applied include FCFS [210] (a typical non-pre-emptive task allocation approach featured of being simple, error-free and able to automatically queueing tasks), ABC (Activity Based Costing, widely used in cloud computing), PSO (Particle Swarm Optimization, applied for the aim of processing-cost minimization), QoS-driven (able to take many elements, such as user privilege, expectation, task length and the pending time, etc. into queueing consideration) and VMT [219] (Virtual Machine Tree, particularly for rising task execution efficiency) (VMT), etc. Moreover, Genetic Algorithms were also mentioned in [220, 221] and Meta-Heuristic algorithms, traditionally applied for optimization, in [205]. These algorithms can provide reasonable results indeed, but costly. In the fields where tremendous cases need to be scheduled, cloud computing, for example, these algorithms could effectively and robustly prioritise tasks, while the coordination burden of SI requests in power electronics is much less than that in these areas, where applying complex algorithms, therefore, will cause unnecessary investments. The author of [212] suggests that there is always a trade-off between prioritization preciseness and the cost of carrying out algorithms, and therefore proposes a simplified ranking approach for sensor management. Respects that mainly be considered are separately weighted by respect factors based on which targets can be ranked. This predetermined and complexity-free quantization way may guarantee both reasonable and cost-saving prioritization process, and therefore, will be applied for SI request prioritization in multi-rail power converters. The four steps described in [222] (1. Numerical representation and normalization of respects, 2. Weights assignment for respects, 3. Score calculation for each request, 4. Ranking) are followed in this paper for ranking SI requests.

Various respects should be considered for task prioritization. In [223], only primary-user (PU) activities would priority rule design for cognitive radio (CR) transmission be based on. While, in [208] to rank probable faults in vital signs monitor, five relevant physiological variables, electrocardiogram, heart rate, respiratory rate, temperature and partial oxygen saturation, are quantitated and corporately considered. Likewise, [224] suggests three typical aspects (business, performance and technique) taken into consideration during optimization of prioritizing test cases. Conventionally, the design of the ranking rule should be based on difficulties in completing tasks. For example, in [210, 225], the tasks expected to spend less time to complete can get higher priorities. However, as task requests in this project are only SI, which is similar to the situation in [219, 226] (execution cost of tasks is uniformed), it is

unnecessary to compare the execution efforts of different tasks. Moreover, many ranking schemes are mainly based on the application importance of each task, such as those in [205, 227-230]. Overall, the commonly and necessarily considered respects are well listed in [231]: the execution efforts of tasks (spending time, overhead and complexity, etc.), fault-occurrence odds, historical priority records, time spent on task ranking, and urgency levels decided by application and fault properties, etc. The most usual and simple way of corporately taking these respects into prioritization consideration is to quantitate them by different weights, such as Simple Additive Method (SAM) and Simple Weighted Product (SWP) [222, 231-233], etc. For example, in [234], three respects, source code information, application coverage and execution time, are quantitated with equally assigned weight, which is, however, doubted by the author of [205], who proposes to automatically allocate more weights to fault-prone software artefacts and to those important in applications. In this project, respects reflecting urgency levels of SI requests include the severity of errors that rails suffer, application importance and system-variation odds, etc.

Except avoiding task request jams, being reasonable and flexible, as references [205, 206, 209, 235] suggest, other common objectives of ranking scheme design include cost-saving (cost constraint), execution time minimization (time constraint) and high efficiency, etc. For example, the ranking approach in [223] is proposed under route overhead limits, and in [225] FCFS and EDF are applied under time constraints. In [224] about fault discovery in source code, two pursuits, cost reduction and earlier completion, are separately called the business and technical criteria. As of [217], where a mix-criticality scheduling system is proposed with objectives including the shortened service time, high quality, flexible real-time control system, minimized scheduling computational cost, it is known that sometimes designing scheduling rule does not only follow a single objective but multiples [231]. As such, several multi-objective decision-making models are proposed such as MOORA, AHP, TOPSIS, VIKOR, ELECTRE, PROMETHEE [205], etc. The corresponding performance comparisons among these models indicate that MOORA might be the best when computational time, simplicity and mathematical calculation complexity are the main objectives of decision making [236]. Another method of corporately following several criteria is to simply produce multiple ranking queues, then mixed [210]. As described in [237] and [238], two ranking queues are generated by applying EDF and Adaptive Double Ring Scheduling (ADRS), and in [210] three queues for both execution efficiency and flexibility in real time. Based on these previous studies, the objectives of the proposed scheduling approach in this paper can be to minimise the average pending time of SI requests, simplified and cost-saving ranking scheme.

Overall, the current ranking scheme for SI requests of multi-rail power converters is not able to decide priorities based on the severity of system variations in real-time, according to the prioritization approaches applied in various areas and their characteristics, a new task allocation way, derived from [209] and [219], will be proposed in this paper, required to: 1. dynamically update request priorities in hard-real-time to guarantee that SI can produce the newest estimation results, 2. be able to provide SI opportunity for both high and low priority rails, 3. comprehensively consider various respects to define priorities, 4. be cost-saving and efficient.

3.5 Chapter Summary

The chapter shows an overview of system identification techniques applied in SMPCs, including the introductions of non-parametric estimation and parametric estimation methods. Iterative parameter estimation, mainly applied in this project, is reviewed in a detailed way. The process of achieving parameter estimation is divided into four steps (data sampling and collection, model selection, algorithm selection and hardware implementation), specifically introduced in **3.3**. Particularly, most algorithms commonly used in parameter estimation of SMPCs are listed and compared. Besides, to build a prioritization workflow for system identification of multi-rail power conversion architectures, various task ranking/sequencing workflow (or framework) proposed in other application areas are investigated. The optimization approaches of the parameter estimation process proposed in Chapter 4 and 5, and the prioritization workflow designed in Chapter 6 are both based on the background knowledge introduced in this chapter.

Chapter 4 Two Proposed Computational Burden Alleviation Approaches (Iteration Decimation and CMA/CT Reusing)

4.1 Introduction

According to pieces of literature, there have been many recursive algorithms, including RLS, AP (FAP) and KF, etc., performing well in real-time parameter estimation for single power converters [15, 24, 26, 29, 189]. However, the algorithm processing speed of the applied DSPs still limits the switching frequency of power electronics. With the development of the signal processing units, these algorithms can be easily implemented in DSPs with the minimized operation time. In [189], they will respectively take 37μs and 34μs to complete one iteration cycle when applying KF and RLS on a DSP (TMS320F28335). As such, the sampling frequency, equaling the switching frequency in that work, had to be as low as 20kHz to guarantee the sampling intervals being longer than 37μs. If a higher switching frequency is required, 50kHz for example, more advanced processors which could complete one iteration cycle in less than 20μs might be necessary. Otherwise, when the adaptive filters are applied in multi-rail architectures with a centralized single controller, the computational burden will become significantly heavy, increasingly proportionately with the addition of rails. For example, if the available computation time is 50μs, the employed processor should finish 64 additions, 109 multiplications, and 1 division in 50μs for single-rail parameter estimation by using RLS (see Table 4-1). If three rails are simultaneously identified, the computational burden in the 50μs will be increased to 192 additions, 327 multiplications, and 3 divisions. The significant increase in the computational burden could cause the need for advanced processors more computationally capable particularly, resulting in extra investments. As such, this paper considers two approaches to reduce the computational complexity of multi-rail converters and better facilitate centralized single processor control. These experimentally validated approaches are 1. Iteration frequency reduction. 2. Frequency reduction of updating Covariance Matrix Approximation (CMA), or Correction Term (CT) by re-using CMA/CT. RLS, AP and KF algorithms are employed to validate the proposed approaches, which can be more widely applicable to other recursive estimation algorithms though, in both simulation and practical.

4.2 Parameter Estimation of SMPCs

4.2.1 Introduction of Parameter Estimation of a buck converter

There are various modelling ways reviewed in Chapter 2, and small signal modelling is selected in this project to mathematically express the power converter architectures. Equation (4-1) is the transfer function in z-domain of a single buck converter.

$$\frac{V(z)}{D(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{4-1}$$

Specifically, the relation between the sampled output voltage at ith iteration and the previously sampled signals, including the sampled output voltage and the sampled duty cycle, at $(i-1)th$ and $(i-2)th$ iteration can be shown as follows:

$$V(i) = -a_1 V(i-1) - a_2 V(i-2) + b_1 D(i-1) + b_2 D(i-2), \tag{4-2}$$

where V is the voltage output, D is duty cycle, a_1, a_2, b_1, and b_2 are the four parameters which need to be identified after every sampling action.

Figure 4.1 shows the proposed real time parameter estimation process of a multi-rail architecture containing three rails. By superimposing a small frequency rich perturbation signal, here a Pseudo Random Binary Sequence (PRBS) [160], the resultant duty cycles, $d\hat{}(n)$, will momentarily excite the output voltage of the corresponding power rail. Then, the algorithm will process the sampled duty cycle and the voltage signals to estimate the parameters of the transfer function in (4-2) (a_1, a_2, b_1, and b_2).

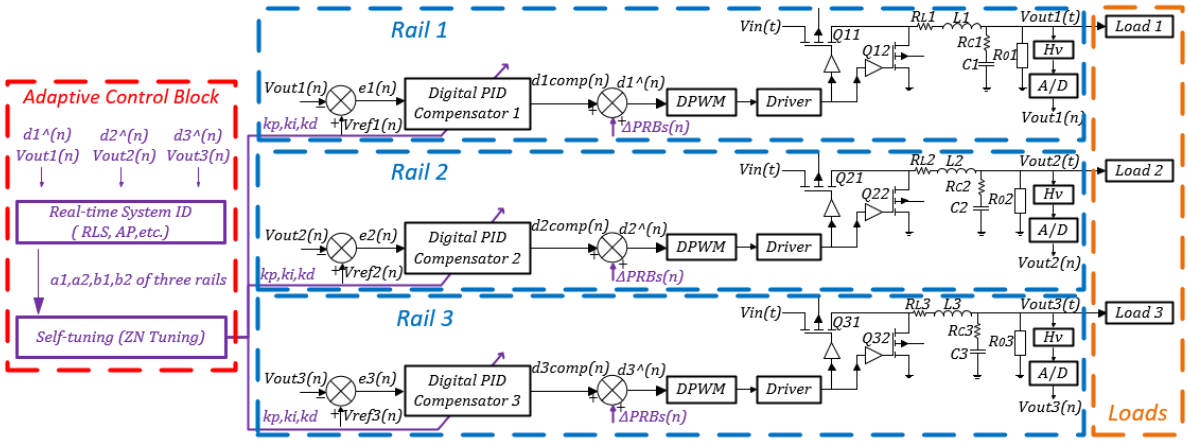


Figure 4.1 A Three-Rail Power Converter Architecture (Blue, Right Side) and the Real-Time System Identification Process (Red, Left Side) [28]

4.2.2 Adaptive Filters Employed for Parameter Estimation

As Section 3.2.2 shows, the steepest-descent methods iterate the optimal guess vector by minimizing the cost function, (3-1). In this project, the best fit of the parameter vector, $\omega \triangleq [a_1\ a_2\ b_1\ b_2]$, should be found. Most adaptive filters used for parameter estimation are based on Regularized Newton's recursions, such as Least Mean Squares (LMS), RLS and Affine Projection (AP), etc., which update New Guess (NG) through adding Correction Term (CT) calculated in the current iteration cycle to the NG gotten from the last cycle. CT contains a direction part [Gradient Vector (GV)] and a magnitude part [Covariance Matrix Approximation

(CMA)/Kalman Gain (KG)]. Therefore, updating an NG typically includes 5 update steps: 1. Regressor, 2. the magnitude of CT, 3. the direction of CT, 4. Correction Term and 5. New Guess.

A. Recursive Least-Square (RLS)

Figure 4.2 indicates the process of one iteration cycle of RLS [239]. (The detailed derivation processes of RLS and AP are all collected in Appendix A) In Figure 4.2, the true values of the parameter vector, $[a_1\ a_2\ b_1\ b_2]$, multiplying the regressor, u_i, may result in the output voltage signal, $y(i)$, while the iterated New Guess would produce the predicted output, $\hat{y}(i)$. The difference between the sampled output and the prediction in the last iteration cycle, $y(i) - \hat{y}(i)$, is used for proceeding the next cycle.

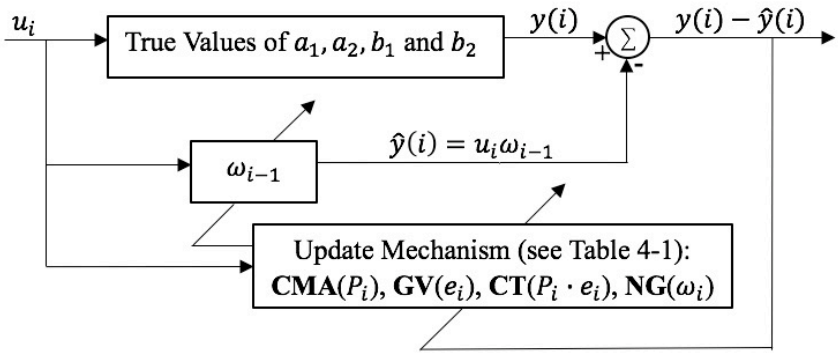


Figure 4.2 Block Diagram of the RLS Algorithm

Table 4-1 The Update Sequence of an Iteration Cycle of RLS

Step	**Updates**	**RLS Formula**
Initialization		$u_0 = 0, y(0) = 0, 0 < \lambda \leq 1, \omega_{-1} = 0, P_0 = g\boldsymbol{I}$, where $\boldsymbol{I}$ is an $M \times M$ identity matrix, g is a large number. i is the current iteration instant and n the current sampling instant, same in the following contents.
1	**Regressor**	$u_i \triangleq [-V(n-1)\ -V(n-2)\ D(n-1)\ D(n-2)]$ $y(i) \triangleq V(n)$
2	**CMA**	$P_i = \lambda^{-1}\left[P_{i-1} - \frac{\lambda^{-1}P_{i-1}u_i^* u_i P_{i-1}}{1 + \lambda^{-1}u_i P_{i-1}u_i^*}\right]$
3	**GV**	$e_i = u_i^*[y(i) - u_i\omega_{i-1}]$
4	**CT**	$P_i \cdot e_i$
5	**NG**	$\omega_i = \omega_{i-1} + P_i e_i$

The mechanism of the four update steps – the magnitude of CT, the direction of CT, CT and NG – are respectively presented as Step 2, 3, 4 and 5 in Table 4-1:

1. the magnitude of CT (CMA in RLS, or P_i, in Table 4-1): In RLS, P_i, a matrix with the size of M by M (M, the amount of the estimated parameters, is 4 here), can be acquired when the last CMA (P_{i-1}), Forgetting Factor (λ) and regressor are known. λ is a convergence-time-related constant investigated in Section 4.3.1.

2. the direction of CT (Gradient Vector in RLS, or e_i, in Table 4-1): In RLS, the Gradient Vector is the multiplication of the regressor, u_i, and the difference, $y(i) - \hat{y}(i)$.

3. Correction Term: the multiplication of the direction part of CT and the magnitude part.

4. New Guess: the sum of the last guess (ω_{i-1}) and the CT.

The computational complexity of the aforementioned update steps is presented in Table 4-2, which indicates CMA iteration spends more than half of the total computation costs.

Table 4-2 The Computational Cost of RLS per Iteration

Step	**Updates**	**Computational Complexity (RLS)**		
		+	×	/
1	**Regressor**			
2	**CMA**	$3M^2 - M, (44)$	$5M^2 + M + 1, (85)$	1
3	**GV**	$^{a}M, (4)$	$2M, (8)$	
4	**CT**	$M^2 - M, (12)$	$M^2, (16)$	
5	**NG**	$M, (4)$		
	In Total	$4M^2, (64)$	$6M^2 + 3M + 1, (109)$	1

B. Affine Projection (AP)

Same with RLS, AP also includes five update steps in an iteration cycle (see Table 4-3), the computational costs of each step are listed in Table 4-4 [239]. As Figure 4.3 indicates, other than including the current instantaneous values – u_i [or $y(i)$] – shown in RLS, the regressor of AP – U_i (or Y_i) – also comprises several previous ones – u_{i-1}, ⋯, u_{i-N+1} [or $y(i-1)$, ⋯, $y(i-N+1)$]. (N is the amount of the 'u' used, or the row size of U_i. The more the historical data are used which means the larger N is, the more accurate the consequent CMA is while the heavier the computational burdens are.) As Step 2 of Table 4-3 shows, the magnitude of CT (Δ_i) in AP algorithm is acquired when the regressor, and the preconfigured regularization sequence (ϵ) and Step Size (μ) are known. Gradient Vector ($grad_i$ in Table 4-3, the direction part of CT) of AP is the difference matrix between the actual output matrix (Y_i) and the predicted ones ($U_i\omega_{i-1}$). The steps of updating CT and NG are the same with those in RLS: by

multiplying its direction and magnitude parts, CT is acquired and then added to the last guess (ω_{i-1}) for producing NG (ω_i).

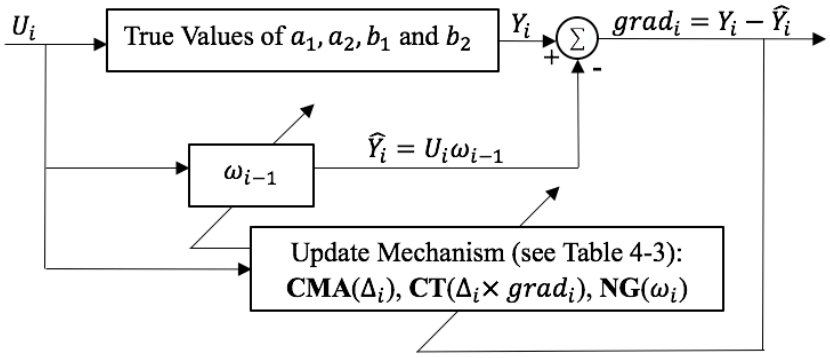


Figure 4.3 Block Diagram of the AP Algorithm

Table 4-3 The Update Sequence of an Iteration Cycle of AP

Step	Updates	AP Formula
Initialization		$u_0 = 0, y(0) = 0, U_0 = 0, Y_0 = 0, \omega_{-1} = 0, \mu$ is a constant step-size sequence, ϵ is a constant regularization sequence.
1	**Regressor**	$u_i \triangleq [-V(i-1)\ -V(i-2)\ D(i-1)\ D(i-2)]^T$ $y(i) \triangleq V(i)$ $U_i \triangleq [u_i\ u_{i-1} \cdots u_{i-N+1}]^T$ $Y_i \triangleq [y(i)\, y(i-1) \cdots y(i-N+1)]^T$
2	**CMA**	$\Delta_i = \mu U_i^* [\epsilon I + U_i U_i^*]^{-1}$
3	**GV**	$grad_i = [Y_i - U_i\omega_{i-1}]$
4	**CT**	$\Delta_i \times grad_i$
5	**NG**	$\omega_i = \omega_{i-1} + \Delta_i\, grad_i$

Table 4-4 The Computational Cost of AP per Iteration

Step	Updates	Computational Complexity (RLS)	
		+	×
1	**Regressor**		
2	**CMA**	*(N^2+N-1) M+N^3, (71)*	*(N^2+N) M +N^3, (75)*
3	**GV**	*NM, (12)*	*NM, (12)*
4	**CT**	*N^2-N, (6)*	*N^2, (9)*
5	**NG**	*M, (4)*	
	In Total	*(N^2+2N) M+N^3+N^2-N, (93)*	*(N^2+2N) M+N^3+N^2, (96)*

C. Kalman Filter (KF)

Different from RLS and AP, Kalman Filer is not derived from Regularized Newton's recursions, which therefore includes an additional update step: Kalman Gain (KG) update (Step 3 in Table 4-5). Nevertheless, the recursive cycle of KF presented in Figure 4.4, based on the difference between the practical output and the prediction, $y(i) - \hat{y}(i)$, is the same with that of RLS/AP. The six update steps for acquiring NG and the computational costs of each step are respectively presented in Table 4-5 and Table 4-6. The specific deriving process of KF is presented in Chapter 3.

The regressor of KF, u_i, is the same of that of RLS as shown in Table 4-5 and Figure 4.4. The magnitude of CT in Kalman Filter is KG (G_i in Table 4-5) before getting which the CMA (or P_i) should be acquired according to the regressor, the last KG (G_{i-1}) and $\hat{E}_i$ of which the definition equation is presented in Table 4-5. G_i, then, could be calculated based on the pre-configured r (see Table 4-5), the regressor and P_i. Gradient Vector (the direction of CT, or e_i in Table 4-5) in KF is the difference between the sampled output and the prediction, $y(i) - \hat{y}(i)$. The way of updating CT and NG in KF is the same with that of RLS or AP.

Table 4-5 The Update Sequence of an Iteration Cycle of KF

Step	**Updates**	**KF Formula**
Initialization		$u_0 = 0, y(0) = 0, \omega_{-1} = 0, P_0 = g\boldsymbol{I}$, where $\boldsymbol{I}$ is an $M \times M$ identity matrix, g is a large number. r is a constant observation noise variance. $\hat{E}_i \triangleq diag\big[[\hat{a}_1(i-1)]^2.[\hat{a}_2(i-1)]^2.[\hat{b}_1(i-1)]^2.\ [\hat{b}_2(i-1)]^2\big]$.
1	**Regressor**	$u_i \triangleq [-V(n-1)\ -V(n-2)\ D(n-1)\ D(n-2)]$ $y(i) \triangleq V(n)$
2	**CMA**	$P_i = P_{i-1}(\boldsymbol{I} - G_{i-1}u_i) + \hat{E}_i$
3	**KG**	$G_i = P_i u_i^*[u_i P_i u_i^* + \mathrm{r}]^{-1}$
4	**GV**	$e_i = [y(i) - u_i\omega_{i-1}]$
5	**CT**	$G_i \cdot e_i$
6	**NG**	$\omega_i = \omega_{i-1} + G_i e_i$

Table 4-6 The Computational Cost of KF per Iteration

Step	Updates	Computational Complexity (RLS)		
		+	×	/
1	**Regressor**			
2	**CMA**	$M^3 + M^2, (80)$	$M^3 + M^2, (80)$	
3	**KG**	$M^2, (16)$	$M^2 + 2M, (24)$	1
4	**GV**	$M, (4)$	$M, (4)$	
5	**CT**		$M, (4)$	
6	**NG**	$M, (4)$		
	In Total	$M^3 + 2M^2 + 2M,\ (104)$	$M^2 + 2M^2 + 4M, (112)$	1

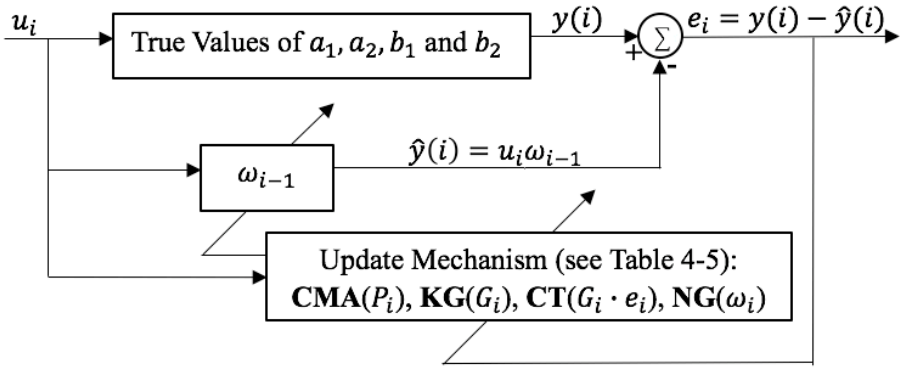


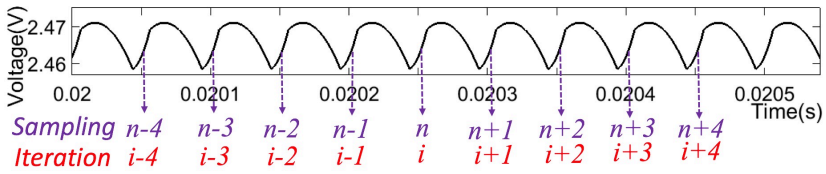
Figure 4.4 Block Diagram of the KF Algorithm

The three algorithms have been introduced in real-time parameter estimation of power electronics by virtue of different merits. RLS, as the most classic one, has been widely used and compared with its variants and other candidate algorithms [25, 26]. The KF is demonstrated to have advantages in estimation accuracy and dealing with abrupt load changes [189].

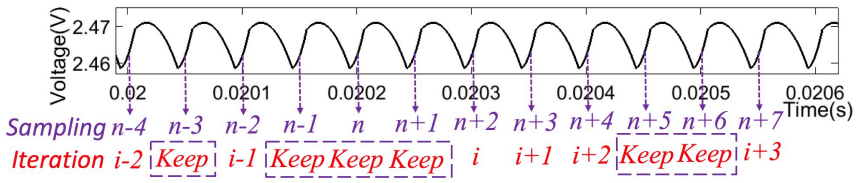
4.3 Two Proposed Approach: Variable Iteration Frequency

4.3.1 Introduction of the Iteration Decimation Approach (The First Approach)

In the conventional iteration process (Figure 4.5(a)), the iteration frequency is chosen to equal the sampling frequency, which means iterations, $i, i \pm 1, i \pm 2$ ...., act after every sampling event (the dashed purple lines in Figure 4.5), $n, n \pm 1, n \pm 2$ ....



(a)



(b)

Figure 4.5 The Comparison of Iteration Frequency between the Conventional Process (a) and Proposal (b)

In this work, the computational cost is reduced by lowering the iteration frequency. The recursive algorithms no longer update NG after every sampling event, instead, there are several intermediary sampling time intervals between iterations (see Figure 4.5(b)). During these intermediary intervals, Step 2-5 in Table 4-1 and Table 4-3 (or Step 2-6 in Table 4-5) are eliminated/tailored after regressor updates. CMA, KG, GV, CT and NG will simply hold the most recent values until the next iteration phase completes. The regressor vectors, Step 1 shown in the three tables, however, need to update at every sampling instant to ensure the same parameter values are identified from the original identification process. For instance, if the sampling frequency is 40kHz, the iteration frequency can be decimated, but the regressor update frequency must be kept at 40kHz. A reduced regressor update frequency will result in estimated model parameters that do not match the 40kHz form (discrete models are sampling-frequency-dependent).

The option to flexibly change the iteration frequency is beneficial when attempting to reduce computational burden and filter the disturbances caused by abrupt system variations in the sampled signals. Therefore, a variable K is included in the parameter estimation process to represent the number of sampling events which take place between those samples used to derive the next parameter estimation update. Thus, the iteration frequency can be defined as:

$$SF = IF \times K \tag{4-3}$$

where, SF represents the Sampling Frequency, IF represents the Iteration Frequency.

In simple terms, K can be chosen as a constant, serving as a decimating factor. However, to optimize the algorithm further, it is possible to dynamically vary K based on the magnitude of the control error signal. In doing so, it is possible to prioritize the need to update the parameter estimation in the event of substantial system disturbances which are likely to perturb the controller error. While, there is no need to particularly set up which sampled data will be skipped or kept, as long as the iteration frequency is reduced and iteration events of rails are stagger with each other instead of happening together. The sampling frequency depends on the

computational complexity of the parameter-estimation-used algorithm and the processing capability of the employed processor. Once the algorithm and the processor were selected, the time spent on the processor finishing one iteration cycle can be estimated. Then there would be an upper limit of the iteration frequency to guarantee that the whole iteration cycle can complete in an iteration interval. In the conventional way in which the iteration frequency equals the sampling frequency, the upper limits of the two frequencies are the same. However, if the proposed iteration decimation approach was applied, the sampling frequency can be defined as long as the decimation factor, K in (4-3), is known. Meanwhile, the limit on the sampling frequency will lead to the same limit on the switching frequency of SMPCs, as the two frequencies are typically configured to be equal.

Reducing iteration frequency may achieve computational burden alleviation in every sampling interval, however, may also prolong the time spent on parameter estimation. To solve this, the forgetting factor (λ) in RLS (see Table 4-1) is investigated. As the factor also affects the identification speed, carefully tuning λ may shorten the prolonged estimation time caused by the reduced iteration frequency. Furthermore, to demonstrate the effects brought by the proposed approach on estimation performance, three commonly-used indices (convergence time, estimation error and variance of estimated results) expressing estimation performance are introduced (see Figure 4.6). In Figure 4.6, the estimation process is divided into two stages. In Stage 1, the guesses of the estimated parameters are being iterated to acquire the optimal values, which therefore cannot be used for adaptive controller tuning (recursive curves indicating estimation results have not converged to the true values). In Stage 2, recursive curves have converged to the true values (the optimal guesses have been found) and kept the values for 0.01s to complete the controller parameter update. Based on Figure 4.6, the three indices describing estimation performance are:

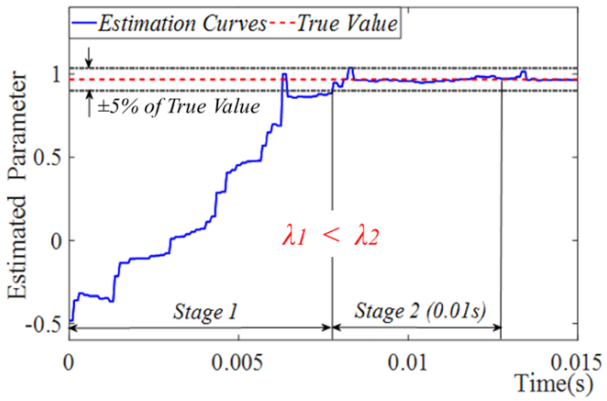


Figure 4.6 The Two Stages of Parameter Estimation Process

1. *Convergence Time:* the duration of Stage 1 in Figure 4.6, starting at the beginning of parameter estimation and ending at the time when the recursive curves have entered and remained within their error bands $\pm 5\%$ of real values [189].

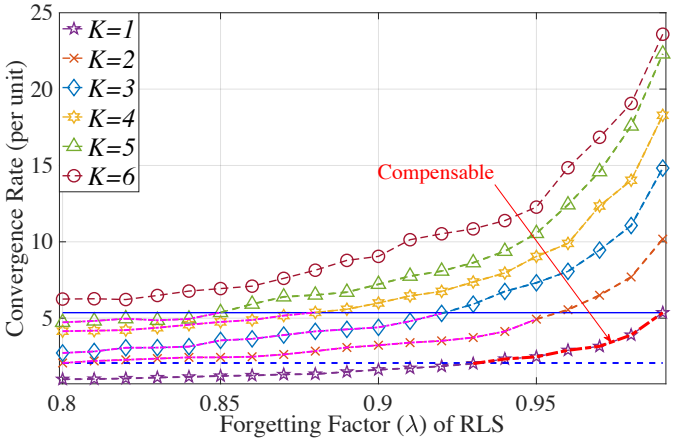
2. *Estimation Error:* Stage 2 (see Figure 4.6) begins with the end of Stage 1 and ends with 0.01s after. The average value of the curves in Stage 2 is usually the identified model parameter taken into adaptive control account. The difference between this average value and True Value is known as the estimation error which implies estimation accuracy.

3. *Variance of Estimated Results:* the variance of Stage 2 (Figure 4.6). Variance is another way to reflect the estimation accuracy in case the average is affected by extreme values.

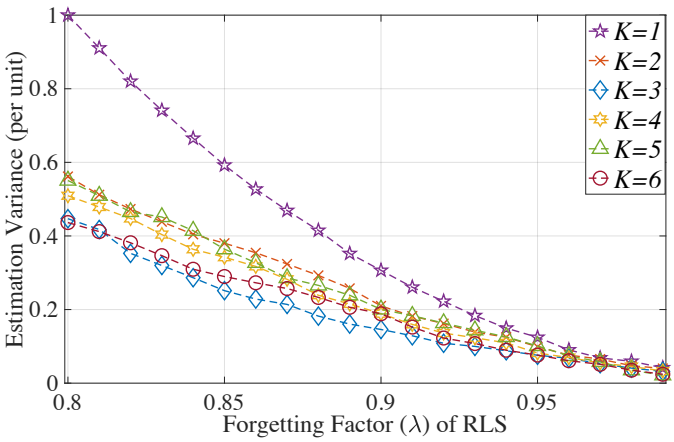
To shorten the prolonged convergence time caused by lowering iteration frequency, by manipulating Forgetting Factor (λ) of RLS, the relation among K, λ, and estimation performances are investigated (see Figure 4.7). Here, convergence time is shown in Figure 4.7(a), variance in Figure 4.7(b), and estimation error in Figure 4.7(c). The Y-axis values in Figure 4.7 are shown on a self-defined "per unit" scale. where the convergence time, estimation error and variance at $\lambda = 0.8$ and $K = 1$ is considered 'unity' or '1'. As such, all other points on the graphs are reference values with respect to the per-unit case. For example, when $\lambda = 0.82$ and $K = 3$ the convergence time is 3 times longer than that when $\lambda = 0.8$ and $K = 1$. From Figure 4.7(a), it can be observed that:

1. Increasing K leads to a longer convergence time of RLS, about inversely proportional to the iteration frequency. if the iteration frequency is reduced to a third, its corresponding convergence time will be about three times longer.

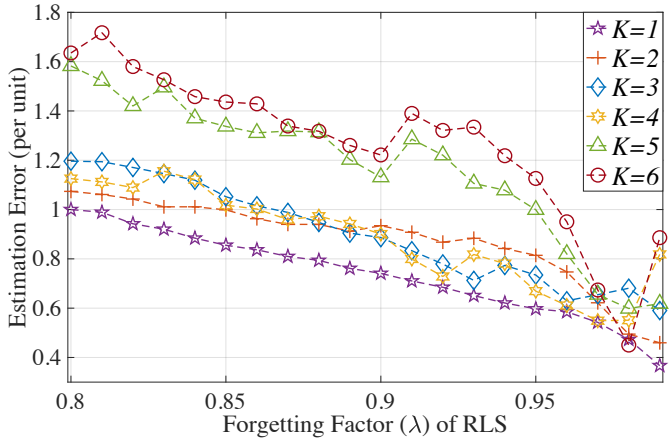
2. Decreasing λ will reduce the convergence time, but the larger K would make it more difficult to reach an acceptable convergence time. The solid blue line in Figure 4.7(a) shows the convergence time when $\lambda = 0.99$ and $K = 1$ and the dashed blue line is convergence time when $\lambda = 0.8$ and $K = 2$, so within the two blue lines, when a larger K is adopted for saving computational complexity, a small λ (Curves depicted in pink) can then be selected to guarantee the convergence time of the larger K is always the same with that of its $K = 1$ counterpart (Curves depicted in red).



(a)



(b)



(c)

Figure 4.7 Performance Comparison of Parameter Estimation under Different Iteration Frequency and Forgetting Factor for RLS

According to Figure 4.7(b), larger λ and K are preferred as they produce less variance. converged curves have less fluctuations. In Figure 4.7(c), the estimation error when $K = 1$ is less than that of others, and the errors when $K = 5$ or 6 are significantly higher. 5 or higher

values may not be suitable selections of K, but even if K equals 2 or 3, the lower estimation accuracy cannot be ameliorated unless a larger λ is applied which negatively influences convergence time. Therefore, adaptive λ, which is adjustable in different estimation stages, is proposed here. At stage I (Figure 4.7), λ is selected as a smaller value for fast estimation speed. In Stage II, starting at the end of Stage I to the end of parameter estimation process, λ is increased to ensure stability in curves. Theoretically, this approach can be applied in most stochastic-gradient based algorithms. Typically, they always include a factor directly affecting transient behaviours, e.g. Step Size (μ) in AP and Least-mean Square (LMS), Observation Noise Variance (r) in KF, Forgetting Factor (λ) in RLS, etc.

With the same groups of the sampled data, there is no obvious relation between the estimation performance and the two convergence-time-related factors in both KF and AP acquired. However, it has been known that a smaller μ would prolong the convergence time of AP with a higher estimation accuracy. Conversely, a smaller r may shorten the convergence time of KF and result in a larger estimation error. As such, through carefully tuning the values of r and μ in different estimation stages, the proposed iteration decimation approach can also be employed on KF and AP without noticeable compromises of estimation performances. In Figure 4.6 the factors would be '$\mu_1 > \mu_2$' and '$r_1 < r_2$' when AP and KF are separately applied.

4.3.2 Load Change Rejection by Filtering Transient Responses

Figure 4.8 demonstrates the impact of temporarily reducing iteration frequency to cope with abrupt disturbances (resistance load changes here). RLS is exampled here. Because the load change occurs in Stage 2, λ is normally a large value such as 0.98. With a reduced iteration frequency, the proposed approach would spend longer time to update the estimation results. Reducing λ for shortening convergence time, however, may not be suitable for disruption rejection, as it will assign more weights to recently updated regressors. Therefore, K is temporarily increased to 10 (decreasing the iteration frequency) to dispose disturbances in sampled signals. Figure 4.8 demonstrates the iteration arrangement when an abrupt disturbance occurs in one rail of a three-rail architecture, in which no iteration will be allocated when sampled signals are transient responses dealing with system variations. Likewise, this disturbance disposal approach can also be applied on other adaptive filters, e.g. AP and KF, etc. The corresponding performance comparisons are both shown in **Section 4.6** (simulation results) and **Chapter 5** (experimental results), which indicate that the recursive curve of $K = 10$ (the sampled transient responses are filtered) enters the accuracy tolerance range, ±5% of real values, faster than the other ways do and features the most moderate transient behaviour.

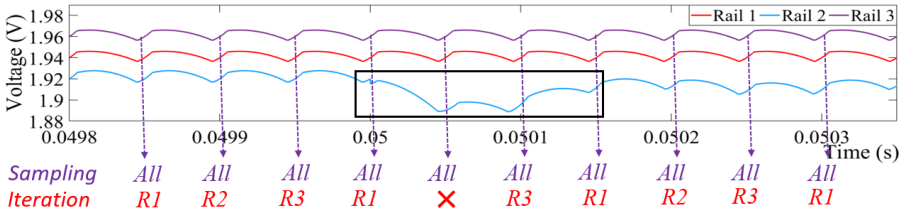
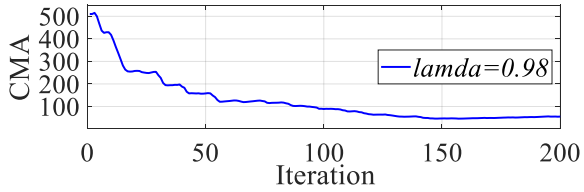


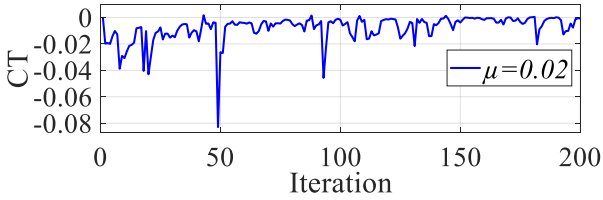
Figure 4.8 The Disturbance Disposal with Sampled Signals

4.3.3 Introduction of the CMA/CT Reusing Approach (The Second Approach)

According to Tables 4-1, 4-3 and 4-5, the computational costs of CMA updates in both RLS or KF, and of CT updates in AP, are higher than the sum of the costs spent on other steps. Therefore, reducing the CMA/CT update frequency is one clear way to reduce computational burdens. Between updates, the same CMA/CT value might be re-used. As such, investigations of the magnitudes of CMA/CT were taken in this project.



(a)



(b)

Figure 4.9 Reduction of CMA of RLS and CT of AP in the First 200 Iteration Cycles

Figure 4.9 presents the magnitude changes of CMA in RLS and CT in AP in the first 200 iterations of parameter estimation, acquired by implementing RLS and AP on practical data for real-time system identification. In Figure 4.9 CMA reduced by 450 and CT reduced by 0.08. Then, the difference between two consecutive CMAs in RLS [see Figure 4.11(a)] and CTs in AP [see Figure 4.11(b)] are investigated. Figure 4.11(a) indicates the values of $CMA2 - CMA1, CMA3 - CMA2, CMA4 - CMA3$,…, shown in Figure 4.10, and Figure 4.11(b) the values of $CT2 - CT1, CT3 - CT2, CT4 - CT3$, ….. Given that the values of forgetting factor, λ, of RLS and Step Size, μ, of AP would affect the magnitude of CMA/CT, different values of λ or μ are also presented in Figure 4.11.

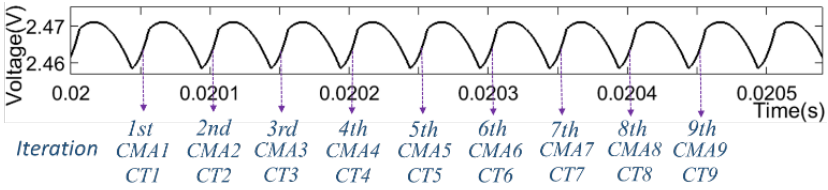
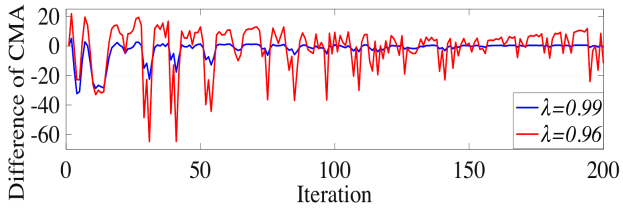
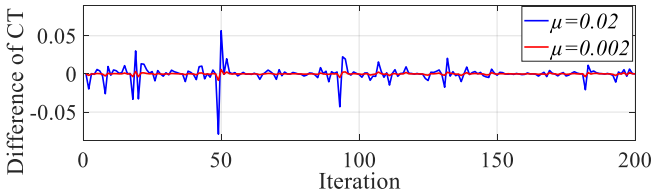


Figure 4.10 The Relation between CMAs and Iterations in the Conventional Parameter Estimation



(a)



(b)

Figure 4.11 The Magnitude of CMA of RLS (a) and CT of AP (b) when $Q = 2$

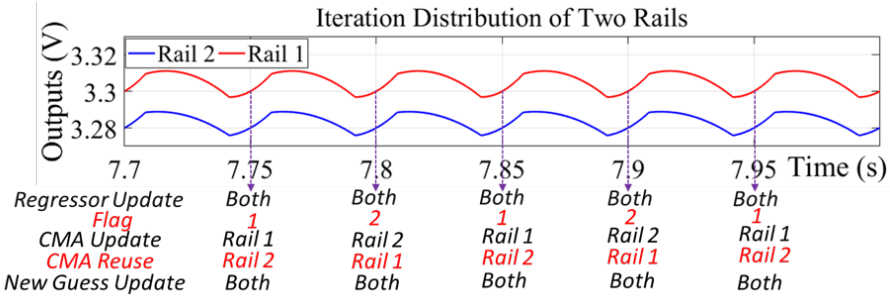
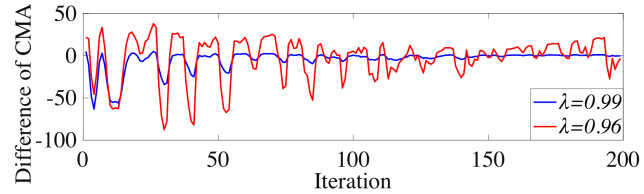


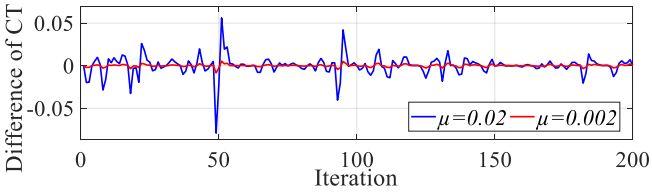
Figure 4.12 The Iteration Distribution of Reusing CMA in a Two-Rail Architecture

As of Figure 4.11, a smaller λ would result in a larger magnitude of the CMA Difference, and a smaller μ a smaller magnitude of the CT Difference. Besides, the maximum fluctuation amplitude, about 80, in Figure 4.11(a), is merely about 17% of CMA magnitude reduction (450 in Figure 4.9(a)), which suggests the two consecutive CMAs in RLS are almost the same. Similarly, the insignificant CT Difference magnitude in Figure 4.11(b) indicates the CTs of AP between two consecutive iterations are almost the same. Therefore, it is assumed that the CMA calculated in the last iteration is a reasonable substitution for the current iteration cycle for RLS, and CT is suitable for AP.

As such, the proposal of CMA/CT substitution allows a reduction of computational burden per iteration if two rails are identified simultaneously (see Figure 4.12). A variable Q is introduced to indicate how many times a CMA/CT will be used in the iteration process. In Figure 4.12, as the CMA/CT calculated from the last iteration is only once reused (into the current one), Q equals 2. 'Flag' indicates which converter/rail should take CMA/CT update after a sampling event. Here, 'Whole iteration' is stipulated to stand for conducting all steps of an iteration cycle and 'Partial Iteration' doing all steps apart from Step 2 (CMA/CT updates) in Table 4-1 and 4-3. As such, after every sampling event, one rail will conduct 'Whole Iteration' and the other one takes 'Partial Iteration (see Figure 4.12). Then the computational consumption is reduced from how much two times a 'Whole Iteration' costs to how much a 'Whole Iteration' and one 'Partial Iteration' cost, which saves more than half of the computational efforts of the conventional estimation way (see Table 4-7).



(a)



(b)

Figure 4.13 The Magnitude of CMA of RLS (a) and CT of AP (b) when $Q = 3$

To alleviate computational burdens further, the CMA/CT calculated at the last iteration is reused twice for both the current and the next iteration cycles. Figure 4.13 shows the difference between the next CMA/CT and the last one respectively, with λ = 0.96 and 0.99 or μ = 0.002 and 0.02. Figure 4.13(a) indicates the values of $CMA3 - CMA1, CMA4 - CMA2, CMA5 - CMA3$,…, shown in Figure 4.10, and Figure 4-10(b) the values of $CT3 - CT1, CT4 - CT2, CT5 - CT3$,…. Q equals 3 now. As the magnitude differences of CMA when $Q = 3$ are very similar to those when $Q = 2$ [compare Figure 4.11(a) and Figure 4.13(a)], it is assumed reusing CMA/CT twice is also a feasible option to further reduce the computational complexity, which may be applied into simultaneously identifying three rails (see Figure 4.14). After every

sampling event, one rail conducts ‘Whole Iteration’ and the other two take ‘Partial Iteration’. The significantly reduced computational consumption is listed in Table 4-7.

As shown in Figure 4.14, after every sampling event, one rail will conduct ‘whole iteration’ and the other two take ‘partial iteration’. Thus, the computational consumption is reduced from three times of a ‘whole iteration’ costs to a ‘whole iteration’ and two ‘partial iteration’ cost, which also saves more than half of the computational efforts of the conventional estimation way (see Table 4-7).

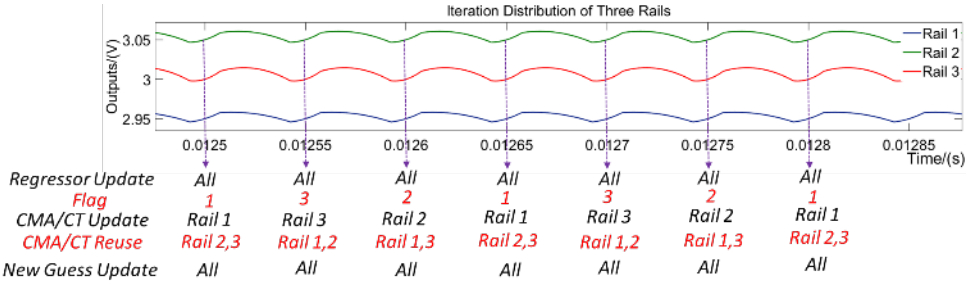


Figure 4.14 The Iteration Distribution of Reusing CMA in a Three-Rail Architecture

4.3.4 Summary of Two Proposed Approaches

To conclude contributions of the two proposed approaches, computational burdens of parameter estimation of a three-rail power converter in different scenarios, using or not using the proposed approaches, are compared in Table 4-7. As the first two rows list, the computational costs of finishing a ‘Partial Iteration’ are even less than half of those of finishing a ‘Whole Iteration’. If a three-rail power conversion architecture is being identified in the conventional way ($K = 1$ or $Q = 1$), after every sampling event, all the three rails would conduct ‘whole iterations’. The corresponding computational costs would be three times the costs of a ‘Whole Iteration’ (see Table 4-7). However, using the iteration decimation approach ($K = 3$) there is only one rail conducting a ‘Whole Iteration’, whilst one rail conducts ‘Whole Iteration’ and the other two take ‘Partial Iterations’ using the CMA substitution approach ($Q = 3$). The computational costs of $K = 3$ and $Q = 3$ are both listed in Table 4-7.

Accordingly, if the sampling frequency is 20kHz, the employed processor should complete 192 additions, 327 multiplications and 3 divisions in every sampling interval (50us) for not using either of the two proposed approaches. This heavy computational burden may cause that all iteration tasks not to be able to finish in a given sampling event. The solution can be to replace the currently used processor with another more computationally capable one. This may guarantee a shorter computation time though, would cause extra hardware investments. However, the proposed approaches may alleviate the burdens in a cost-exempt way. The

computational burden during each sampling interval would be almost halved by using the CMA re-using approach ($Q = 3$), and even be less than halved by staggering iteration actions among the three rails ($K = 3$). As such, neither the extra processor costs nor the reduction of the sampling frequency are needed.

Table 4-7 The Computational Cost of Different Iteration Strategies of the Three Algorithms

Iteration Event	**Computational Complexity**							
	RLS			**AP**		**KF**		
	+	×	/	+	×	+	×	/
a 'whole' iteration	64	109	1	93	96	104	112	1
a 'partial' iteration	20	24		4		24	32	1
$K = 1$ or $Q = 1$ of three rails (Three 'whole')	192	327	3	279	288	312	336	3
$K = 3/2$ of three rails (One 'whole', One 'partial')	84	133	1	97	96	128	144	2
$K = 3$ of three rails (One 'whole')	64	109		93	96	104	112	1
$Q = 3$ of three rails (One 'whole', Two 'partial')	104	157	1	101	96	152	176	3

4.4 Simulation Results

To verify the performance of the proposed approaches, a three-rail power conversion architecture, containing three single buck converters, is simulated using Simulink/Matlab. Moreover, RLS and AP algorithms are implemented to estimate transfer function coefficients of a DC-DC buck converter in (4-1).

Table 4-8 The Circuit Component Values of Each Rail

Parameters	**Rail 1**	**Rail 2**	**Rail 3**
C(μF)	470	330	220
R(Ω)	5	5	10
$\mathbf{V_{out}(V)}$	1.8	3.3	5

According to the small signal modelling method and the further discretion way introduced in **Chapter 2**, the four coefficients of the digital transfer function of each rail, a_1, a_2, b_1 and

b_2 in (4-1), are listed in Table 4-9. The three rails share the same values of circuit components apart from the output voltage, capacitors, and resistance loads, listed in Table 4-8. The circuit components of the converter are designed as: $V_{in} = 10\text{V}, L = 220\ \mu\text{H}, R_C = 25\ \text{m}\Omega, R_L = 68\ \text{m}\Omega,\ f_S = 20\ \text{kHz}$.

Table 4-9 The Transfer Function Coefficients of Each Rail

Parameters	Initial Values		
	Rail 1 (5Ω)	Rail 2 (5Ω)	Rail 3 (10Ω)
a_1	-1.9348	-1.9163	-1.9066
a_2	0.9586	0.95	0.9572
b_1	0.1759	0.2258	0.3099
b_2	0.0624	0.1118	0.1955

4.4.1 The First Approach

Conventionally, iteration events occur in all three rails after every sampling action whereas in the proposed method after the update of regressors of the three rails, only one rail will take the iteration cycle. The distribution of iteration events of each rail is presented in Figure 4.15, where the iteration frequency of each rail in the proposal is three times lower than the sampling frequency. Here, a Flag is allocated to indicate which rail is taking places to process with the iteration action.

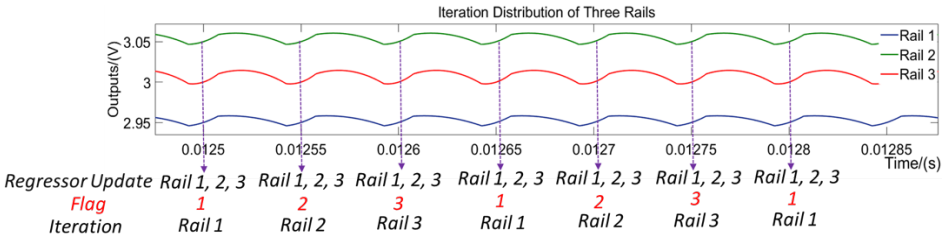
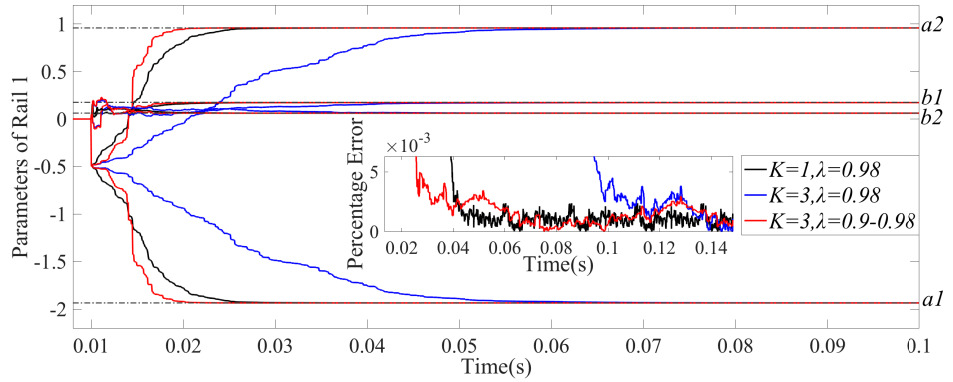
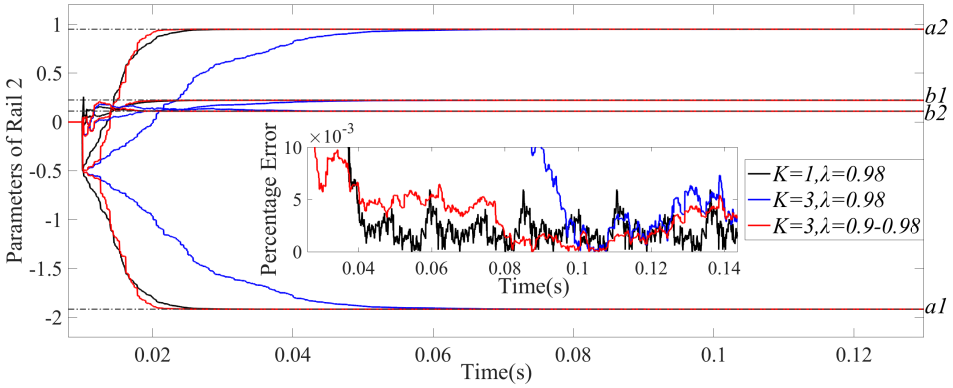


Figure 4.15 The Iteration Distribution of Lowering the Iteration Frequency in a Three-Rail Architecture

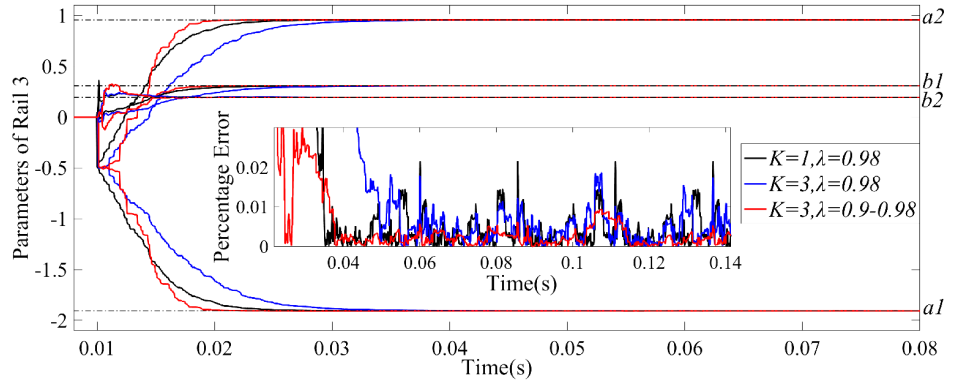
Figure 4.16 and Figure 4.17 respectively present the recursive curves of RLS and AP used to estimate the parameters of the three-rail power conversion architecture in different scenarios: in the conventional way ($K = 1, \lambda = 0.98$ or $\mu = 0.3$), in the proposed iteration frequency reduction way without the adjustment of λ or μ ($K = 3, \lambda = 0.98$ or $\mu = 0.3$) and in the proposed way with a lower λ or μ in Stage 1 ($K = 3, \lambda = 0.9 - 0.98$ or $\mu = 0.9 - 0.3$).



(a)



(b)

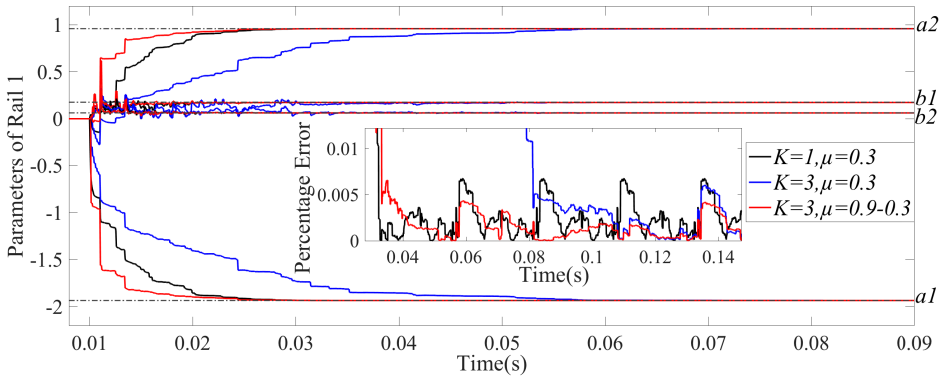


(c)

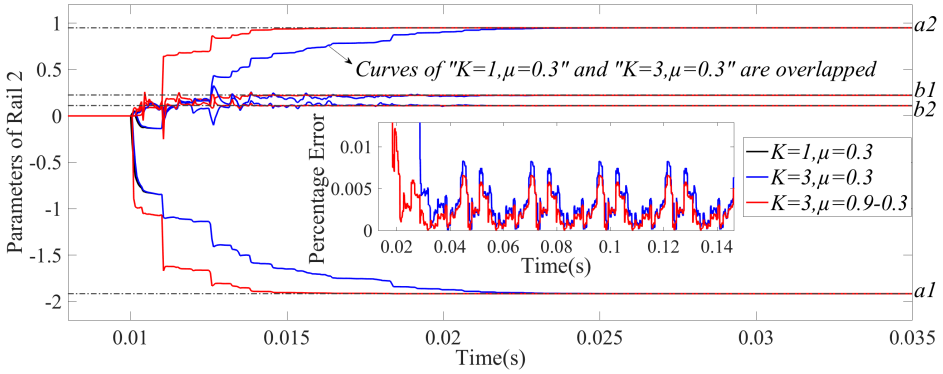
Figure 4.16 The Estimation Curves in Different Iteration Frequencies Using RLS

As mentioned in **Section 4.3**, the number of iterations for the estimated parameters to converge is dependent on the performance of the applied algorithms and the complexity of the circuit model identified. It is found that the classical RLS algorithm iterates for about 200 times to come out with reliable coefficient values of a second order linear transfer function (4-1). As shown in Figure 4.16, with the same forgetting factor (λ) value, 0.98, the speed of minimizing the error when iteration frequency (IF) equals sampling frequency (SF), the scenario of $K = 1$, is three times faster when IF is one third of SF ($K = 3$). According to the proposed approach,

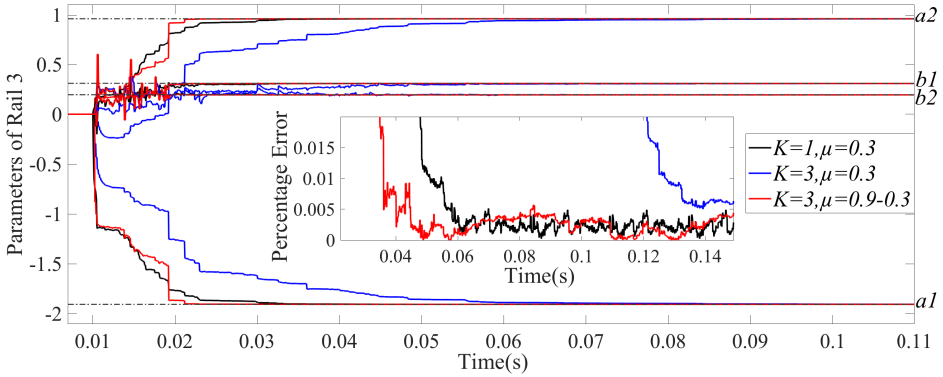
after reducing λ only in Stage 1 of the low IF identification processes from 0.98 to 0.9, the convergence times are correspondingly shortened and estimation error keeps the same level with that of $K = 1$ in all the three rails (see Figure 4.16). In Figure 4.17, the step size, μ, of AP also should be tuned larger in Stage 1 to shorten the prolonged convergence time by $K = 3$ from 0.3 to 0.9, and the estimation error would not be influenced. Therefore, through being demonstrated in both AP and RLS, the proposed approach of reducing IF and adjusting λ or μ outperforms the conventional way in terms of two third less computational cost without compromises in convergence time and estimation accuracy.



(a)



(b)



(c)

Figure 4.17 The Estimation Curves in Different Iteration Frequencies Using AP

4.4.2 The Second Approach

The iteration distribution arrangements of reusing CMA/CT once ($Q = 2$) and twice ($Q = 3$) are separately presented in Figure 4.18 and Figure 4.14. The Flag value indicates which converter is to be identified after a sampling event. The experimental validation of reusing CMA/CT once is a combination between 'Reducing Iteration Frequency' and 'Recycling CMA/CT'. Shown in Figure 4.18, in every sampling interval, one rail accepts the 'Whole Iteration', whilst the other one takes 'Partial Iteration' and the left one is not allocated with iterations which therefore holds its results until Flag indicates its updates. Because the iteration frequency is 2/3 of the sampling frequency now, this scenario is described as $K = 3/2$ rather than $Q = 2$. As Table 4-7 shows, the computational costs of $K = 3/2$ is reduced from three times of a 'Whole Iteration' costs to a 'Whole Iteration' and a 'Partial Iteration' cost, which saves more than half of the computational effort of using the conventional estimation way ($Q = 1$). With lowering the iteration frequency, in simulation λ is lowered from 0.98 to 0.92 (RLS) and μ is increased from 0.3 to 0.6 (AP) in Stage 1 to shorten the prolonged convergence time.

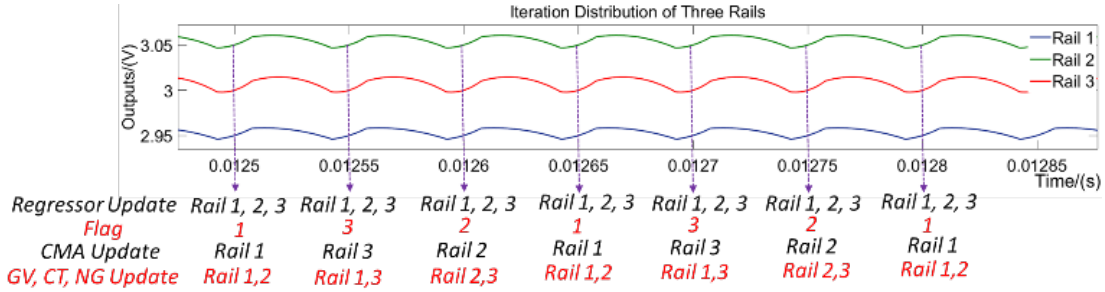
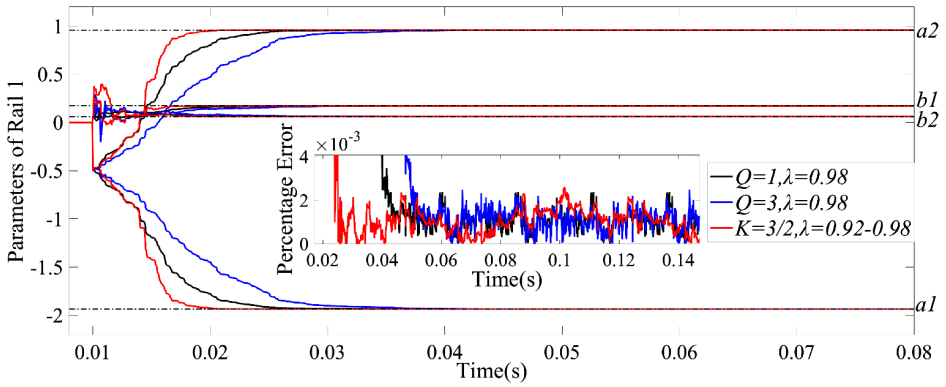
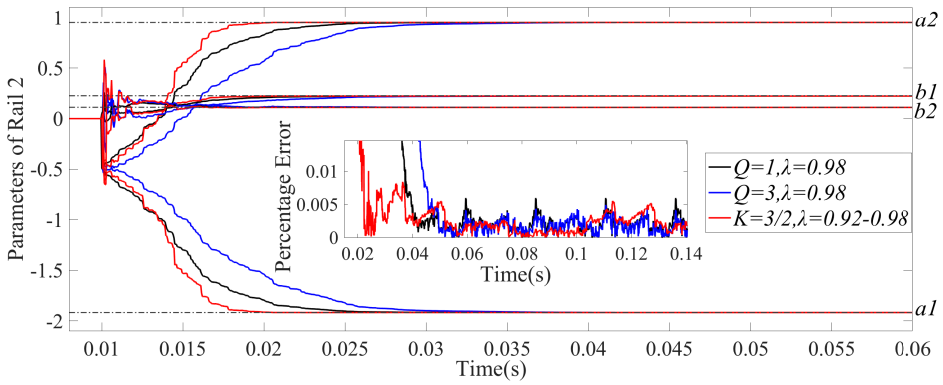


Figure 4.18 The Iteration Distribution of Reusing CMA in a Three-Rail Architecture

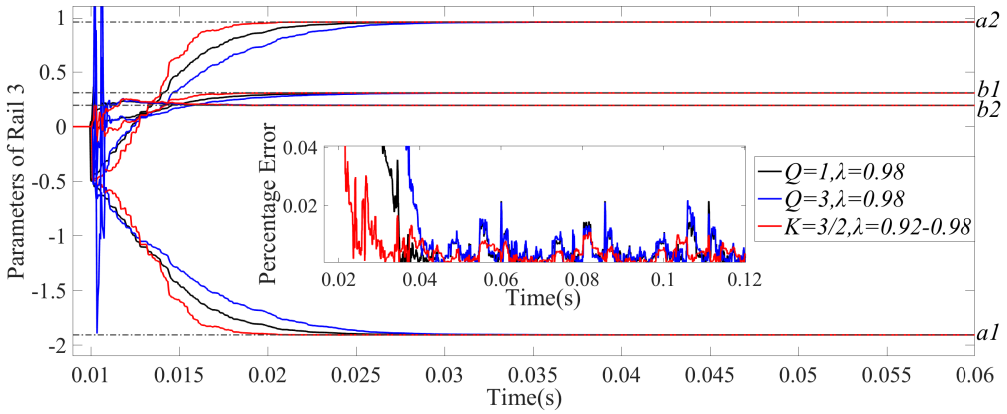
As shown in Figure 4.14, re-using CMA/CT twice ($Q = 3$) indicates that after every sampling event one rail conducts 'Whole Iteration' and the other two take 'Partial Iteration', the corresponding computational cost becomes a 'Whole Iteration' and two 'Partial Iteration' cost (see Table 4-7).



(a)



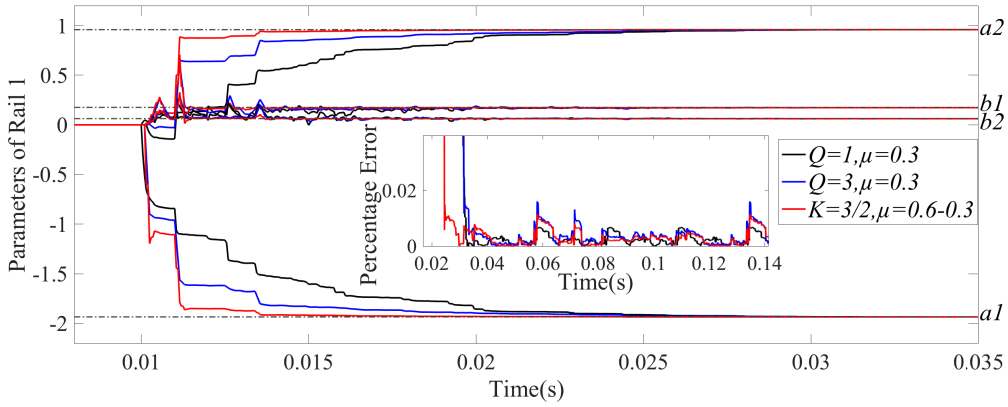
(b)



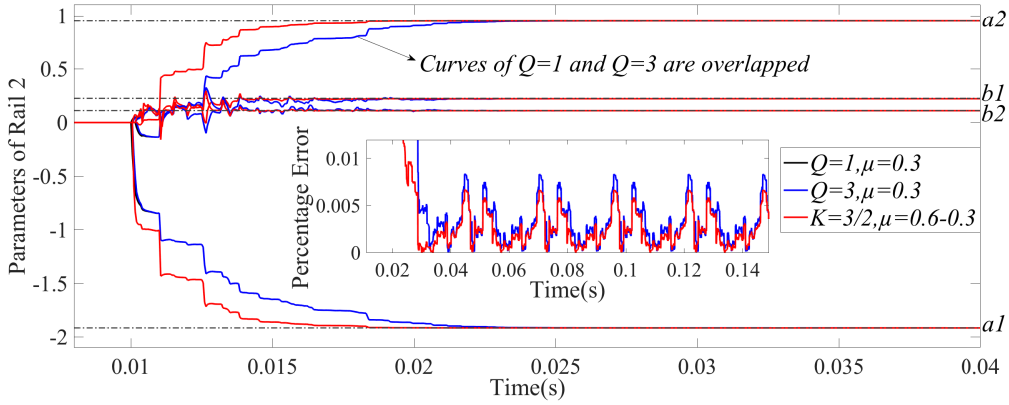
(c)

Figure 4.19 Estimation Curves in Different CMA Reusing Strategies Using RLS

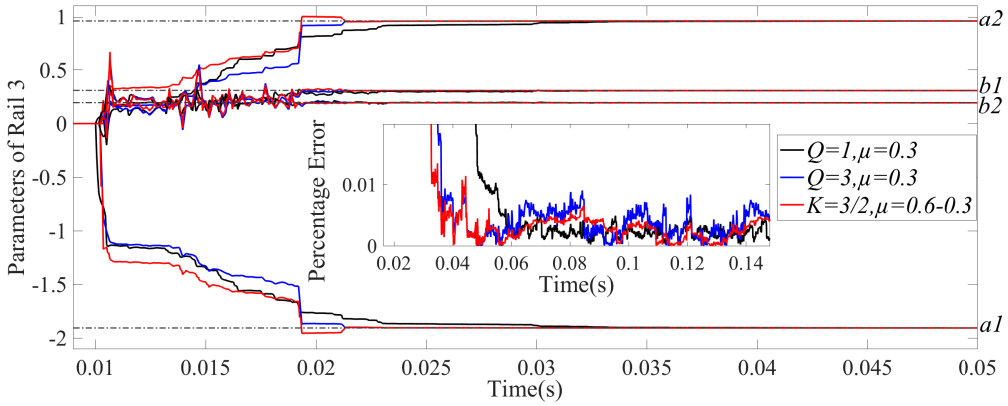
Figure 4.19 (RLS) and Figure 4.20 (AP) highlight the iteration update sequence for each rail when $K = 3/2$ (re-using CMA/CT once) and $Q = 3$ (re-using CMA/CT once). In both cases, the sequence is benchmarked against the default of $Q = 1$. Apart from Rail 2 using AP [see Figure 4.20(b)], under the same convergence-time-related factors ((λ =0.98 in RLS and μ = 0.3 in AP)), the convergence of recursive curves of $Q = 3$ is slightly slower than that of $Q = 1$. By reducing λ in RLS or increasing μ in AP only in Stage 1, the convergence time of $K = 3/2$ is shortened and the estimation accuracy would not be affected as λ or μ is still large/small enough in Stage 2 to guarantee their percentage estimation error keeping the same level with that of $Q = 1$. As shown in Table 4-7, the computational complexities at $Q = 3$ or $K = 3/2$ is only slightly higher than those at $K = 3$, whilst the scenario of $K = 3$ needs a significant decrease/increase of λ or μ. Overall, reusing CMA/CT, which may be also combined with the iteration decimation approach, will be acceptable for some applications where computational complexity and cost may be the main barrier to implementation.



(a)



(b)



(c)

Figure 4.20 Estimation Curves in Different CMA Reusing Strategies Using AP

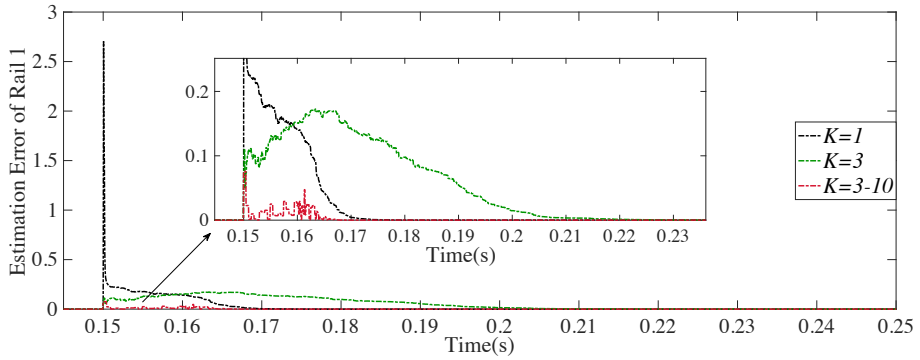
4.4.3 Load Changes Rejection

When system variations, such as frequent and/or periodic load changes in SMPCs, occur during system identification processes, the ability to reject abrupt system changes to make recursive curves quickly updating new results is important [240]. To validate the disturbance rejection ability of locally reducing the iteration frequency, load changes are configured in Rail 1 (the output current changes from 0.36A to 1.8A), Rail 2 (from 0.66A to 3.3A), and Rail 3

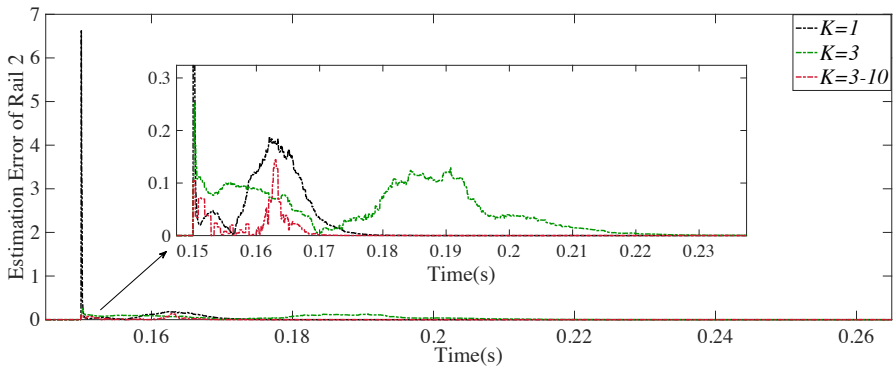
(from 0.5A to 2A) at 15ms. After the load variations, the four model coefficients of the three rails change to the values listed in Table 4-10. Figure 4.21 (RLS) and Figure 4.22 (AP) present the estimation curves with or without locally removing the sampled transient responses coping with abrupt load changes. As all iterated New Guess (NG) when the recursive curves are converged might be considered (averaged) to acquire the estimation results, the transient responses in recursive curves caused by the disturbances in the sampled output voltage and duty cycle should be as insignificant as possible. Compared with the scenarios of $K = 1$ and $K = 3$, the recursive curve of temporarily decreasing K to 10 during the disturbance ($K = 3 - 10$) enters the accuracy tolerance range, ±5% of real values, faster than other ways do and features the most moderate transient behaviour.

Table 4-10 The Transfer Function Coefficients of Each Rail after Load Changes

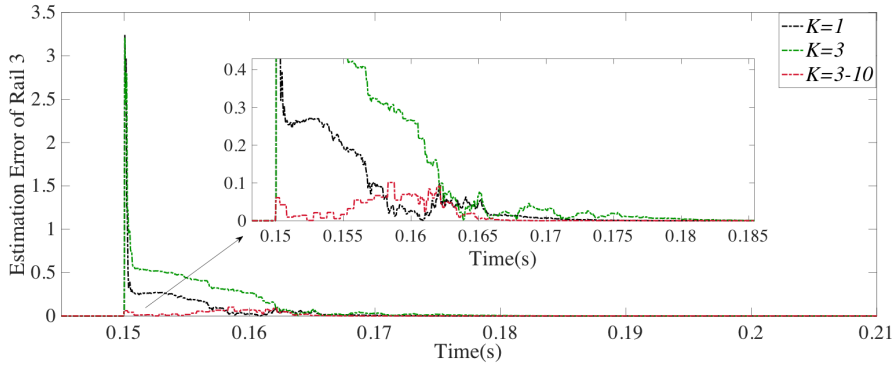
Parameters	**After Load Change**		
	Rail 1 (1Ω)	**Rail 2 (1Ω)**	**Rail 3 (2.5Ω)**
a_1	-1.8591	-1.8117	-1.8454
a_2	0.8827	0.8447	0.8949
b_1	0.1761	0.2234	0.3063
b_2	0.0603	0.1058	0.1887



(a)

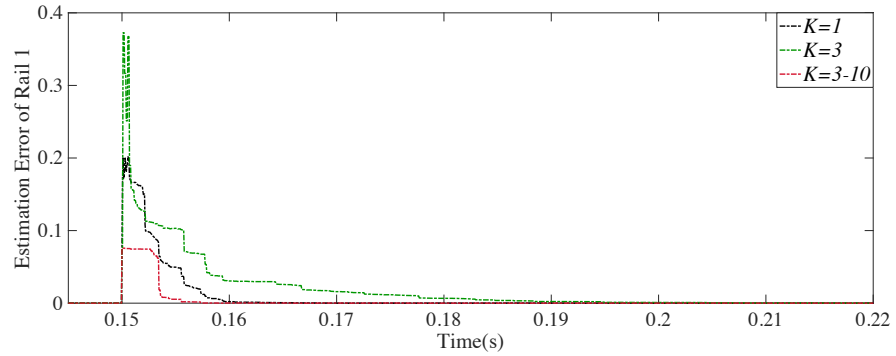


(b)

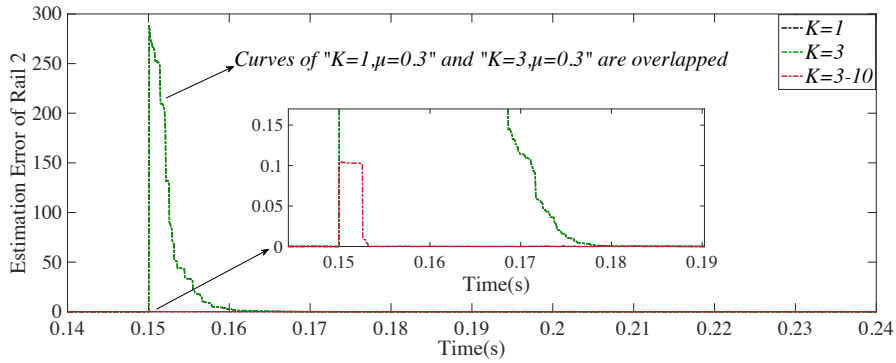


(c)

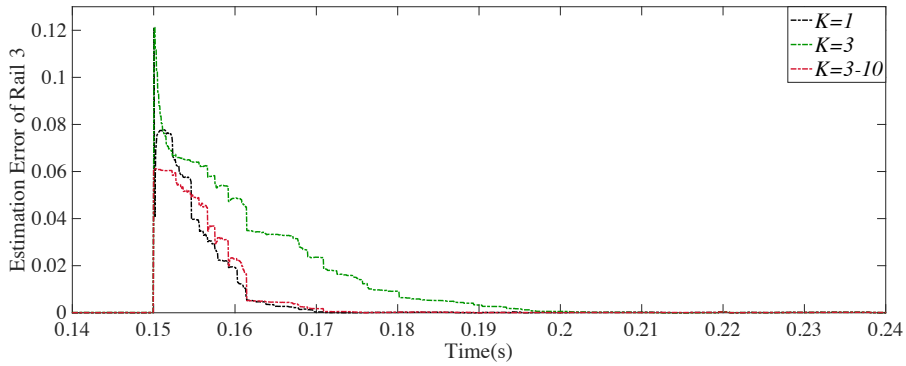
Figure 4.21 Estimation Curve Comparisons of $K = 1$, $K = 3$ and $K = 3 - 10$ (RLS)



(a)

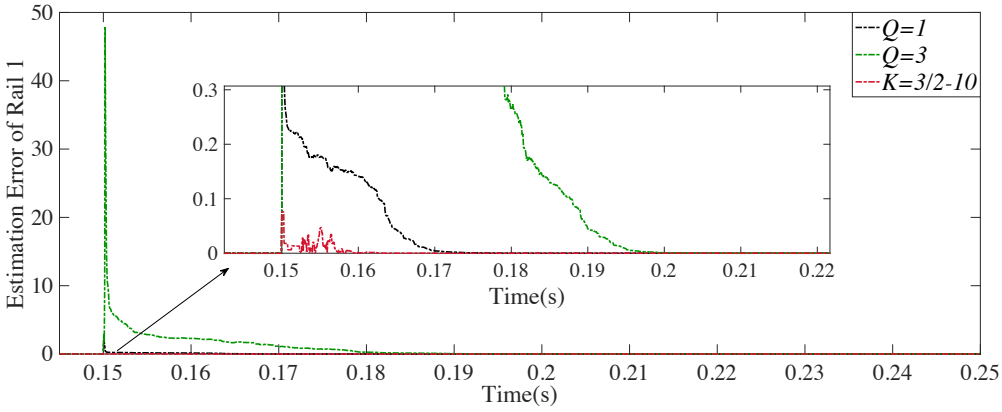


(b)

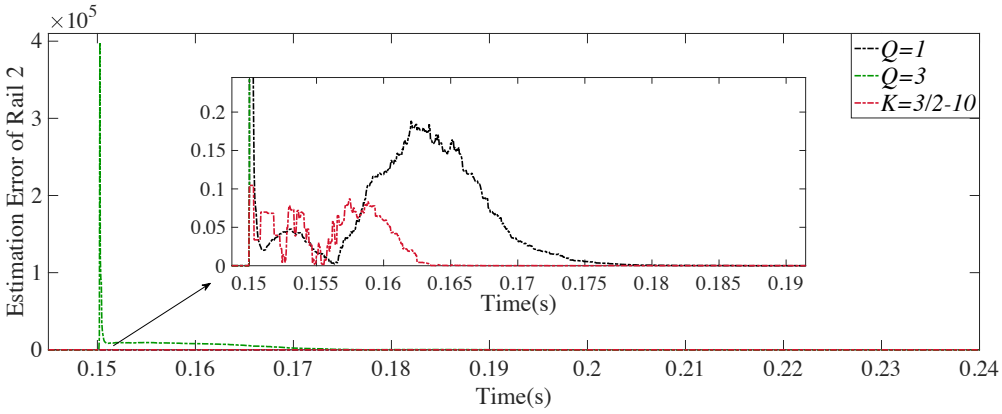


(c)

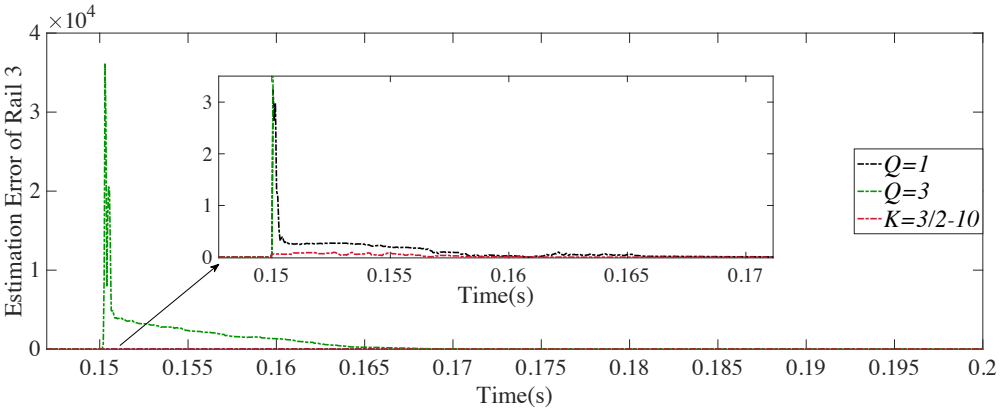
Figure 4.22 Estimation Curve Comparisons of $K = 1$, $K = 3$ and $K = 3 - 10$ (AP)



(a)



(b)

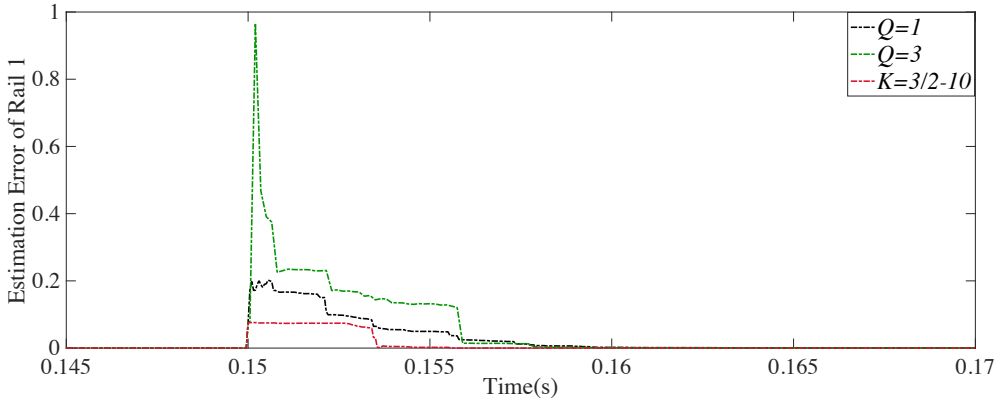


(c)

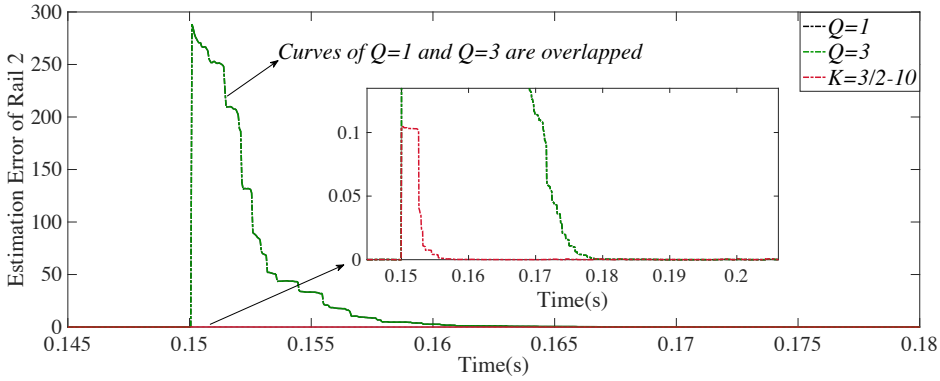
Figure 4.23 Estimation Curve Comparisons of $Q = 1$, $Q = 3$ and $K = 3/2 - 10$ (RLS)

Figure 4.23 (RLS) and Figure 4.24 (AP) present the comparison between reusing CMA/CT and locally decreasing the iteration frequency to deal with sudden load changes. CMAs/CTs respectively calculated from two consecutive iterations contain differences already, the transient responses of voltage and duty cycle signals to reject load changes would make the differences larger. Consequently, the CMA/CT updated in the last iteration cycle will not be usable for the current one, proved by the severely fluctuated transient behaviour in the recursive

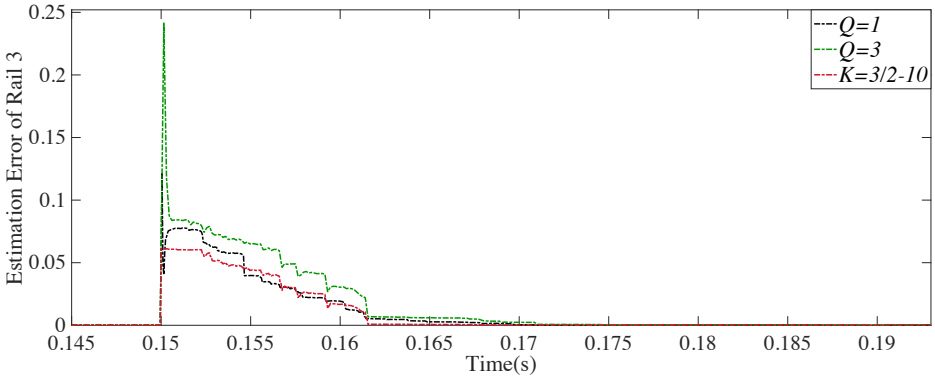
curve of $Q = 3$. Therefore, iteration frequency is also locally reduced as $K = 10$, here for removal of sampled disturbances (see Figure 4.23 and Figure 4.24), both convergence time and transient behaviours are improved.



(a)



(b)



(c)

Figure 4.24 Estimation Curve Comparisons of $Q = 1$, $Q = 3$ and $K = 3/2 - 10$ (AP)

4.5 Conclusion

As the current parameter estimation process of multi-rail power converters suffers huge computational burdens, either the overall costs or the burden in every sampling event. This chapter introduces two computational-cost-saving approaches. Firstly, the iteration frequency,

conventionally equalling to the sampling frequency, is proposed to be reduced. However, this approach would cause the convergence time of parameter estimation longer than that of using the conventional way, consequently, delicate adjustments of the convergence-time-related factors of the applied algorithms (μ in AP, r in KF and λ in RLS) are necessary. Estimation accuracy will not be significantly affected as long as the factors are tuned for speeding the estimation up only when the recursive curves have not been converged yet. Besides, through locally decrease the iteration frequency, the transient responses coping with abrupt load changes in the sampled data will be removed, which could avoid the recursive curves suffering severe transient behaviours during parameter estimation. The second proposed approach is to reuse the CMA/CT calculated in the last iteration cycle to the current, or even in the next one, as the computational costs spent on updating CMA/CT are more than half of those spent on completing an entire iteration cycle. A three-rail power conversion architecture is simulated and estimated to demonstrate that the two proposed approaches can be applied on various adaptive filters (AP and RLS here). According to the simulated results, there is no noticeable compromises of estimation performances when the two approaches are applied, compared with the results acquired in the conventional way.

Chapter 5 Experimental Validation

5.1 Introduction

To verify the two proposed approaches in Chapter 4, the hardware employed includes three synchronous DC-DC buck converters operating in parallel (a three-rail power conversion architecture), a Digital Signal Processor (DSP) where individual digital controllers are programmed to build the closed-loop control systems of each rail (a single buck converter). If the parameter estimation section is implemented in the DSP, i.e. realising real-time estimation, the algorithms should be coded in the DSP where recursive processes take place, which would result in an overburdened computation tasks for DSP. Specifically, if RLS was applied, to achieve the result comparison between the proposed approach and the conventional one, in one sampling interval (50μs) the iteration of RLS should execute five times, that the DSP of TMS320F28335 – would spends 170μs – is inadequate to complete. Real time estimation also encounters a concurrent collection of a large amount of data, as not only the input and the output of three converters are sampled, but the recursive results of every iteration cycle are collected. There are 36 groups of the results (3 comparison groups, 3 rails and 4 parameters of each rail), that should be concurrently stored. The two tremendous computation and data collection/storage burdens need a processor more capable, with a higher cost certainly, than the DSP (TMS320F28335). Therefore, a cost-efficient way, that conduct identification algorithms off-line with the real-time sampled voltage and duty cycle signals, adopted in most existing studies, could make TMS320F28335 being computationally sufficient to complete the experiment verification of this project. As such, the only data that need to be processed by the DSP is the input and output signals of converters. Changing the execution place of the estimation algorithms from the DSP to Matlab blocks would avoid the two afore-pointed tasks, but would make no difference of the practical results. This chapter contains the detailed introduction of the experimentally employed prototype hardware, the DSP code development tool and the specific experimental setups, such as component parameters of buck converters, initial coefficient configurations of algorithms and filter selections, etc. Most importantly, this chapter presents the practical validation of the two novel computational complexity alleviation approaches on different adaptive filters used for real time parameter estimation.

5.2 Hardware

5.2.1 The Employed DSP

Digital Signal Processors (DSPs) have been dominantly used in building control systems of various power conversion topologies and motor drives for operation optimization. Their

delicate design brings the great convenience into system programming, enabling various advanced algorithms to be implemented. The DSP employed in this project is a floating point processor, TMS320F28335 [a member of Delfino™ C2000 DSP series from Texas Instruments (TI)], of which the operation frequency can be as high as 150MHz. Before this project, it has been selected by many researchers into the experimental verification work of real-time parameter estimation [28, 179, 189].

Figure 5.1 shows the Harvard-architecture-based TMS320F28335 platform. This platform contains a 512KB flash memory, a 68KB RAM and a 6-channel Direct Access Memory (DMA). The core of the processor comprises Arithmetic Logic Unit (ALU), 32×32-bit multipliers, 32-Bit Floating Point Unites (FPU) and 32-bit Timers. Serially-communicated peripherals with the processor include General Purpose Input/output (GPIO), 6-channel/32-bit enhanced Capture Input (eCAP), 12-bit/16-channel ADC, 18-output-channel ePWM and several communication interface circuits, such as the Enhanced Controller Area Network (eCAN), Serial Peripheral Interface (SPI) and Serial Communication Interface (SCI), etc. In these modules ADC and ePWM are the two most frequently used peripherals.

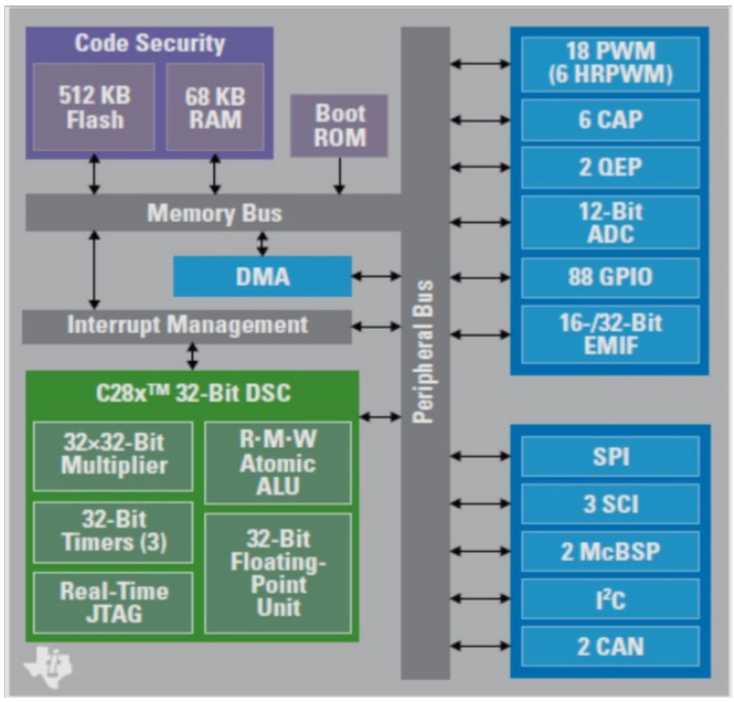


Figure 5.1 TMS320F28335 eZdsp Architecture [241]

The ADC is used to measure the analogue voltage signals ($V_{Analogue}$) and convert them to digital numbers which can be processed in the processor, e.g. control signal computation. The range of the analogue voltage sampled by the ADC of TMS320F28335 is from 0 to 3V. In each

sampling instant, the converted digital voltage ($V_{Digital}$), suffering low-quantization effects due to the internal 12-bit resolution, is computed as (5-1) shows:

$$V_{Digital} = \frac{V_{Analogue} \times 2^{12}}{Analogue\ Voltage\ Range} = \frac{V_{Analogue} \times 4096}{3} \quad (5\text{-}1)$$

In TMS320F28335, the ADC includes its independent clock [12.5 Mega Samples per Second (MSPS)], a built-in dual sample-and-hold (S/H) circuit (12-bit core), 16-channel, multiplexed inputs and 16 result registers for storage of the converted data. It features a quite rapid conversion speed (25MHz) and two sampling modes (simultaneous or sequential mode).

There are two 8-state sequencers in the ADC of TMS320F28335, which may independently operate and also be connected in series as a 16-state sequencer. To start the conversion operations, a trigger signal is typically generated by event manager model (there are two independently-worked ADC triggering events: EVA and EVB), or an external trigger signal can be set up through the GPIO module.

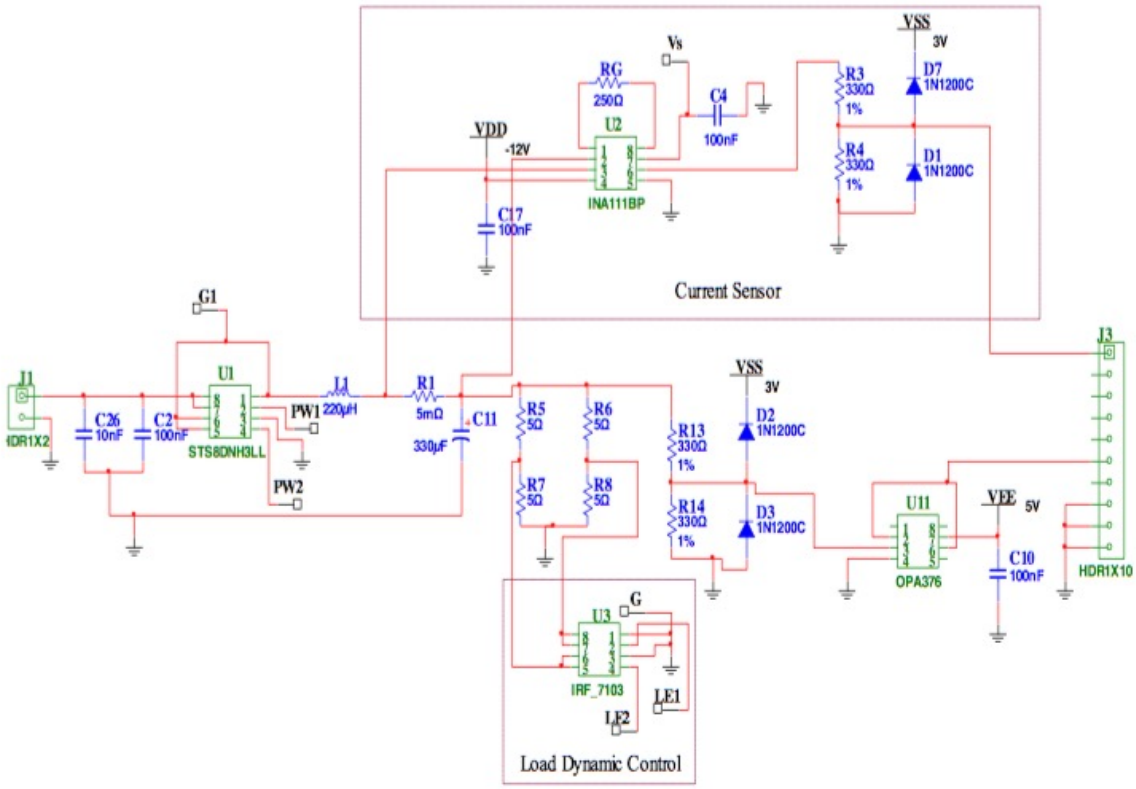
The 16-bit enhanced PWM module of TMS320F28335, as the tool of control signal computation, is essential in almost all control applications, e.g. controls of digital motor, Switch Mode Power Supply (SMPS), and uninterruptable power supplies (UPS), etc. Opposite to the function of the ADC, PWM typically converts the analogue duty cycle signals to ePWM signals, through comparing the associated control signal with the timer registers in compare registers.

The PWM module of TMS320F28335 contains 6 independently-programmed dual channels for symmetric/asymmetric PWM signal generation. Besides, there is a 16-bit purpose timer in each event manager module, which may be configured as up/down counters for PWM operation emulation. In this project, the timer is 150MHz, the carrier signal (PWM frequency) is 20kHz and the counting mode is up-down (symmetric carrier waves), then the Time-Base Period Register (TBPRD) should be configured as 3750. Besides, the start of sampling actions of ADC is triggered when Time-Base counter equals the pre-configured TBPRD, the middle point (the upper peak) of the triangle carrier wave, to circumvent the switching sparkling disturbances. As most of electric topologies include switch bridge circuits, e.g. DC-DC converters, etc., to avoid both switches to concurrently conduct [241], the dead-band configuration is prepared for each PWM output pair in the PWM module of TMS320F28335 with the minimum duration of 6.67ns (a device clock cycle). In this project, the deadtime is 1μs. For extension, the eCAP module may also produce 6 PWM channels in the TMS320F28335 processor.

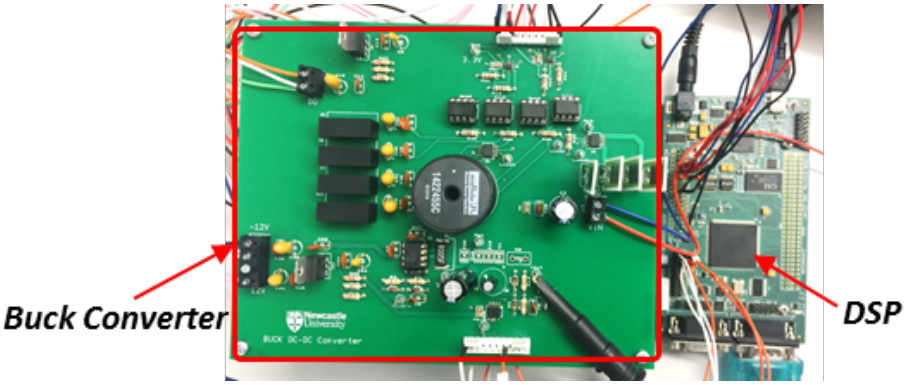
5.2.2 A Multi-Rail Power Conversion Architecture

There are three single-phase synchronous DC-DC buck converters connected in parallel to build up a multi-rail power conversion architecture. Figure 5.2(b) presents the prototyped DC-

DC buck converter, which comprises DC-input voltage source with decoupling capacitors, a power stage filter (LC circuit) and output load resistors, and of which the switching devices are two N-channel MOSFETs (STS8DNH3LL). Apart from the circuit which realises the basic function of a buck converter, the corresponding PCB board also comprises two signal measurement circuits for the measurements of the regulated voltage outputs and inductor currents. The measurement circuit includes a voltage output divider circuit for ADC (or DSP) protection, a Schottky-Diodes (BAT 85) to guarantee that the sampled voltage would not exceed the full scale of the ADC, and a unity-gain Fast Operational Amplifier (OPA376) as a buffer protection circuit. The output voltage of the buck converter will be scaled down to be half of its original value by the divider circuit firstly, and then pass through the diode and amplifier before being sampled by the ADC. The inductor current measurement, employing a series shunt resistor with a fast instrument amplifier (IN111BP) rather than adding a Hall-effect transducer, follows a similar process. As a result, both the space and the costs of the printed circuit board are saved. Nevertheless, the current measurement section is not used as the parameter estimation process in this project only needs duty cycle and voltage signals.



(a)



(b)

Figure 5.2 The Circuit Diagram of the Single Buck Converter (a) and The PCB Board of Buck Converter with DSP (b)

There are two load branches connected in parallel with the output of the buck converter. Each branch has two chip resistors of 5Ω for fast heat dissipation as Figure 5.2(a) shows [133]. Originally, the normal operation is to use all the four resistor slots to produce an equivalent load resistance equalling 5Ω. Thus, the load can be varied to 2.5Ω and 3.3Ω for validating the proposed approaches under the scenario of abrupt load changes. However, only one load branch is used in this project to realise more load variations. In Rail 1 and Rail 2, two resistance loads with 4Ω and 1Ω are linked in series, and in Rail 3, both the two branches are applied to change the load from 10Ω to 2.5Ω.

5.3 DSP Code Development Tool and Experiment Setups

5.3.1 DSP Code Development Tool

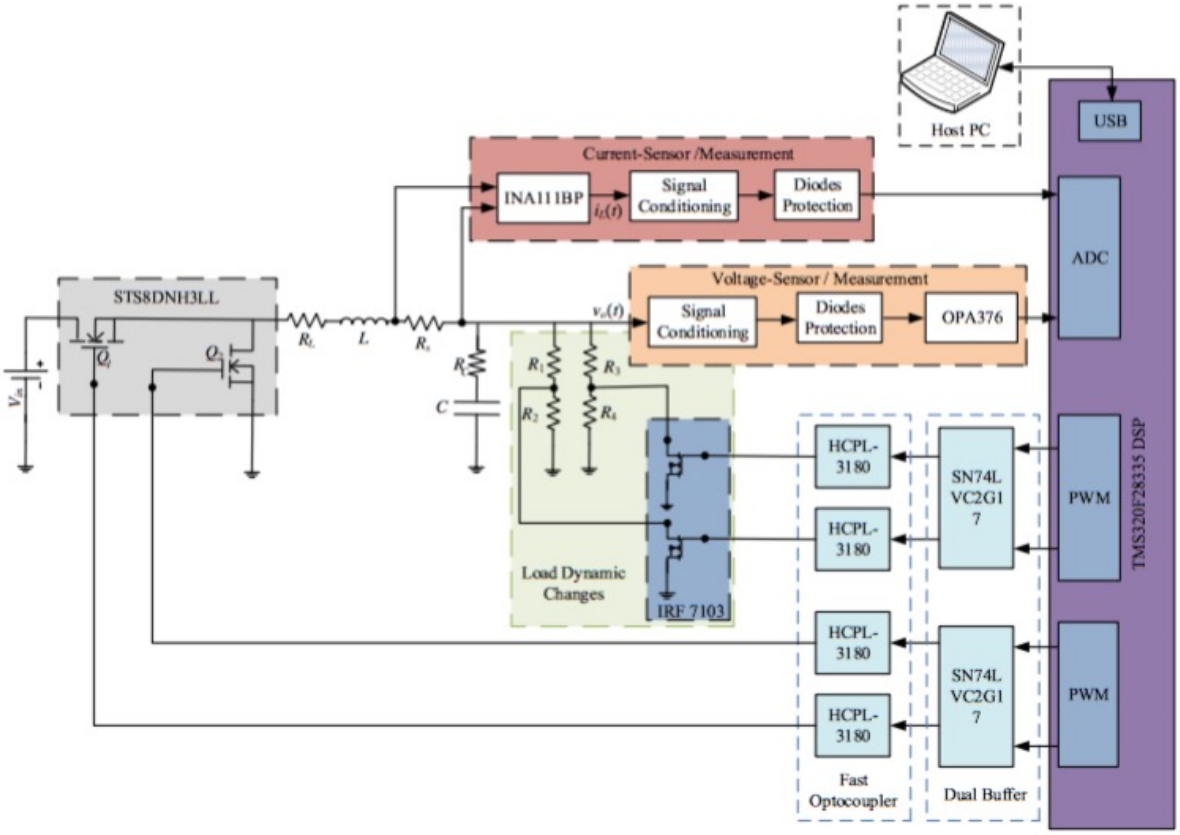
To implement and experimentally validate the proposed iteration decimation approach and CMA/CT reusing approach on the TMS320F28335 eZdsp, Texas Instruments Code Composer Studio (CCS) based Integrated Development Environment (IDE) is employed on the host PC as it may achieve programming (C language), code compiling and download the compiled files to the target DSP. For the input/output module (ePWM, ADC and SCI peripherals, etc.) configurations, rather than programming in the CCS, the Embedded Coder Support package for C2000 in which these modules are available for configuration, developed by Matlab/Simulink, is used as it can convert Simulink blocks to executable C code for CCS.

Furthermore, External Mode in Matlab allows the configured Simulink blocks synchronously running with the operation of DSP, as long as an RS232 communication SCI_A is used to connect Simulink and DSP. In this mode, users may supervise and adjust the parameters of experimental setups, such as algorithm factors and controller coefficients, etc., in real time without the need to interrupt the code operation. Besides, the Scope block from

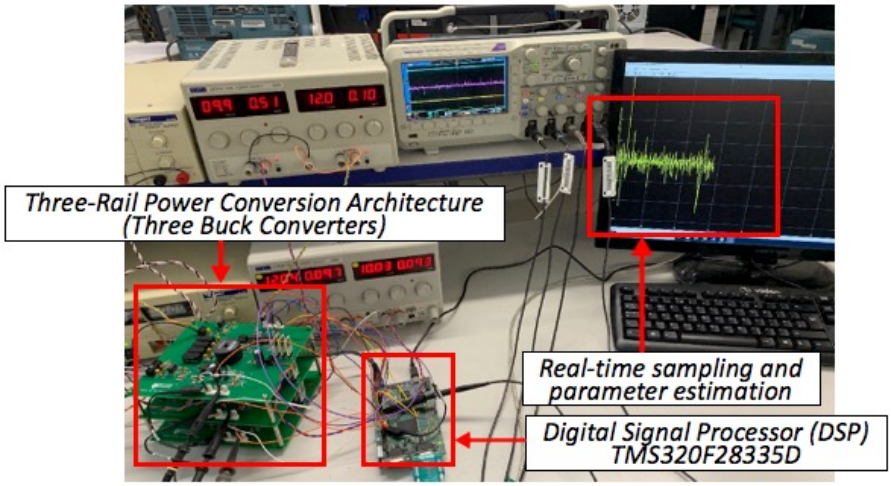
Simulink Library can be used for real-time signal monitoring and display. As such, the deep integration among simulation, experiment design/operation, and real-time amendments/control can be realised. Consequently, the efficiency of the experiment preparation are significantly improved.

5.3.2 Experimental Setups

Figure 5.3(a) shows the control process of a single buck converter, similar with that of multi-rail power conversion architectures in which only two extra channels of these control-loop-related modules should be further developed. The input signals of the DSP (the output voltage of the buck converter) is sampled by the built-in ADC firstly. The sampled data is then processed by the control algorithms pre-downloaded from CCS for producing duty-cycle signal, which would be converted to the ePWM signal in PWM module. After passing through a dual buffer circuit (SN74LVC2G17), delicately selected to generate an output-matching level to the DSP-PWM output voltage for PWM channel protection, the buffered PWM signal would be amplified in the isolated Date Drives (HCPL-3180). The dynamic load circuit is activated by two GPIO pins. Figure 5.4 shows the PWM signals (controlling the MOSSFET of the upper bridge of the buck converter) and the steady-state output voltage signals of each rail in the three-rail power conversion architecture used for experimental validation. The duty cycles of each rail are 18%, 33% and 33% respectively.



(a)



(b)

Figure 5.3 The Parameter Estimation Process Diagram [133] (a) and The Experiment Rigs (b)

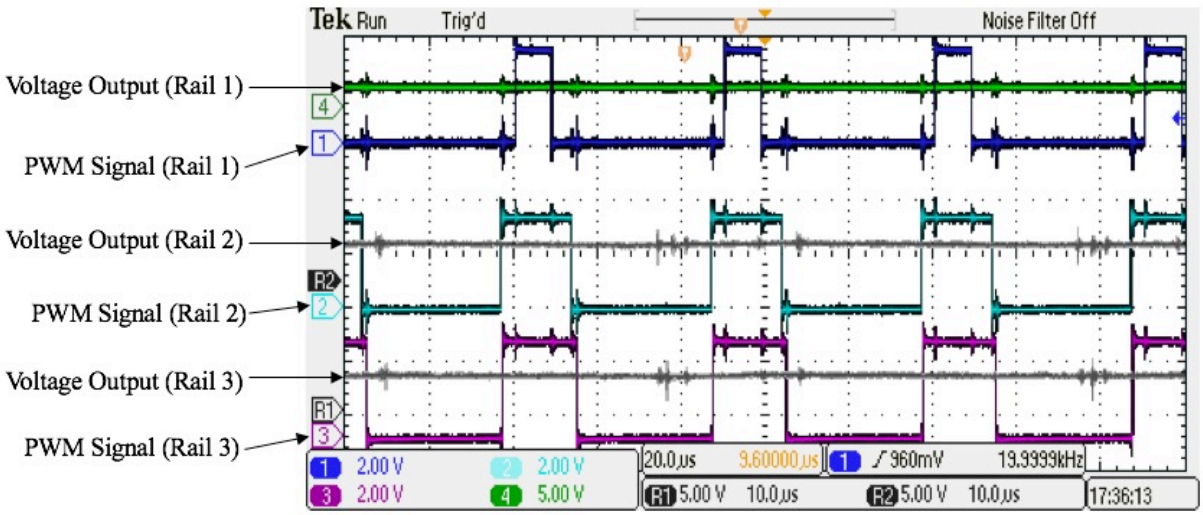


Figure 5.4 The PWM Signals and the Steady-State Output Voltage Signals of Each Rail in the Three-Rail Power Conversion Architecture

As shown in Figure 4.1 in Chapter 4, each rail (Buck Converter) is controlled by independently tuned digital PI controllers. The transfer function of the controller is shown below:

$$G_c(z) = \frac{0.41 - 0.4z^{-1}}{1 - z^{-1}} \tag{5-1}$$

System excitation is necessary for real time parameter estimation. An 11-bit PRBS of which the total data length is 2047 and the sequence length is 102ms under the sampling frequency equalling 20kHz, therefore, is injected in the control (duty-cycle) signal for 100ms. The magnitude of the injected PRBS is ±0.025. The perturbation in output voltages of three rails caused by PRBS injection are shown in Figure 5.5. At the perturbation period, 800 samples of the control signals and the output voltages for the three rails are collected at the sampling frequency equalling 20kHz.

The measured data from the ADC is stored in the DSP memory, and exported to MATLAB for post-processing after the full test sequence of PRBS has been applied to the power converter. Before going into algorithm blocks, sampled duty cycle and voltage signals should be filterd to remove unwanted high frequency measurement noises and be only in the estimation-relevant frequency range. here Moving Average Filters (MAFs) are used for this purpose (the specific filter selection process is collected in Appendix B). Accoding to pieces of literetures and previous work, there is an accuracy tolerance band, ±5% of expected results, is introduced, in which the estimated results can be considered as acceptable. From the measured data, the three applied algorithms, RLS, AP and KF, perform the cycle-by-cycle parameter estimation algorithm previously descibed to estimate the tap-weights of the transfer functions of each rail

and minimise the prediction error signals. The experimental parameters and setups, including iteration distribution arrangements, initial value configurations of the employed algorithms, etc., are chosen to be the same with the simulation setups.

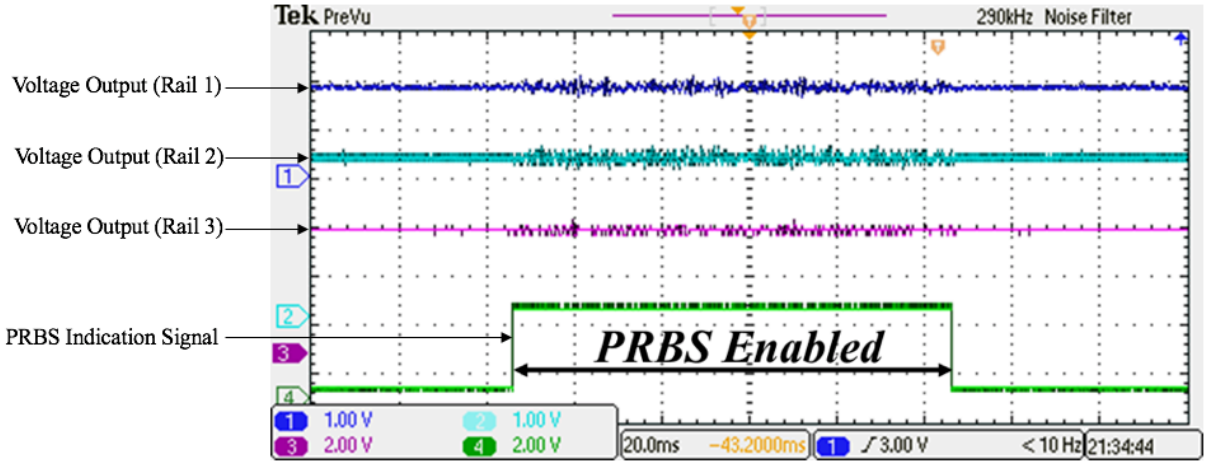


Figure 5.5 The Three Output Voltages from The Multi-Rail Power Converter with PRBS Effects

5.4 Experimental Validation

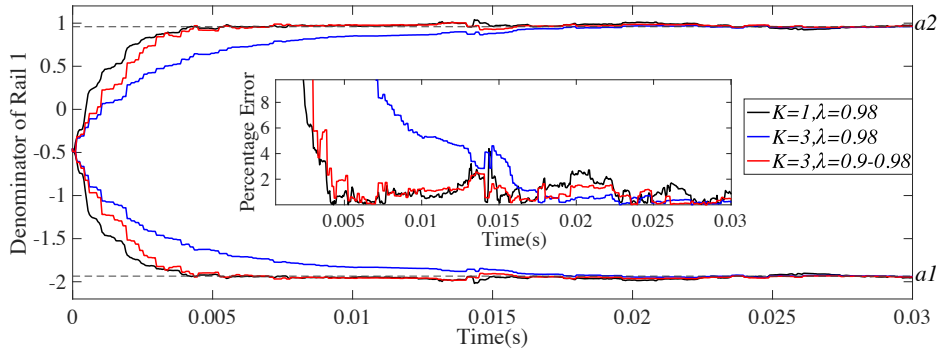
The experimental validation includes performance comparisons, in terms of estimation accuracy, computational costs and Convergence Time, of 1. Two iteration frequencies respectively equalling to the sampling frequency and one-third of it. 2. The same CMA being once, twice and thrice used in consecutive iteration intervals. 3. Abrupt disturbance rejection with or without locally and temporarily disposing of transient responses caused by sudden load changes.

5.4.1 The Iteration Decimation Approach

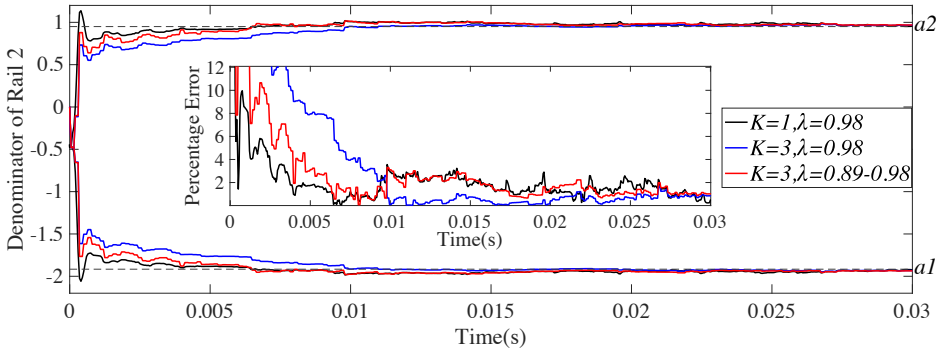
The iteration event distribution of each rail is the same with that used for simulation validation in Chapter 4 (see Figure 4.12 in Chapter 4).

To clearly analyse the estimation accuracy, estimation errors are considered in two ways described as ‘Average Error (AE)’ and ‘Process Error (PE)’. AE means the offset between the true value of power converters’ parameters and the average, averaging estimated results from the point that recursive curves start to enter and remain in the accuracy tolerance band (±5% of real values) to the end of the estimation process (here, at 0.03s). As this average is typically the results that controller retuning is based on, AE could express the performance of adaptive control. PE stands for fluctuations of recursive curves, which can indicate the stability/robustness of the estimation process, so does the Variance of recursive curves.

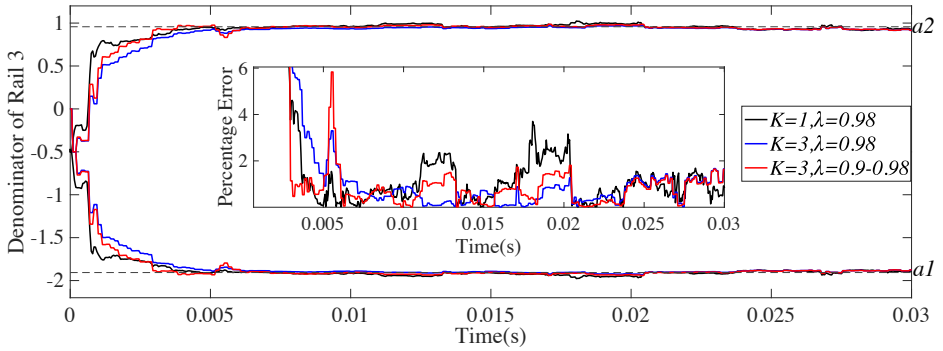
Therefore, an estimation approach very capable of noise rejection would feature recursive curves with low PE and small variations.



(a)

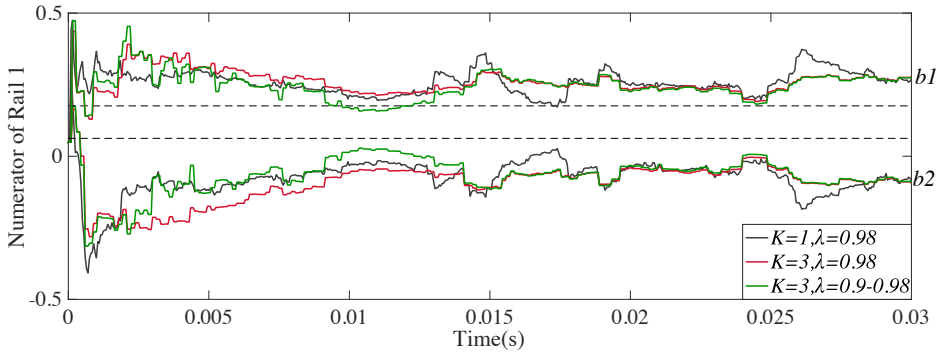


(b)

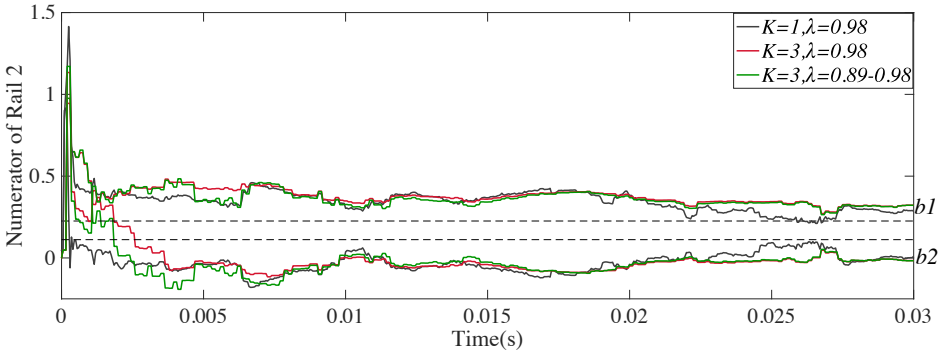


(c)

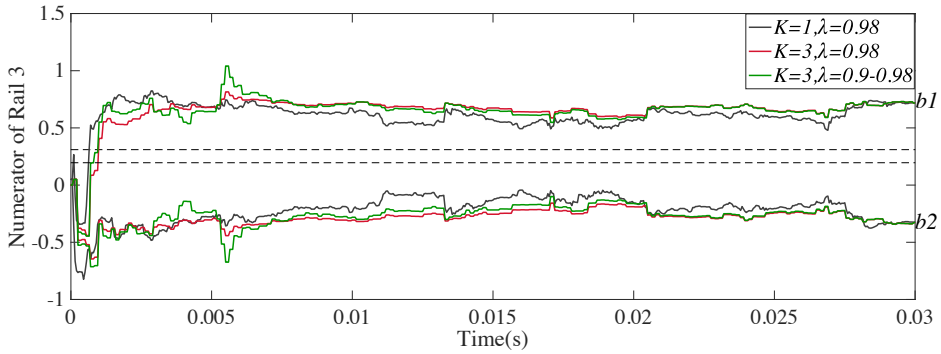
Figure 5.6 Recursive Curves (Denominators) of Three Rails with Different Iteration Frequencies Using RLS



(a)



(b)



(c)

Figure 5.7 Recursive Curves (Numerators) of Three Rails with Different Iteration Frequencies Using RLS

Figure 5.6 (denominators) and Figure 5.7 (numerators) show the convergence time and the estimation error of the coefficients of the three converter rails when RLS is applied for parameter estimation with $\lambda = 0.98$. Like simulation results, decreasing the iteration frequency 3 times leads to the convergence time increasing about 3 times in experimental results. In Rail 1 it is more than three times while in Rail 3 the two rates, acquired from the conventional way ($K = 1$) and the proposed approach ($K = 3$), are almost the same. As such, λ of each rail when $K = 3$ are respectively adjusted, after which the prolonged convergence time is shortened (see Figure 5.6). In both simulation and experiment, λ in Stage 1 is only reduced by 0.08 to 0.9. As

Figure 5.6 shows, PEs when $K = 1$ and $K = 3$ in three rails are all under ±5% of the real values (within the accuracy tolerance band), same with simulated results. Table 5-1 lists estimation error (AE), the convergence time, and variance of recursive curves when $K = 1$ and $K = 3$ of each rail. The biggest difference of the convergence time between $K = 1$ and $K = 3$ is 1.55ms in Rail 2, whereas there is barely compromise of identification speed in other rails. Besides, apart from Rail 2, the approach of reducing iteration frequency and λ features the highest estimation accuracy (see AE) and the strongest stability (see Variance). Nevertheless, the most contribution of this work is in computational cost-saving (see Table 5-2). The overall computational cost of $K = 3$ throughout Stage 1 is even less than half of it of $K = 1$. Therefore, the proposed method can be reliably applied on, and quite suitable for, online system identification of multi-rail architectures, as it achieves the computational costs on concurrently identifying multi-rails equals that on doing single rail, without noticeable compromises on other performances.

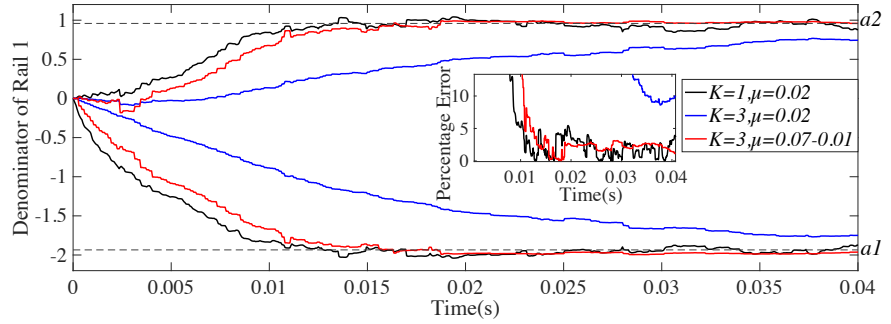
Table 5-1 Performance Comparisons of Parameter Estimation with Different Iteration Frequencies

Rails	**Iteration Frequency**	**Convergence time (ms)**	**Estimation Error (e-4)**	**Variance (e-4)**
Rail 1	$K = 1$ $(\lambda = 0.98)$	3.05	119	5.7215
	$K = 3$ $(\lambda = 0.98)$	11.45	167	11
	$K = 3$ $(\lambda = 0.9)$	3.95	102	3.3989
Rail 2	$K = 1$ $(\lambda = 0.98)$	2.3	155	9.976
	$K = 3$ $(\lambda = 0.98)$	6.85	0.577	6.6944
	$K = 3$ $(\lambda = 0.89)$	3.85	190	8.5467
Rail 3	$K = 1$ $(\lambda = 0.98)$	3	46	5.9641
	$K = 3$ $(\lambda = 0.98)$	3.75	128	2.4643
	$K = 3$ $(\lambda = 0.9)$	3	14	3.6318

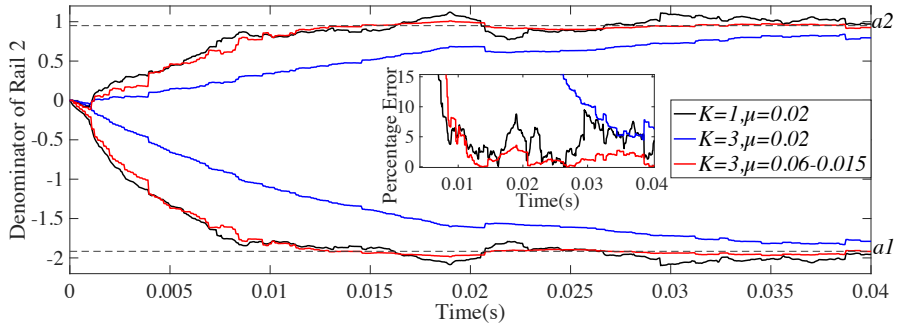
Table 5-2 Computational Cost in Total Using RLS

Iteration Frequency	**Iteration Times**	**Computational Complexity**		
		+	×	/
$K = 1$	167	10688	18203	167
$K = 3$ $(\lambda = 0.9 - 0.98)$	72	4608	7848	72

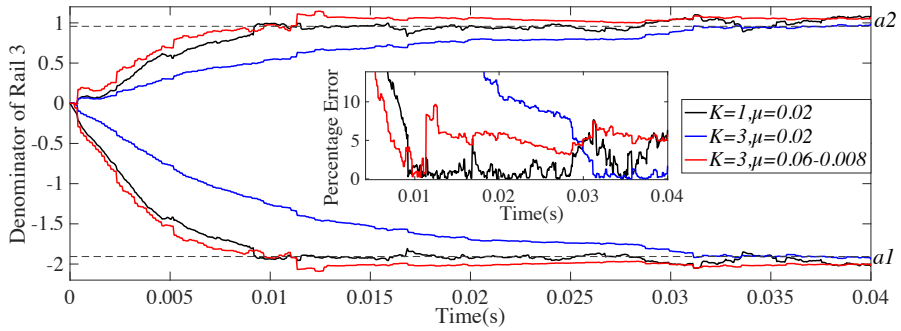
Compared with the recursive curves in simulation results, that precisely converge to the true values in all scenarios, there are consistent deflections from the true values to the estimated numerator values in the practical results. The difference between the simulated buck converters and the practical prototype is that the sampled signals from the practical ones include noises to which the parameter estimation process is very sensitive. Besides, the manufacturing or soldering defects, component tolerances, etc., may also contribute to true values acquired from SSA modelling not exactly equalling the actual ones. In addition, the frequently happened degradations of the output capacitor would influence the tap weights of the transfer functions of buck converter. This effect on the numerators are more significant than on the denominators, according to (2-30).



(a)



(b)



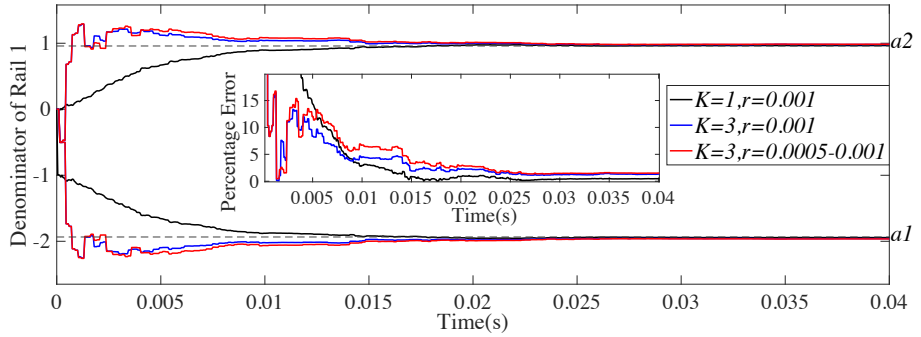
(c)

Figure 5.8 Recursive Curves (Denominators) of Three Rails with Different Iteration Frequencies Using AP

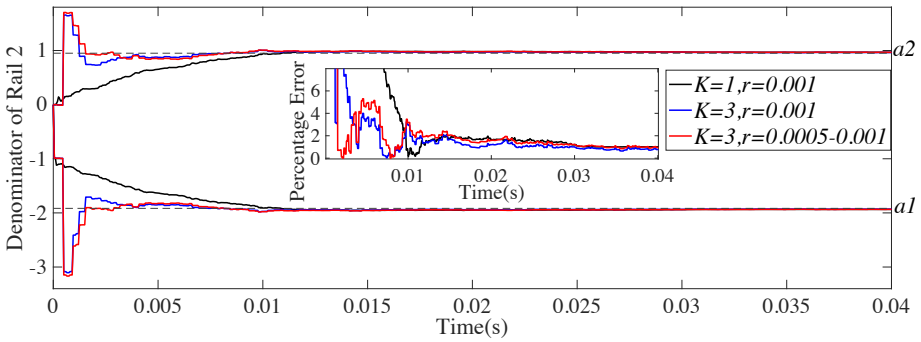
Figure 5.8 (denominators) show the estimation performances of the coefficients of the three converter rails when AP is applied with $\mu = 0.02$. As analysed in Chapter 4, under the same step size (μ) value, the iteration frequency 3 times lower than the sampling frequency results in a longer convergence time, even more than 3 times in Rail 1 and Rail 2 different from the simulation results demonstrated in Chapter 4. After increasing μ in Stage 1, the speed of the estimation curves of $K = 3$ converging to the real values is almost the same with that of $K = 1$ in all the three rails. In both simulation and experiment, μ in Stage 1 is 3 times enlarged. As Figure 5.8 shows, apart from Rail 1 in which PEs of both $K = 1$ and $K = 3$ are under ±5% of the real values, the converged curves in Rail 2 ($K = 1$) and Rail 3 ($K = 3$) are above the accuracy tolerance band. This might be caused by insufficiently filtering noises in the sampled data, although strictly filtering might attenuate the PRBS effects. According to Figure 5.8, the biggest difference of the convergence time between $K = 1$ and $K = 3$ occurs in Rail 3, whereas in all the three rails the lowered identification speed caused by the reduced iteration frequency is well-compensated. Besides, apart from Rail 3, the estimation results of reducing the iteration frequency and increasing μ feature the estimation accuracy and the stability as high as those of K equalling 1 does. The alleviation of computational costs in each sampling event has been presented in Table 4-7 in Chapter 4, which suggests that, similar with being applied on RLS, the proposed iteration decimation approach may also reduce 2/3 computational burdens when AP is used.

The estimation performances of the iteration decimation approach demonstrated on KF are shown in Figure 5.9(denominators) and Figure 5.10 (numerators). In the default setup ($K = 1$), the convergence-time-related factor (Observation Noise Variance), r, is 0.001. Unlike the results of RLS and AP which suggests that the convergence time is proportionally prolonged with the reduction in the iteration frequency, it seems the estimation speed is not significantly influenced by changing the value of r. It has been know that under the same iteration frequency, the smaller r is, the faster the estimation curves converge. However, when $K = 3$, even if r is decreased to 0.0005, the resultant curves still share the same identification speed with r equalling 0.001. As described in Chapter 4, based on the currently collected data, it is not enough to build up a conclusive figure (Figure 4.4 alike) suggesting the relation among the iteration frequency, the value of r, and the experience-based convergence time. As such, a broader and accurate data collection and discussion of the relation is necessary. Fortunately, according to Figure 5.9 and Figure 5.10 in which the convergence time and the estimation

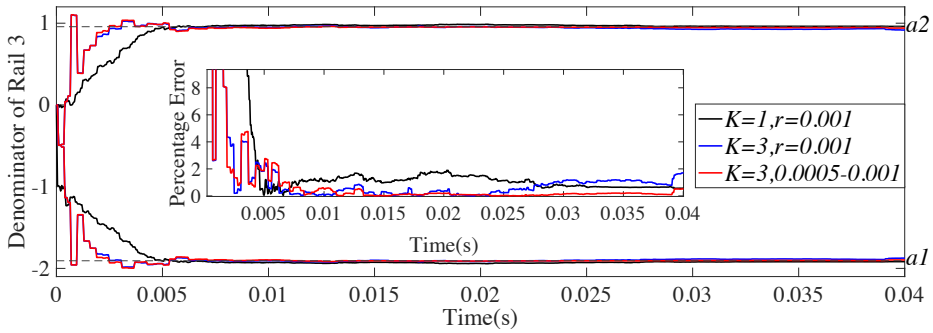
accuracy of $K = 3$ are all as same good as those of $K = 1$, it might also be reliable to apply the iteration decimation approach on KF for computational burden relief.



(a)

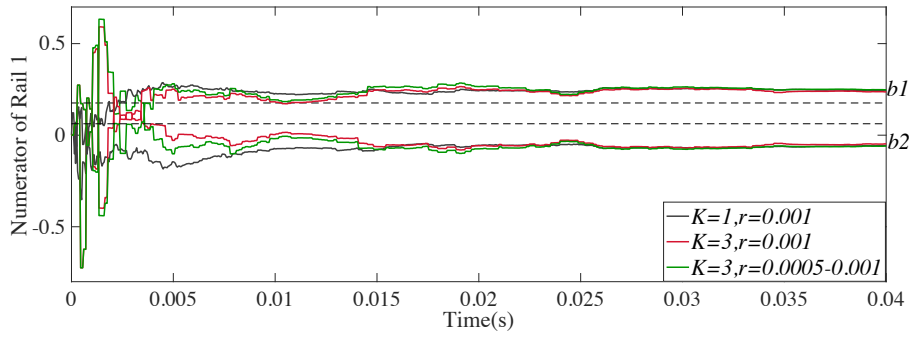


(b)

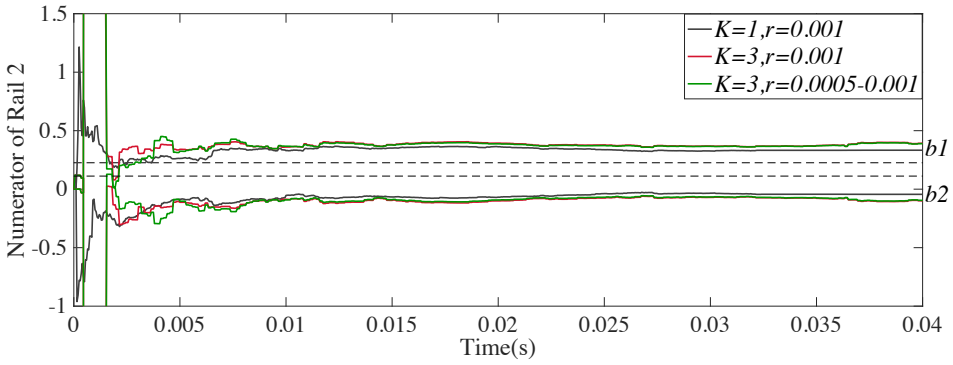


(c)

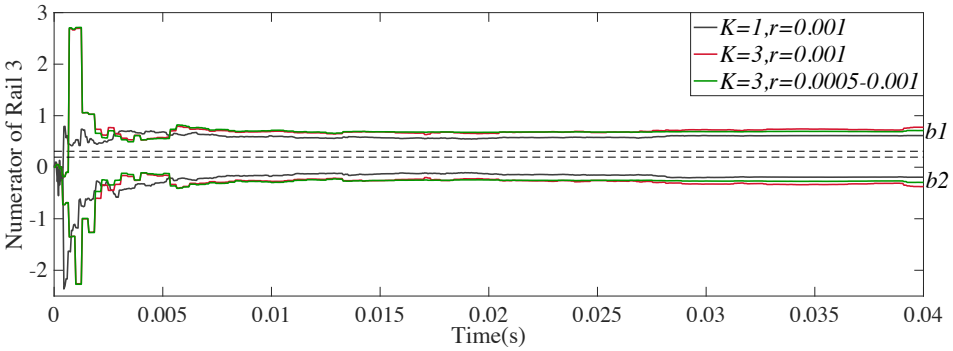
Figure 5.9 Recursive Curves (Denominators) of Three Rails with Different Iteration Frequencies Using KF



(a)



(b)



(c)

Figure 5.10 Recursive Curves (Numerators) of Three Rails with Different Iteration Frequencies Using KF

In summary, the iteration frequency reduction approach is experimentally validated to be able to implement on RLS, AP and KF for computational burden alleviation in real-time parameter estimation of multi-rail power converters. The experimental results indicate that though adjusting the convergence-time-related factors of the adaptive filters in different estimation stages, the estimation performances, the convergence time particularly, would not be negatively affected by the reduction of the iteration frequency. Meanwhile, the computational burdens would be significantly relieved both in each sampling event and in total.

5.4.2 The CMA/CT Reusing Approach

The iteration event distribution of each rail is the same with that used for simulation validation in Chapter 4 (see Figure 4.11 and Figure 4.15). RLS is firstly applied with the CMA reusing approach. Figure 5.11 (denominators) and Figure 5.12 (numerators) show the estimation curves of the four coefficients of each rail under the two CMA reusing scenarios ($K = 3/2$ and $Q = 3$), of which performances are compared in Table 5-3 in terms of the convergence time and the estimation accuracy. In both scenarios, the sequence is benchmarked against the default of $Q = 1$. In simulation results (in the ideal conditions without any noises),

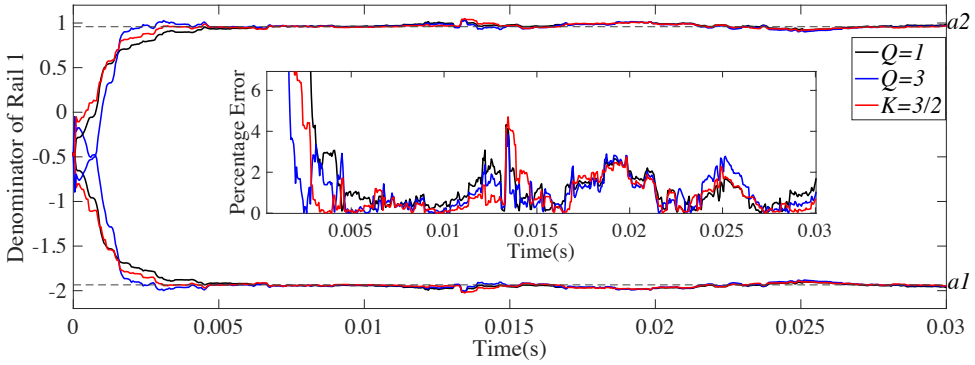
the convergence time of $K = 3/2$ should be compensated by decreasing the value of λ in Stage 1 as its iteration frequency is 2/3 times lower than the sampling frequency. However, as the estimation speed is not significantly lowered in practice, all results are still acquired with the same λ equaling 0.98. For all the three rails, there is no noticeable difference of convergence time in Q equaling 1 and 3 and of K equalling 3/2. As shown in Figure 5.11, PEs at $Q = 1$, $K = 3/2$ and $Q = 3$ of each rail are all in the same level (within the accuracy tolerance band). Besides, AE or Variances, all in the same level, implies that in practical work, in the presence of noises, the differences of CMA in two, or even in three consecutive iterations would be small enough for reusing CMA. The total computational costs of each scenario are also compared in Table 5-4, which suggests that the proposed approaches may alleviate more than half of the computational burdens of the conventional way.

Table 5-3 Estimation Performance Comparison of Reusing CMA of RLS

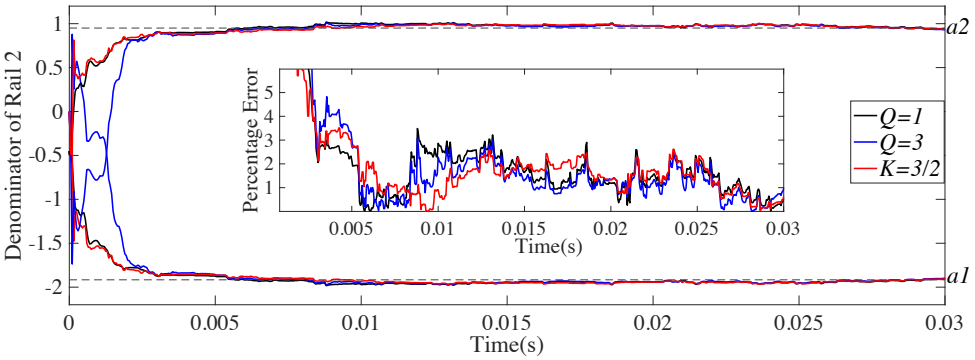
Rails	**Iteration Strategies**	**Convergence Time (ms)**	**Estimation Error (e-4)**	**Variance (e-4)**
Rail 1	$Q = 1$	2.95	97	6.1205
	$Q = 3$	1.75	68	5.5957
	$K = 3/2$	2.45	60	5.4385
Rail 2	$Q = 1$	2.7	180	7.7182
	$Q = 3$	2.8	92	9.5067
	$K = 3/2$	2.75	104	9.8121
Rail 3	$Q = 1$	2.2	46	5.818
	$Q = 3$	2.5	4.8968	5.3967
	$K = 3/2$	2.5	22	4.9184

Table 5-4 Computational Cost in Total Using RLS

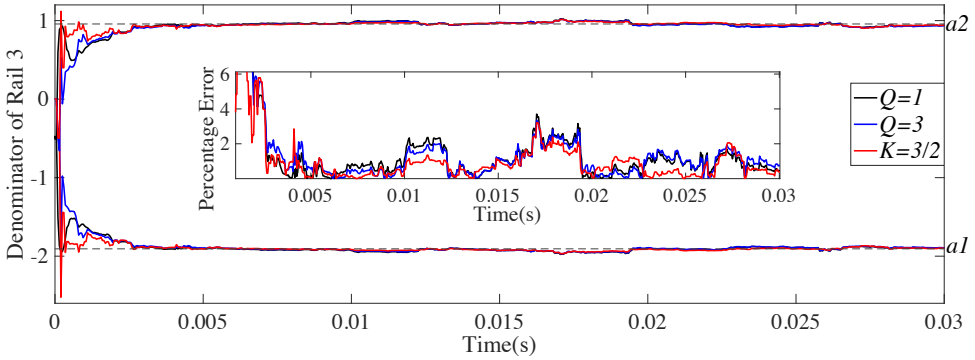
Iteration Frequency	**Iteration Times in Total**	**Computational Complexity**		
		+	×	/
$Q = 1$	157 'whole'	10048	17113	157
$Q = 3$	45'whole', 96'half'	4800	7209	45
$K = 3/2$	53'whole', 51'half'	4412	7001	53



(a)

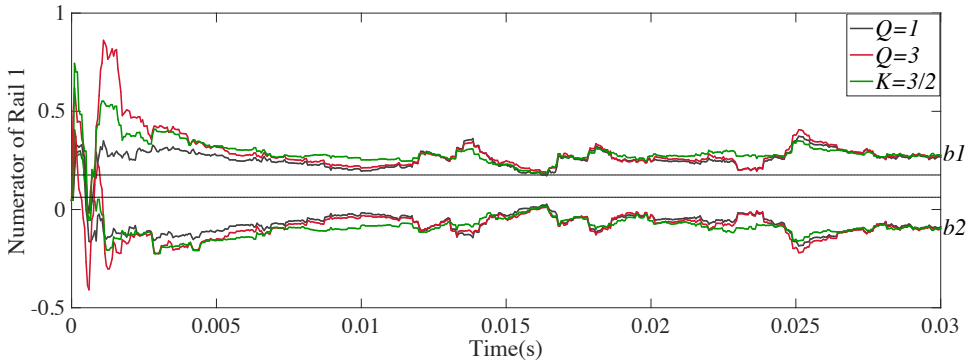


(b)

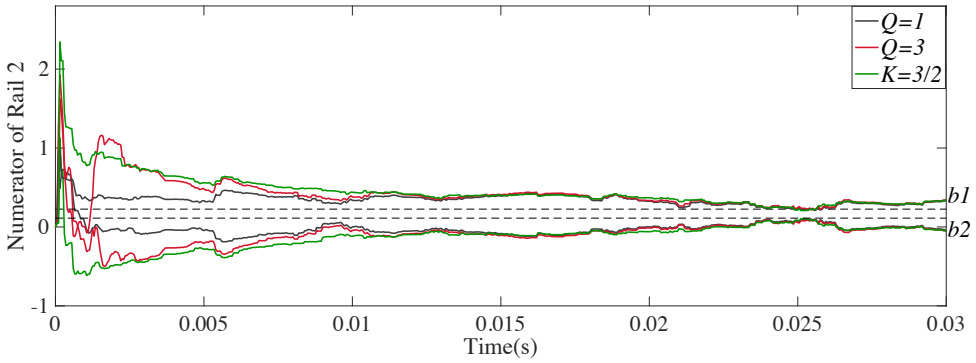


(c)

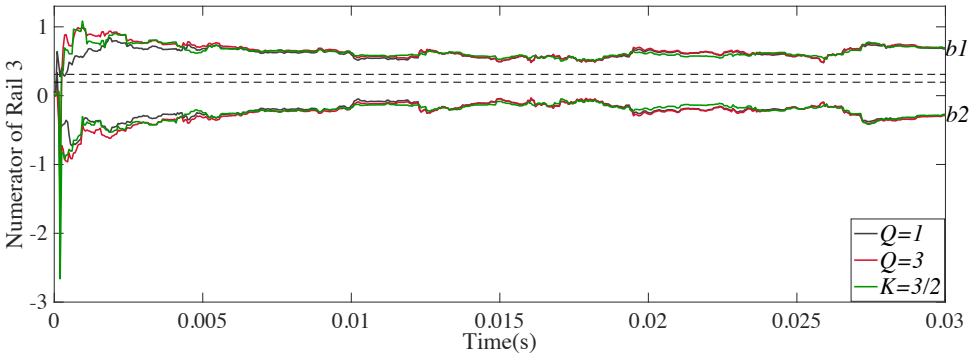
Figure 5.11 Recursive Curves (Denominators) of Reusing CMA in Three-Rail Architecture Using RLS



(a)



(b)

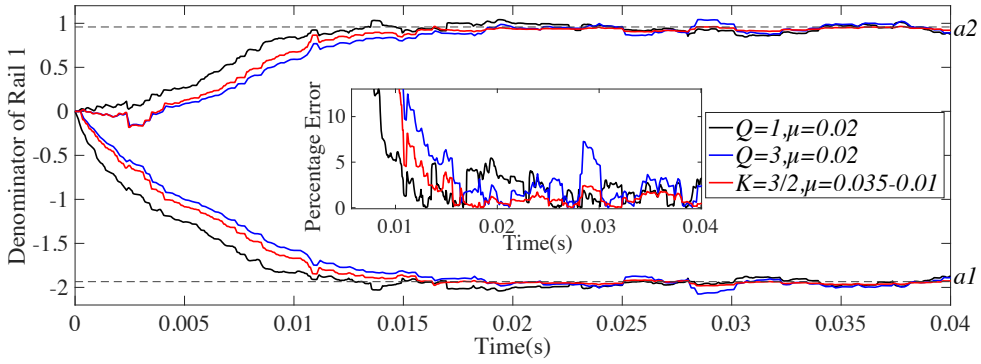


(c)

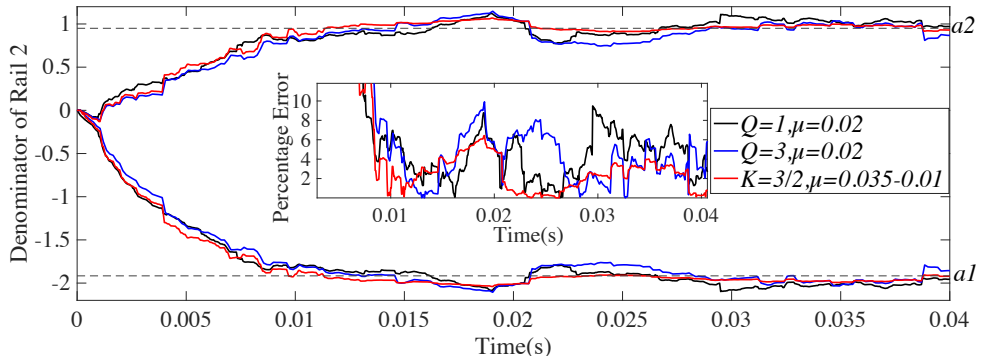
Figure 5.12 Recursive Curves (Numerators) of Reusing CMA in Three-Rail Architecture Using RLS

Figure 5.13 show the estimated parameter curves of each rail using AP under the three scenarios: $Q = 1$ (the conventional way), $K = 3/2$ (reusing CT once) and $Q = 3$ (reusing CT twice). Different from the convergence-time-related factor adjustment in RLS, for the case of $K = 3/2$, the step size (μ) of AP is increased from 0.02 to 0.035 to speed the prolonged convergence process up. As a result, although there are still slight differences in the convergence time of the three cases in Rail 1, the recursive curves of the other two rails enter the accuracy tolerance band at the same time. As mentioned before, the filtered data might not be smooth enough for AP so that the corresponding validation results suffer the severer fluctuations than those of using RLS. μ in Stage 2 of the two reusing CT scenarios, therefore, is decreased to 0.01 in all three rails. As a consequence, the estimation errors of $Q = 1$ and $K = 3/2$ can be in the same level, as well as their stability (see Figure 5.13). However, the estimation accuracy of $Q = 3$ is still not as high as that of the other two cases. Reusing CT twice may enlarge the noise effects which leads to the recursive curves that have been converged exceed the accuracy tolerance band ((±5% of real values). Nevertheless, as there is no significant compromise on estimation performances when the CT reusing approach is applied on AP, it

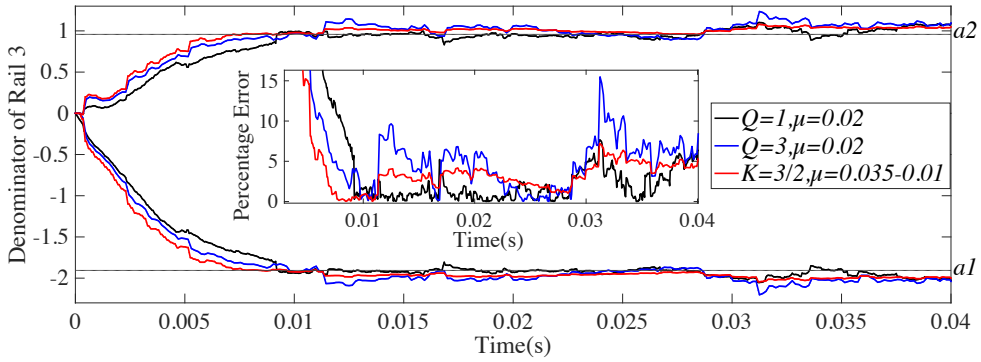
still can be considered as being reliable to be employed on AP. After all, according to the total computational cost comparison of each scenario listed in Table 4-7 in Chapter 4, reusing CT may alleviate almost two third of the computational burdens of the conventional way.



(a)

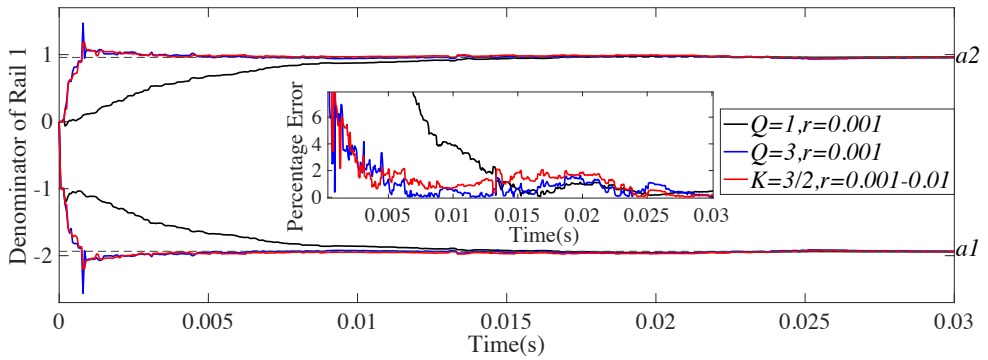


(b)

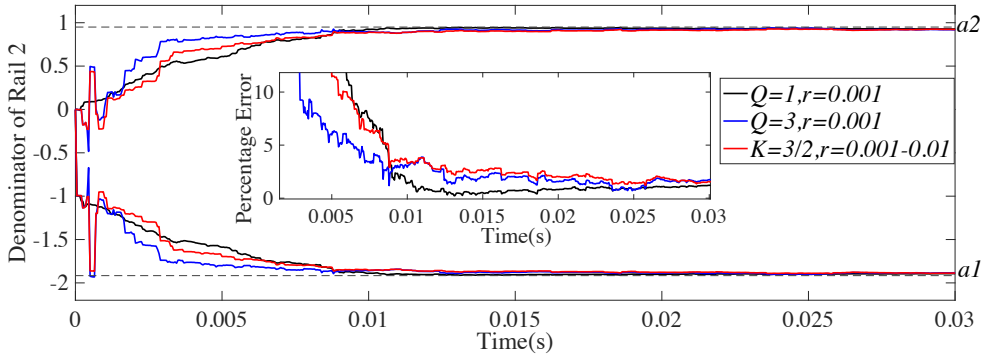


(c)

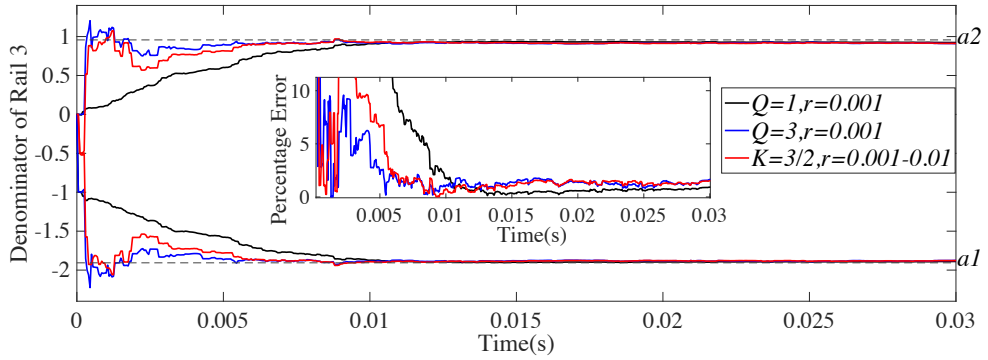
Figure 5.13 Recursive Curves (Denominators) of Reusing CMA in Three-Rail Architecture Using AP



(a)

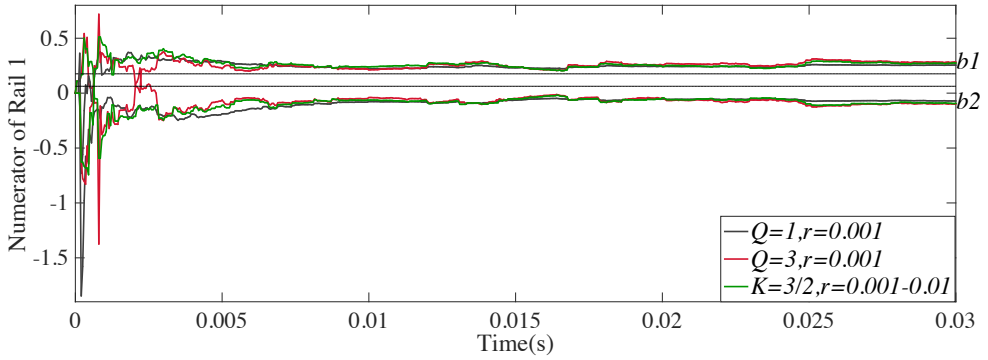


(b)

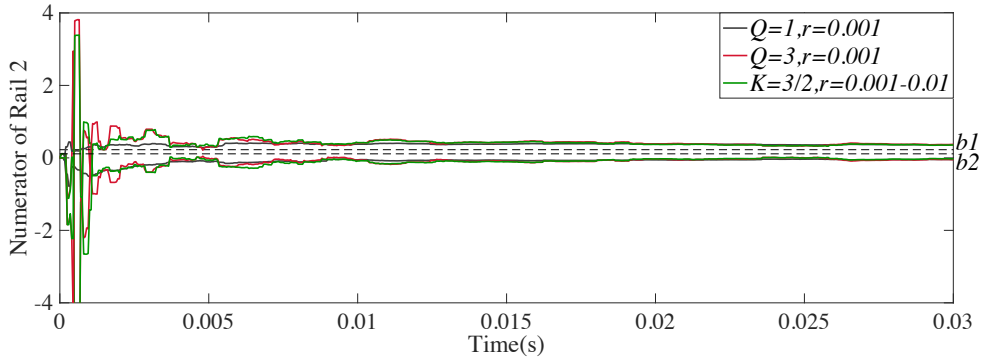


(c)

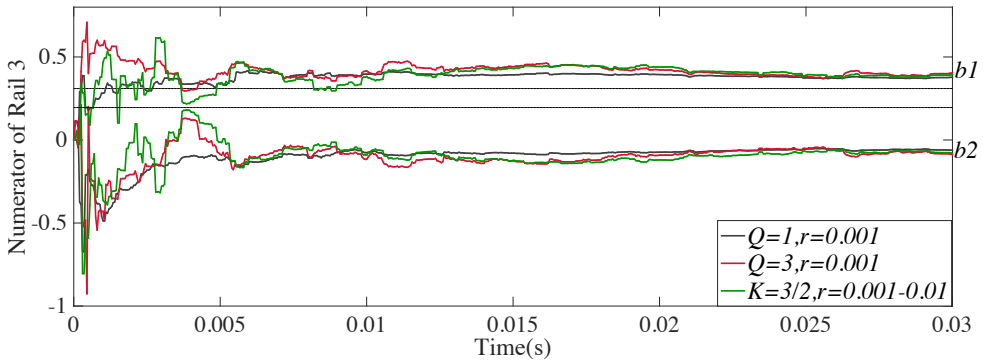
Figure 5.14 Recursive Curves (Denominators) of Reusing CMA in Three-Rail Architecture Using KF



(a)



(b)



(c)

Figure 5.15 Recursive Curves (Numerators) of Reusing CMA in Three-Rail Architecture Using KF

The realization of the CMA reusing approach on KF applies the same iteration distribution arrangements used on RLS. Figure 5.14 and Figure 5.15 shows the recursive curves of the estimated parameters of the three rails, which indicates the convergence time of reusing CMA once and twice in Rail 1 and 3 is even shorter than that of the conventional iteration way. The reused CMA, calculated from the last iteration cycle, may cause the CT overcorrecting the new guess in the current iteration cycle, then the next CMA, consequently, will keep being overcorrecting. As a result, the convergence time might be shortened but there are severer fluctuations in recursive curves. Nevertheless, all recursive curves still converged into the accuracy tolerance band ($\pm$5% of the true values). Table 4-7 in Chapter 4 shows that applying the CMA reusing approach on KF may save more than half of the computational cost of using the conventional way, as this approach being verified on RLS and AP does. These qualified performances demonstrate that CMA reusing approach cannot only be employed on RLS but also on KF for computational burden alleviation.

5.4.3 Abrupt Load Change Rejection

To validate the disturbance rejection ability of locally reducing the iteration frequency, load changes are configured in Rail 1 (output current changes from 0.36 A to 1.8 A) and Rail 2 (from 0.66 A to 3.3 A) at 10ms, and in Rail 3 (from 0.5 A to 2 A) at 15ms. The coefficients that should be identified of each rail after load changes are the same with those in simulation setups (see Table 4-10 in Chapter 4). Figure 5.16 shows the transient response of the output voltage from Rail 2 coping with an abrupt load change during being estimated (PRBS is being injected), same with those in the other two rails. 600 data including transient responses of output voltage coping with abrupt load change are sampled and estimated. For the space limit, only one parameter curve, a_1, is presented to show the effects of load changes on estimation curves.

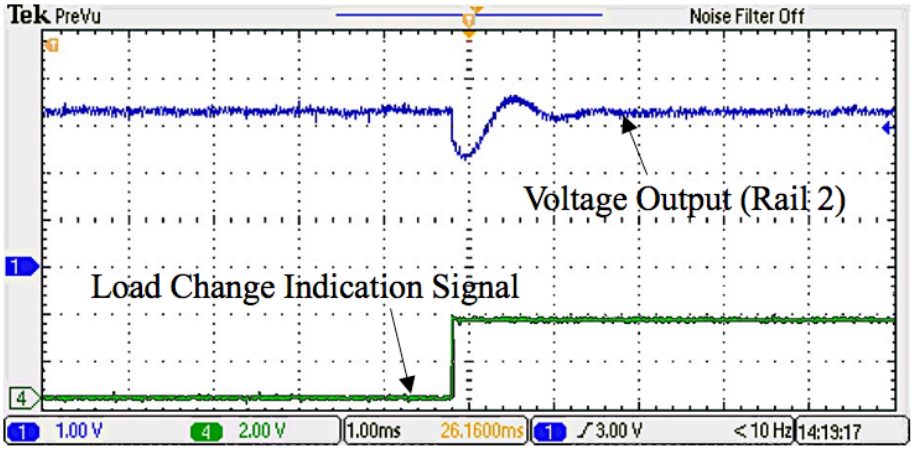
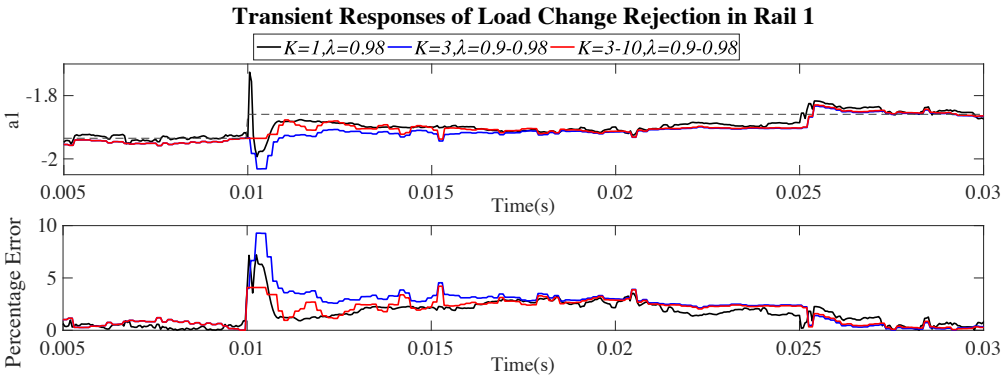


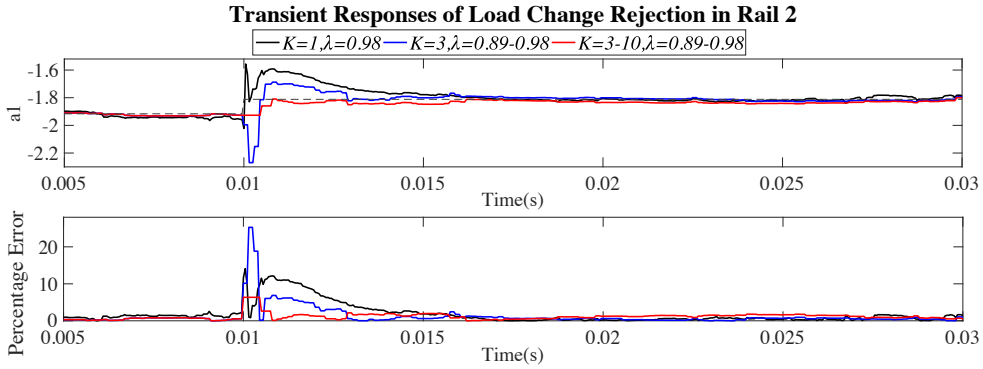
Figure 5.16 The Transient Response of the Output Voltage from Rail 2

During oscillations (transient behaviours dealing with load changes), for performance comparison of disturbance rejection, the iteration frequency is differently configured. Figure 5.17 shows the transient convergence behaviours for new result updates of the iteration frequency equalling the sampling frequency ($K = 1$), of lowering the iteration frequency throughout the entire estimation process ($K = 3$) and of locally and temporally lowering the iteration frequency ($K = 10$) only for the 0.5ms in which oscillations exist. Accordingly, the recursive curves of $K = 10$ features the fastest update speed and barely contain fluctuations, same as simulated results. At $K = 1$ and $K = 3$, however, oscillations of the sampled data, taken into algorithm recursion account, result in severer fluctuations of recursive curves. Meanwhile, as the convergence time also depends on the performance of the employed algorithm, the reduced iteration frequency, (or the lowered iteration density), of $K = 3$ prolongs the convergence time consequently. However, at $K = 10$, removing sampled

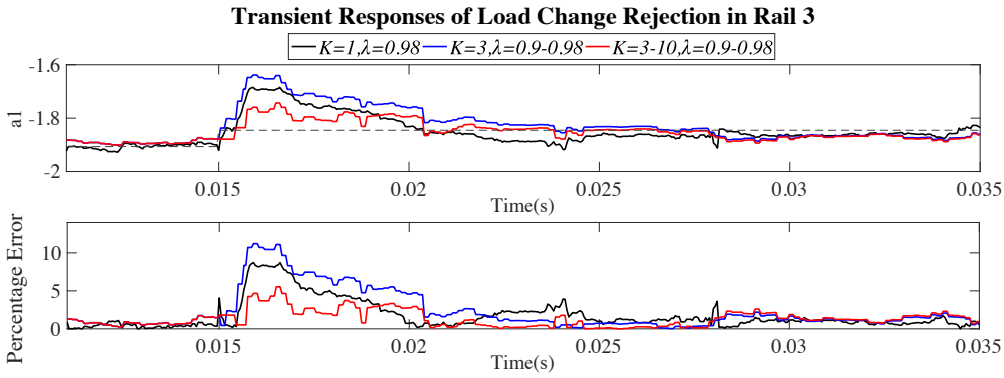
oscillations out from iterations avoids noticeable fluctuations in recursive curves, and consequently, there is no need to consume time on coping with oscillations.



(a)

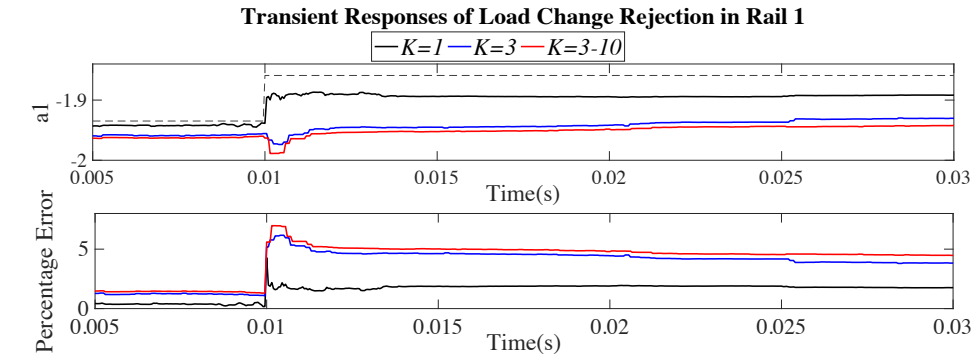


(b)

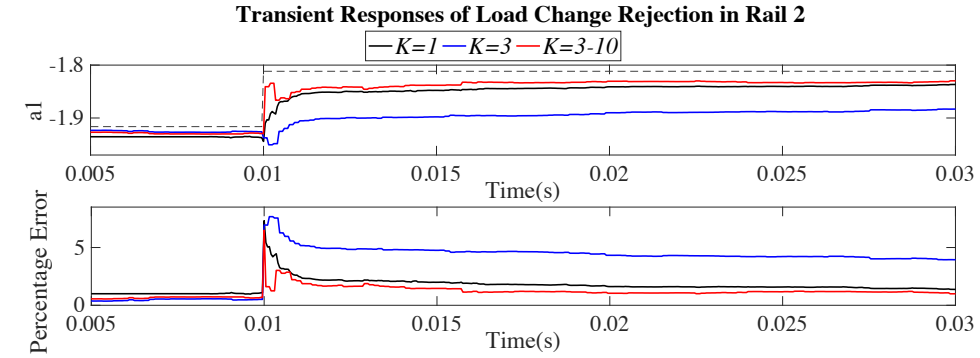


(c)

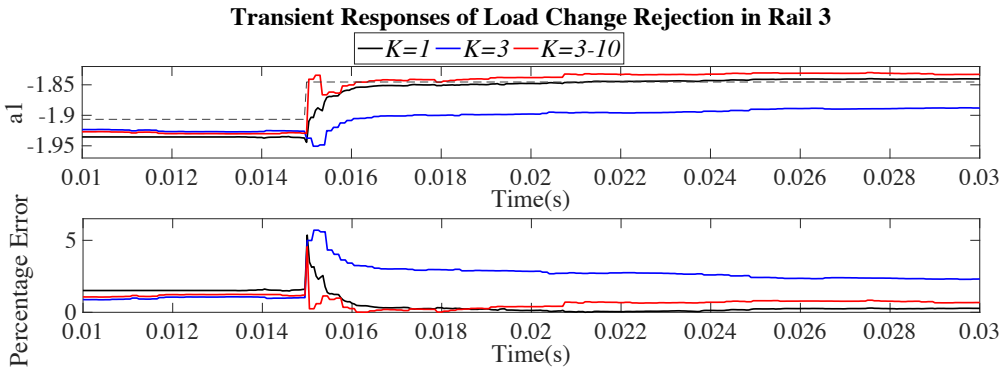
Figure 5.17 Recursive Curves Comparisons between Locally Disposing Disturbance Signals and Fixed Iteration Frequency Approaches (RLS)



(a)



(b)

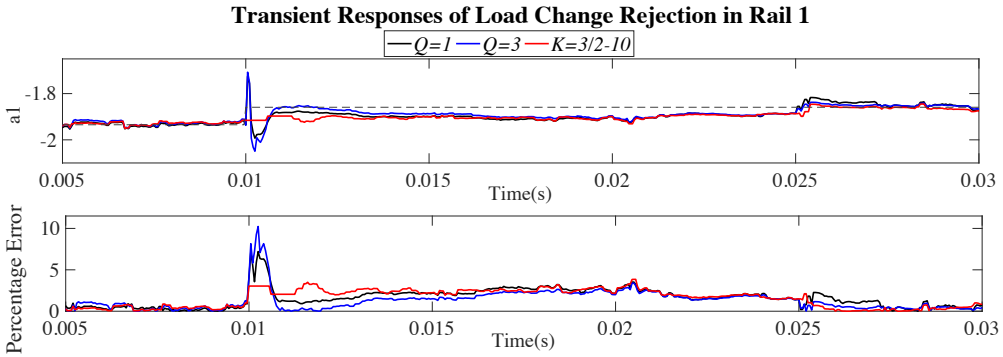


(c)

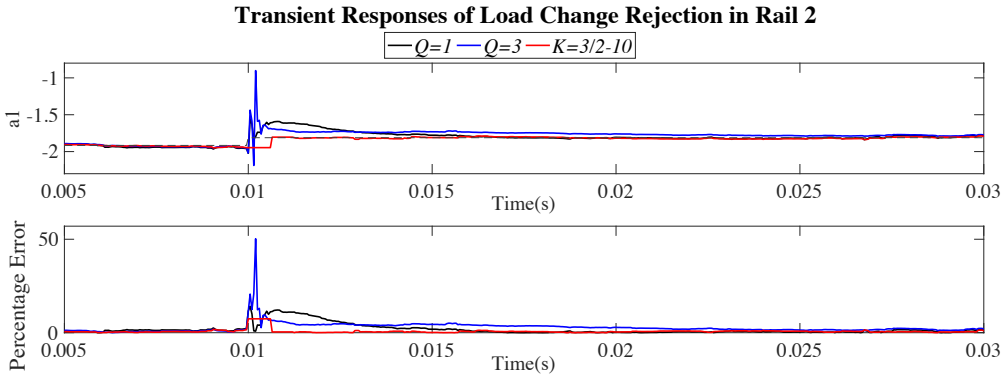
Figure 5.18 Recursive Curves Comparisons between Locally Disposing Disturbance Signals and Fixed Iteration Frequency Approaches (KF)

The estimation curves of KF rejecting abrupt load changes are collected in Figure 5.18. The load change setups in the three rails for KF validation here are the same as those for RLS validation. Figure 5.18 shows the transient convergence behaviours for new result updates of three iteration processes. In Rail 1, there is a deflection in both the updated recursive curves from the two reduced iteration frequency cases, although their percentage errors are still in the tolerance band. In Rail 2 and Rail 3, the recursive curves of locally and temporally lowering the iteration frequency ($K = 10$) may update the new results as accurate as those of $K = 1$. In all rails the estimation curves of $K = 3$ fail to accurately converge to the new real values after

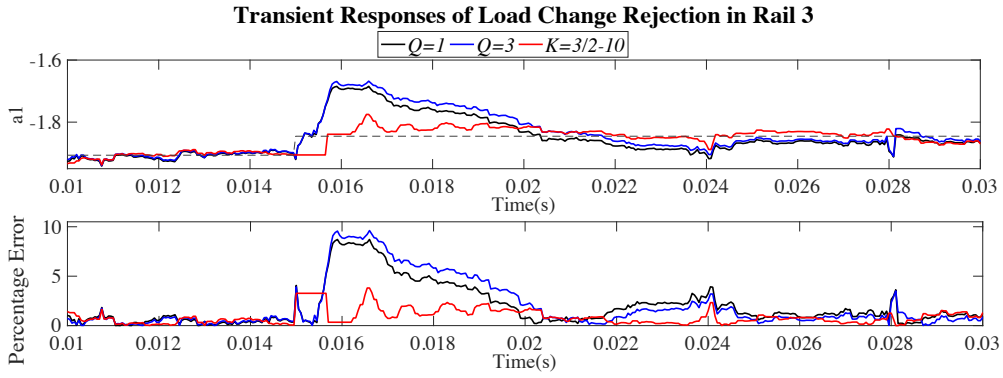
load changes. the unfiltered transient responses in the sampled data influences/deflects the updated results and the reduced iteration times makes the recursive curves not being able to converge to the true values any more.



(a)



(b)



(c)

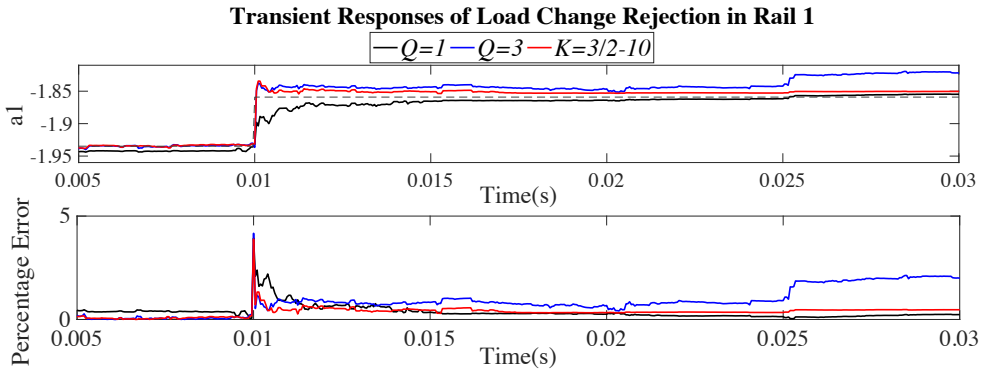
Figure 5.19 Recursive Curve Comparisons between Locally Disposing Disturbance Signals and Fixed Iteration Frequency Approaches (RLS)

Figure 5.19 shows the convergence behaviours of respectively using the conventional way ($Q = 1$), the CMA reusing approach ($Q = 3$) and locally lowering iteration frequency approach ($K = 3/2 - 10$). Here, the iteration frequency is significantly lowered only for the 0.5ms where oscillations exist. It turns out that the experimental results are almost the same as simulation

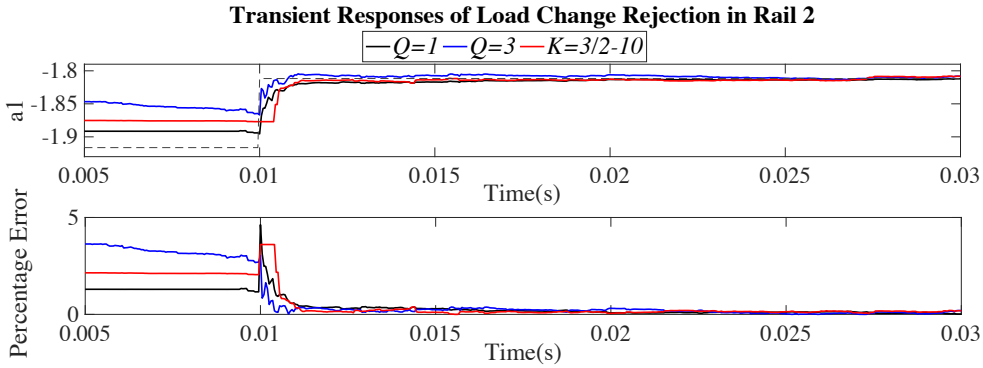
ones. In $Q = 3$, when the sampled transient behaviour of rejecting abrupt system changes is considered in recursive algorithms, two CMAs respectively calculated in the last iteration and the next, next one, are hugely different, which means CMA gotten from the last might not be a suitable replacement for the current or the next iteration events. Even worse, these inappropriate substitutions could bring more fluctuations in recursive curves and then prolong the estimation duration, demonstrated in Rail 2 (see Figure 5.19(b)).

As Figure 5.19 shows, the proposed approach of $K = 10$ performs best for disturbance rejection. However, K equaling 10 may be only suitable in this case as the value selection of K for transient response removing depends on the disturbance severity and the sampling frequency. A severer disturbance may cause a transient response with a larger overshoot and a longer settling time, K, therefore, should be adjusted larger. A high sampling frequency may lead to more transient response signals being sampled, which will be removed, K, therefore, should be adjusted larger. For a broader discussion of the selection of K, the magnitude of control error signals can be tracked to investigate the severity of variations and disturbances suffered by systems. Then a larger magnitude of the error signal may decide a larger K to guarantee the sampled transient responses being removed. Better than simulation, in practical work, only the recursive curves of Rail 2 [in Figure 5.19(b)] are slightly excess the accuracy tolerance band (±5% of real values) in rejecting disruptions, which suggests the proposed approach may provide acceptably accurate results throughout estimation process even under disruption occurrences.

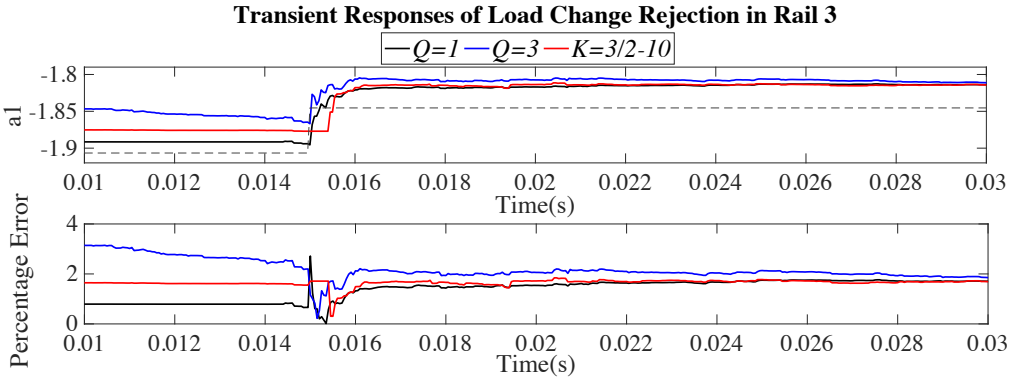
The estimation curves of KF rejecting abrupt load changes are collected in Figure 5.20. The load change setups in the three rails for KF validation here are the same as those for RLS validation. Here, the CMA of KF is reused. Different from the proposed approach being applied in RLS, there is no need to remove transient responses of the sampled signals for KF. Although all of the recursive curves after load changes are still in the accuracy tolerance band, that of reusing CMA twice ($Q = 3$) after load changes suffer severer fluctuations, which might suggest that CMA calculated in the last iteration cycle may be an inappropriate substitution for the next one. To reduce computational costs with high estimation accuracy, CMA is preferred to be reused only once.



(a)



(b)



(c)

Figure 5.20 Recursive Curves of the CMA Reusing Approach in Dealing with Load Changes (KF)

5.5 Chapter Summary

The chapter presents the experimental validation of the iteration decimation approach and the CMA/CT reusing approach introduced in Chapter 4. Through conducting a complete real-time parameter estimation platform, the two proposed approaches are implemented and verified on three algorithms (RLS, AP and KF). The experimental results of the two computational cost-saving approaches are in very good accordance with the simulation results presented in Chapter 4, suggesting the applicability of the two approaches in a broad kind of adaptive filters. Besides,

the estimation performances, in terms of the convergence time, estimation accuracy and stability, of employing the approaches on algorithms are compared with the case of not using them. The comparison validates that the approaches could save two third of the computational burden suffered in the conventional cases, and their results are as reliable as the conventional way, or even features the higher stability. In addition, the most moderate recursive curves, as the results of the iteration decimation approach, during abrupt load change rejection, indicate that the proposed transient response removing way performs well in coping with sudden system variations.

Chapter 6 Prioritization of System Identification Tasks in Multi-Rail Architectures

6.1 Introduction

To cope with simultaneously-occurred events, such as requests, tasks and problems, etc., under resource limit, ranking these events is unavoidable. According to the importance, urgency and severity, etc. of events, they are separately labelled with a unique priority which with higher priority will be dealt with sooner. As to system identification of multi-rail power conversion architecture, it is SI (System Identification) request from each rail that should be ranked due to the likelihood of multiple rails sending SI requests at the same time. In adaptive control loop, it typically takes a few milliseconds to finish real time SI. Given that the maximised capability of simultaneous parameter estimation if only 3, validated in **Chapter 4**, when the number of rails sending SI request is above 3, rails with low priorities should wait until ones with higher priorities finish. Based on recent researches about prioritization and characteristics of this case, the requirements of prioritization approaches can be concluded as follows:

Firstly, many respects should be considered during SI request ranking, such as application of each rail, severity of system variation and the possible frequency of rails suffering from sudden changes, etc. In application respect, the rail providing the regulated voltage to memories is often operationally superior to those for monitors which would, therefore, be assigned with higher priority. As to the likelihood of system changes, rails containing more frequent or periodic changes (in load for instance) should be paid more attention than those rarely suffering variations. Otherwise, when two rails are encountering disturbances at the same time, the one with larger magnitude of its error signal, indicating a severer system change, should be identified first. Consequently, for reasonable prioritization, respects taken into consideration should be comprehensive.

Secondly, as the severity of system changes which trigger SI requests, user's needs and even the number of rails, etc. would vary during converter operation, a dynamic scheduling strategy, which allows priorities of rails flexibly updating, is needed rather than a pre-determined ranking way unchangeable.

Thirdly, other general requirements for prioritization include minimisation of scheduling cost and make span of tasks (the average duration from SI request starts to SI completion should be as short as possible).

To the above-listed requirements, this paper proposed a ranking approach which features that: 1. the frequency of updating ranking results equals the iteration frequency of recursive

algorithms applied for SI, to match the requirement of being dynamic. 2. in terms of being quantitated with weights, multiple respects are able to be readily considered into ranking rails. 3. SI process is divided into 'interruptible' stage and 'uninterruptible' stage to avoid rails with low priorities waiting unreasonably long time which could prolong the average pending time of all rails. 4. the intermediate results, corresponding to an interrupted SI process, would be saved instead of be thrown away, for being continuously used once the SI process recovers, to save prioritization cost.

6.2 The Proposed Workflow

6.2.1 Features

As described in [235], a request management framework can be divided into two parts: ranking rails and allocating tasks into the selected rails. The proposed variable iteration frequency approach in Chapter 4 have validated that the capability of simultaneous SI of multi-rail power converters is three. only three rails can be identified at the same time. As such, designing ranking scheme for system identification of multi-rail systems is under a limited resource condition.

This prioritization approach is featured of being dynamic, selectively pre-emptive, cost saving and multi-respects:

1. Dynamic: Firstly, control error signals of rails are monitored at the sampling rate. If system variations which could trigger SI requests occur within more than three rails, the severity of system variations will be considered for prioritization by comparing magnitude of errors of rails in real time. Secondly, the proposed prioritization approach is programmable and editable even if SI requests have been in the pending queue. Thirdly, there are various user-defined configurations (see in **6.2.2A**) that may match individual performance requirements.

2. Selectively Pre-emptive for cost-saving and make-span minimization: In conventional pre-emptive scheduling, the high-priority tasks can be coped with first by interrupting the low-priority ones. In this approach SI process is divided into two stages, pre-emptive way only works in Stage 1. The SI processes in Stage 2 are uninterruptable even if there are the higher priority tasks waiting for SI. This way may avoid almost finished SI processes being interrupted by the higher priority requests. If they do not stop, estimation results could be updated soon. If interrupted, however, the corresponding rails might wait ages for re-get SI opportunity in rush hours to start over. Nevertheless, the time ratio between Stage 1 and 2 should be carefully configured. The longer the Stage 2 is, the more similar to FCFS the approach is. In opposite, a longer Stage 1 would result in the scheduling way performing more 'pre-emptive' alike. (If

Stage 2 equals the whole period of SI, the proposed approach becomes FCFS, while if Stage 1 equals the whole SI period of SI, the approach becomes pre-emptive based on each rails' urgency coefficient).

3. Cost-saving: Firstly, only if the number of SI requests are above three, the module of severity comparison of system changes will work for ranking. Otherwise, the module is idle for reduce computational efforts. Secondly, conventional pre-emptive approaches may lead to some rails, of which SI were interrupted, throwing away uncompleted iterative results from the previously-interrupted SI processes when they re-get chance of SI. To avoid it, in this approach, these intermediate iteration results will be saved so that the last interrupted SI processes may continue by recovering these results, unless new system variations occur during interruption periods (the intermediate iteration results will be disposed and SI process start over in this case).

4. Multi-Respects: There should be various respects (two essential ones in this paper) considered in ranking SI requests. In each respect a ranking queue will be produced and in this paper the number of ranking queues is two consequently. One of the two queues is default ranking queue in application respect and the other one based on the severity of system variations. In application respect, rails with different importance in applications are featured of different priorities. For example, rails providing the regulated voltages to core processors are more functionally important than those for monitors. The default setup of priority is that the smaller a rail number is, the higher priority the rail has. In variation severity respect, the rails suffering a severer system variation, expressed as a higher magnitude of control error signals, could gain a higher priority as they should be identified sooner.

6.2.2 Initialization, Tracked Signals and Flowcharts (Nassi-Shneiderman Diagram) of the Proposed Workflow

A. Initial Setups

As of the afore-described sequence and allocation way, before carrying out the workflow, there are some initial setups before taking place the workflow:

1. Rail Number (n in flowcharts from Figure 6.1 to 6.4): the number of rails in a multi-rail power conversion architecture, used to define Priority Coefficient Decrement and the matrix sizes of Priority Coefficient Matrix (PCM) and Urgency Coefficient Vector (UCV).

2. Priority Coefficient Decrement (see ***Decrement*** in Figure 6.3): A constant that equals $1/n$, used to calculate PCM.

3. Number of Consideration Respects [m in (6-2)-(6-5)]: the number of respects considered for prioritization, used to define the size of Importance Coefficient Vector [see (6-3)].

4. Importance Ratio of Consideration Respects [r_q in (6-2)]: the criticality degree of each consideration respect, used to calculate percentage Importance Coefficients [b_q in (6-2)].

5. Threshold 1: A value, above which the corresponding rails should be identified, is configured based on experienced error signal magnitude during noticeable system variations.

6. Threshold 2: When a particular number of error signals of a rail, sampled in the most recent iterations, are lower than Threshold 2, the rail has been in steady state and ready for system identification. This value is just slightly higher than control error signal under PRBS effects.

7. Sampled Numbers of Checking Steady State: The number of error signal values, sampled and recorded in the most recent iterations, used to check if rails with system variations have been in Steady State.

8. Steady State Indication Threshold: The proposed prioritization system would count the number of recently sampled error signals lower than Threshold 2. When the number is larger than Steady State Indication Threshold, the corresponding rail has been in steady state.

9. Nominal Voltage Outputs: Used to calculate percentage error magnitudes.

10. Time Length of Stage 1 and 2: Stage 1 is the period, in terms of the assigned times of iteration, during which SI of the corresponding rail can be interrupted by others with higher priority, while in Stage 2 the task of the rail is unstoppable even if other rails have larger Urgency Coefficients. The longer the Stage 1 is, the workflow performs more pre-emptive alike. Otherwise, it is more similar to FCFS. Besides, the SI duration (N_{si} in Figure 6.2) is the sum of Stage 1 and Stage 2.

B. Tracked signals

1. e: Control Error Signal, sampled for detection of system variations.

2. $pcErr$: Percentage form of e, calculated for severity evaluation of system variations.

3. eR: the part of $pcErr$ above Threshold 1, recorded to trigger the error-magnitude-based prioritization process.

4. EG: the highest value of eR, held to indicate and compare the severities of system variations of rails.

5. $Noss$: the number of error signals, sampled in the most recent iterations, below Threshold 2.

6. Steady State Indicator (SSI): a signal indicating the corresponding rail is/has been ready for SI. If an abrupt change occurs in a rail being identified, the rail would still be ‘ready for SI’,

while when it occurs in rails not being identified, the rails would be viewed as 'being ready' until they operate in steady state.

7. App: Application coefficients

8. $NoErr$: the number of rails which need to be/is under SI

9. $Error$: Priority coefficient vector, acquired based on the system variation severities of rails and listed from Rail 1 to Rail n.

10. $Ordermag$: Urgency Coefficients of rails [see (6-5)] that SI task allocation is based on.

11. q: the vector of rail numbers listed from the one with the highest urgency coefficient to the lowest.

12. EI: Waiting & Processing indicator, its 'On' state starts from the SI request of a rail produced and ends when the SI process finishes.

13. $NoSI$: the number of rails being identified. its highest value is 3.

14. SI: System Identification demand signal, the SI task will be allocated to the rail of which SI is on.

15. SI process Counter (I): the iteration Counter of a SI process.

16. $Reset$: when I equals N_{si}(the assigned iteration times for a SI process), the SI process finishes and there will be an impulse signal for resetting.

C. Flowchart (Nassi-Shneiderman Diagram) of One Prioritization Cycle

The workflow of prioritization includes four sections (the first three are about Ranking rails and the last one task allocations): 1. the maximum error value collection, 2. error-respect ranking decision, 3. ranking rails based on Urgency Coefficient calculation, 4. allocating system identification procedure to rails with high priorities. Due to the space limit, the Nassi-Shneiderman diagram, as an alternate of flowcharts, is used to present the proposed prioritization approach.

1. The maximum error value collection: The diagram in Figure 6.1 and Figure 6.2 specifically demonstrate the procedure of this section. Control error signals (e) of each control loop of each rail are monitored (STEP 1 in Figure 6.1) and the corresponding percentage error magnitudes ($pcErr$) are calculated (STEP 2 in Figure 6.1). Once the $pcErr$ is above an experience-based and predefined threshold (Threshold 1 in STEP 3 in Figure 6.1), the maximum value of percentage error (EG) will be recorded and kept for comparison with those of other rails (STEP 4a and STEP 5 in Figure 6.1). In STEP 5a, if Reset is 1, indicating the completion of the SI process of the corresponding rail, the maximum percentage error signal would not be recorded which means the SI request has been coped with (see STEP 6b in Figure

6.1). STEP 5b checks whether the rail being examined is being identified. If the SI process has entered Stage 2, it would start from the beginning of Stage 2 due to the error caused during SI (see STEP 7), while it is unnecessary to start over if the process is in Stage 1. If the rail is not being identified, the SI process Counter (I) should be reset to 1 (see STEP 6d).

The rail with larger percentage error will be assigned with higher priority for being quicker identified. The reason of setting up Threshold 1 is that not all system variations are significant enough to be identified. For example, as STEP 4b in Figure 6.1 shows, load changes, which lead to the $pcErr$ being smaller than Threshold 1, might be too insignificant to be worth being identified. System identification results contain accuracy tolerance. Even if small load changes are marked and the corresponding rail get estimated, if the unnoticeable changes in coefficients of a transfer function brought by system variations are within tolerance band of the old results, SI might be not sensitive enough to update the exact values of new results. In this case, the old results are still usable. To avoid this section to be influenced by excitation signals of SI, Threshold 1 should be highly larger than PRBS magnitudes.

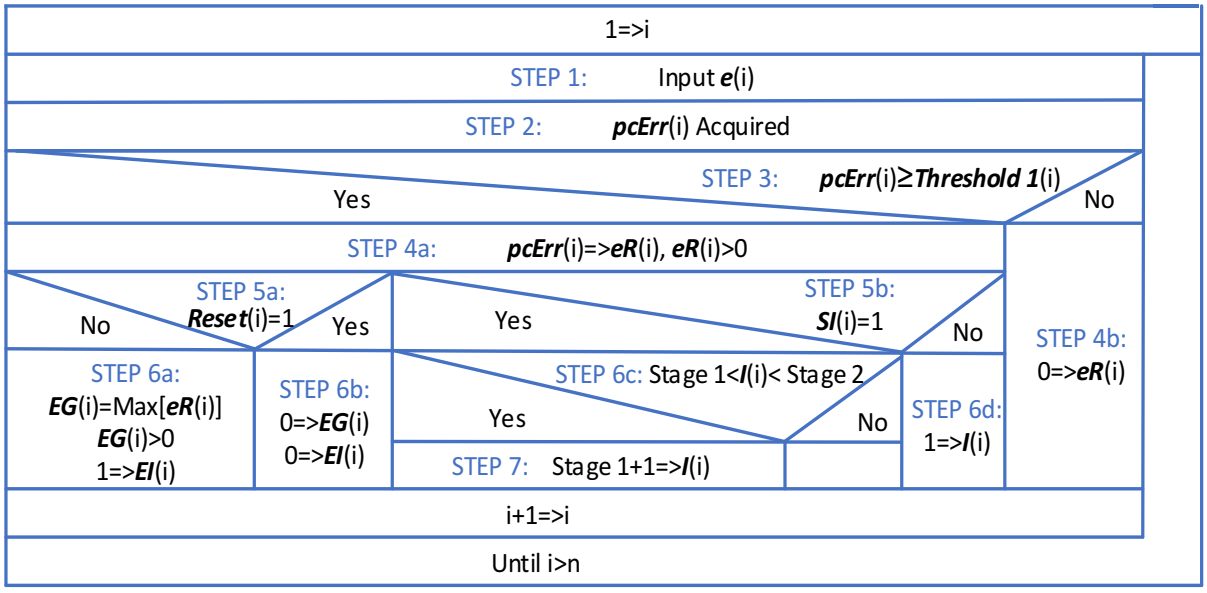


Figure 6.1 Flowchart of Maximum Error Value Collection

Besides, there will be transient responses in the voltage outputs of power converters dealing with abrupt system changes. System identification should take place after these transient behaviours, so another threshold (Threshold 2 in Figure 6.2) is introduced to detect whether the rail which indicates errors is ready for being ranked. Therefore, once there is an error signal magnitude which is above threshold 1 in a rail, several latest sampled error signals of the rail (the amount is pre-configured in **6.2.2A**) will be stored (STEP 1 in Figure 6.2). If the amount

of these sampled data ($Noss$) below Threshold 2 are above Steady State Indication Threshold, the rail will be viewed as 'have been in steady state/ready for being ranked', after which prioritization takes place (see STEP 2, 3, 4a, 5a and 5b in Figure 6.2). Meanwhile, the rails being identified already are still available for being ranked, examined in STEP 4b of Figure 6.2.

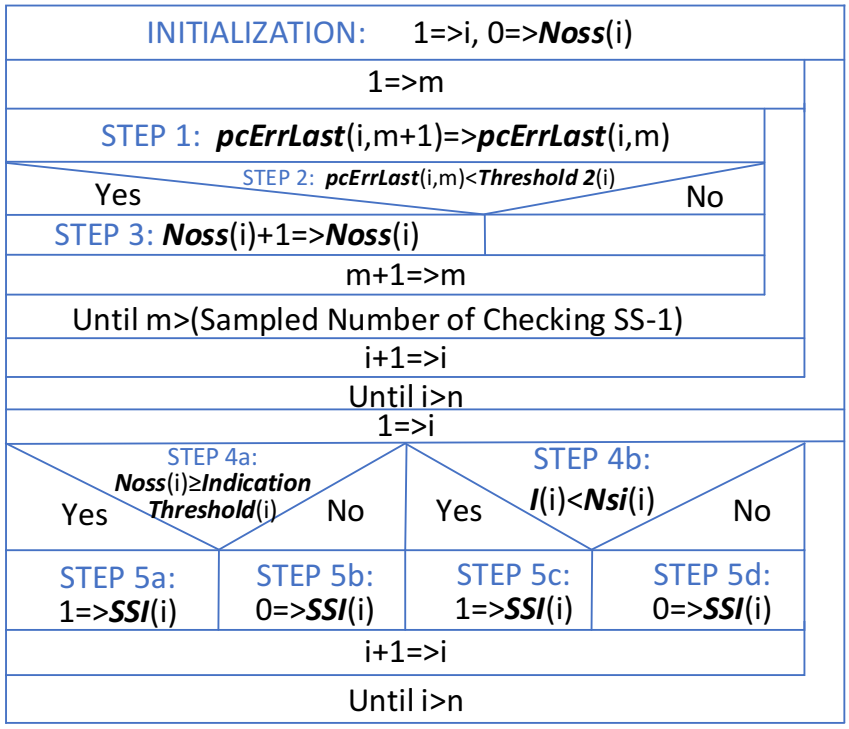


Figure 6.2 Flowchart of Checking the Availability of System Identification

2.Error-respect ranking decision: Figure 6.3 is the diagram showing the process of this section. If there are more than three rails which should, and have been ready to, be identified, error consideration module would be taken into use. Otherwise, the priority of each rail would follow the default rank based on application importance (see STEP 1, 2, 3 and 4b in Figure 6.3). Once error signal ranking module is switched on, the recorded maximum value of error signals of the rails which are ready for ranking (STEP 4a in Figure 6.3) will be compared. The larger the value of a rail is, the higher its priority is (STEP 5, 6 and 7 in Figure 6.3).

In a ranking queue, priorities of rails will be quantitated with numerical weights. As to the rail with the highest priority, its weight will be assigned as 1, and the weight decrement, from one rail to another one only one order lower, is the number of rails of the multi-rail power converter divided by 1. For example, if there are 5 rails in a multi-rail power converter, the weight decrement will be 1/5. Then weights of the five rails from the highest priority to the lowest would separately be 1, 0.8, 0.6, 0.4 and 0.2. These numerical weights based on error magnitude is named as 'error magnitude coefficient', of which Rail i is represented by 'a_i'. If there are 5 rails in a multi-rail power conversion architecture, and the rank of rails from the

highest to the lowest is: Rail 3, Rail 1, Rail 5, Rail 2 and Rail 4, the error magnitude coefficient matrix, '$Error$' in STEP 8 of Figure 6.3, will be written as:

$$[a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5] = [0.8 \quad 0.4 \quad 1 \quad 0.2 \quad 0.6] \tag{6-1}$$

This priority quantization strategy is also applied in the default ranking queue based on applications (see STEP 2b in Figure 6.3) and these coefficients describing priority are named as 'priority coefficients'.

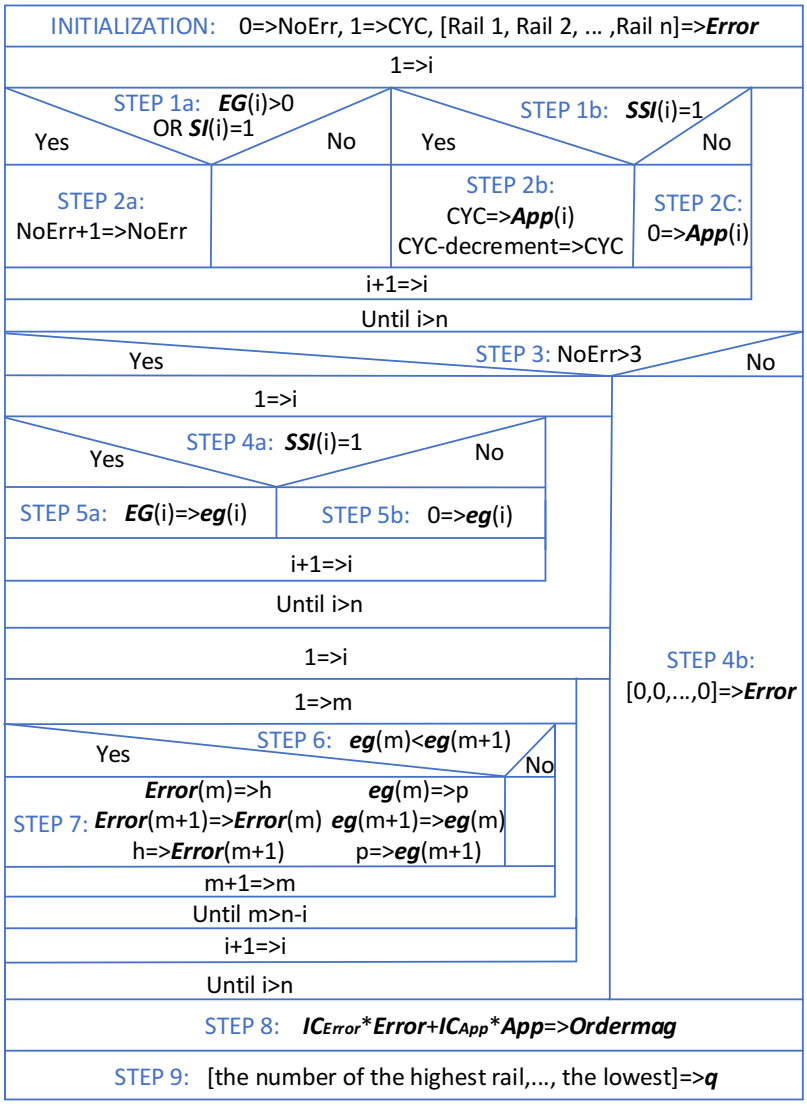


Figure 6.3 Flowchart of Updating of Importance Coefficients and Urgency Coefficient

3. urgency coefficient calculation (see STEP 8 in Figure 6.3): Elements what ranking rails is based on are not limited in application or error magnitude, others include the different frequencies of fault occurrence and requirements, etc. Rails featured of higher likelihood of suffering faults, or higher power density requirements, might be assigned with higher priority. Therefore, the number of ranking queues would equal how many elements considered. The priority of each rail in different queues might be different. the rail not suffering from load

change might be highly fault-prone and its application importance is moderate, then its position in the corresponding ranking queues will separately be the last, in the front and middle. After listing ranking queues of all elements, these queues will be mixed together to build a waiting queue, which suggests the sequence of rails going to be identified.

The specific way of deriving the waiting queue is also quantitation, same as the way numerically representing priorities in ranking queues. Elements are differently weighted in ratio way to indicate the importance of each element. For example, if the ratio of elements is 'application: error magnitude: fault likelihood=2:1:1', 'application' is double important than the other two are. After users configuring these ratio weights indicating element criticality degrees of elements, they would be transferred to percentage coefficients named as 'importance coefficient'. If the ratio value of the qth element is r_q, and there are m elements under consideration, the importance coefficient of the qth element, b_q, is:

$$b_q = \frac{r_q}{\sum_{q=1}^{m} r_q} \tag{6-2}$$

Then by adding up each product of 'Priority Coefficient' and 'Importance Coefficient' of a rail in different ranking queues, urgency coefficient, introduced to represent priorities of each rail in waiting queue, can be calculated. The larger the urgency coefficient of a rail is, more urgent the rail should be identified, and consequently the higher priority the rail is assigned.

$$Importance\ Coefficient\ Vector\ (ICV) = [b_1 \quad b_2 \quad \cdots \quad b_q \quad \cdots \quad b_m] \tag{6-3}$$

$$Priority\ Coefficient\ Matrix\ (PCM) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{q1} & a_{q2} & \cdots & a_{qi} & \cdots & a_{qn} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix} \tag{6-4}$$

$$Urgency\ Coefficient\ Vector\ (UCV) = ICV \times PCM = [b_1 \quad b_2 \quad \cdots \quad b_q \quad \cdots \quad b_m] \times$$

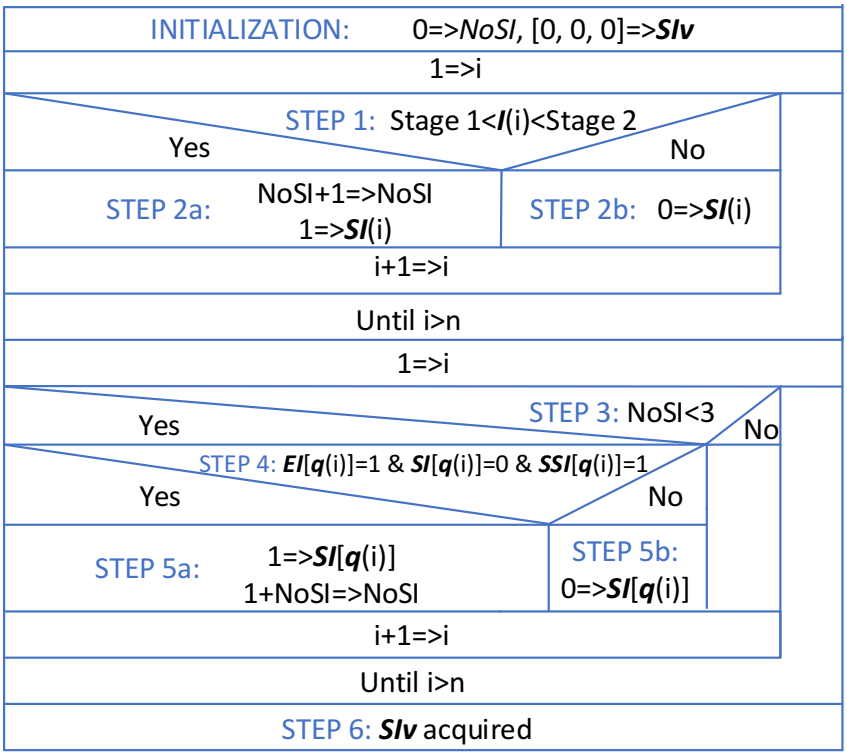
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{q1} & a_{q2} & \cdots & a_{qi} & \cdots & a_{qn} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} b_1a_{11} + b_2a_{21} + \cdots + b_qa_{q1} + \cdots + b_ma_{m1} \\ b_2a_{12} + b_2a_{22} + \cdots + b_qa_{q2} + \cdots + b_ma_{m2} \\ \vdots \\ b_1a_{1i} + b_2a_{2i} + \cdots + b_qa_{qi} + \cdots + b_ma_{mi} \\ \vdots \\ b_1a_{1n} + b_2a_{2n} + \cdots + b_qa_{qn} + \cdots + b_ma_{mn} \end{bmatrix} \tag{6-5}$$

Here, n is the number of rails, m is the number element under consideration, so in Priority Coefficient Matrix (PCM). The qth row represents the rank of each rail considering the qth element and the ith column the rank of the ith rail in different elements. Therefore, in Urgency Coefficient Vector (UCV), the ith row is the urgency coefficient of Rail i, and the whole vector would indicate the rank of all rails.

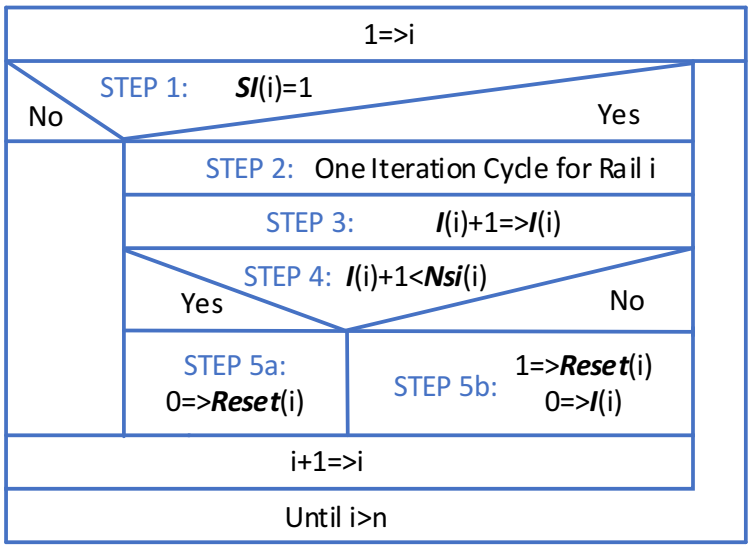
The number of elements can be multiple while only two are considered here ($q = 2$): application and error magnitude. The importance ratio of two respects is Application (r_1): Error Magnitude (r_2) = 3:2, so $b_1 = 0.6$ and $b_2 = 0.4$. In application respect, of which the corresponding rank is default, the number of each rail indicates its rank in the queue. Rail i ranks ith. In error magnitude respect, the rank from highest to lowest is: Rail 4, Rail 5, Rail 3, Rail 2 and Rail 1. Accordingly, the Urgency Coefficient Vector (UCV) can be calculated as (6-5) shows and the rank of waiting queue would be: Rail 1, Rail 2, Rail 4, Rail 3 and Rail 5. When the urgency coefficients of two or above rails are the same, it does not matter which rail is in front of/behind others in the waiting queue because: 1. they have same priorities. 2. the maximum capacity of simultaneous system identification is three, the rails which have same urgency coefficients and would be definitely consecutively ranked in the waiting queue have significant chances to be simultaneously identified.

$$Urgency\ Coefficient\ Vector\ (UCV) = ICV \times PCM = [b_1 \quad b_2] \times$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} = [0.6 \quad 0.4] \times \begin{bmatrix} 1 & 0.8 & 0.6 & 0.4 & 0.2 \\ 0.2 & 0.4 & 0.6 & 1 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.68 \\ 0.64 \\ 0.6 \\ 0.64 \\ 0.44 \end{bmatrix} \qquad (6\text{-}6)$$



(a)



(b)

Figure 6.4 Flowchart of Task Allocation of System Identification to Rails

4. allocating system identification procedure to rails (see Figure 6.4): Based on the urgency coefficients, a waiting queue made of rails which should be identified with different priorities (***q*** in the last step of Figure 6.3) is acquired. Consequently, tasks will be assigned sooner into rails with higher priorities.

In this paper, the task assigned to rail is real time system identification, including PRBS injection, relevant signals sampling and filtering and iteration of recursive algorithms. Allocating tasks into rails contain two steps: 1. Rails of which system identification process has been in Stage 2 will still be accolated with the task [see STEP 1 & 2 in Figure 6.4(a)]. 2. If the number of rails in Stage 2 is less than 3 (STEP 3 in Figure 6.4(a)), which means there is spare capability of simultaneous system identification, the rail with the highest priority among all rails which has not been assigned with system identification procedure will be selected for system identification until the capability is maximised to 3 [STEP 4&5 in Figure 6.4(a)]. Figure 6.4(b) is the iteration cycle flowchart for these task-assigned rails. The specific iteration process can be seen in Chapter 4, an Iteration Counter (I) would count the iteration times of all rails until it equals the pre-configured upper limit (N_{si} in Figure 6.4(b)), then an impulse signal ($Reset$) generates to reset the pending SI requests (EI in Figure 6.1).

6.3 Simulation results

This section verifies the proposed prioritization workflow through two investigations: Firstly, the workflow is compared with FCFS and pre-emptive strategies, to show that the proposed way would neither result in important rails waiting long time as FCFS does, nor lead to low-

priority rails losing identification chances in rush hours as 'pre-emptive' does. Secondly, results of the proposed workflow acquired from different respect ratio configurations are compared, which demonstrate that this workflow can be flexibly adjusted to satisfy different prioritization requirements. Before testing the workflow, the initial setups introduced in **6.2.2A** are configured as Table 6-1 shows.

Table 6-1 Initial Setups for Simulation Preparation

No.	Items	Values	No.	Items of Initial Setups	Values
1	n	5	**7**	***Sampled Numbers of Checking SS***	20
2	***Decrement***	0.2	**8**	***Steady State Indication Threshold***	19
3	m	2	**9**	**Nominal Voltage Output** (Volts)	1.8, 3.3, 5, 4, 2.5
4	r_q	3:7	**10**	***Stage 1*** (Times of Iteration)	60
5	***Threshold 1***	1.5%	**11**	***Stage 2*** (Times of Iteration)	140
6	***Threshold 2***	1.5%	**12**	N_{si} (Times of Iteration)	200

6.3.1 Comparison among the Proposed Workflow, 'FCFS' and 'Pre-Emptive'

There are five rails that will be prioritized when three above of them produce SI requests. The five rails are started at 0s, and then Rail 1, Rail 2 and Rail 3 would encounter sudden load changes respectively at 49ms, 35ms and 38.7ms. All the percentage errors ($pcErr$) changes in the three rails are larger than Threshold 1, the correspondingly triggered SI requests would be used to interrupt SI processes of others.

As the magnitude of eR is much higher than that of system variations, for space limits, Figure 6.5 presents eR, control error signals above Threshold 1, in terms of $ln\ (X + 1)$ rather than X (X is the magnitude of control error signals).

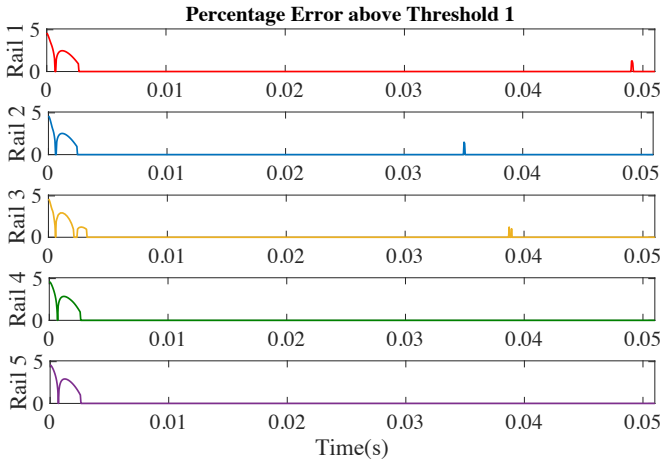
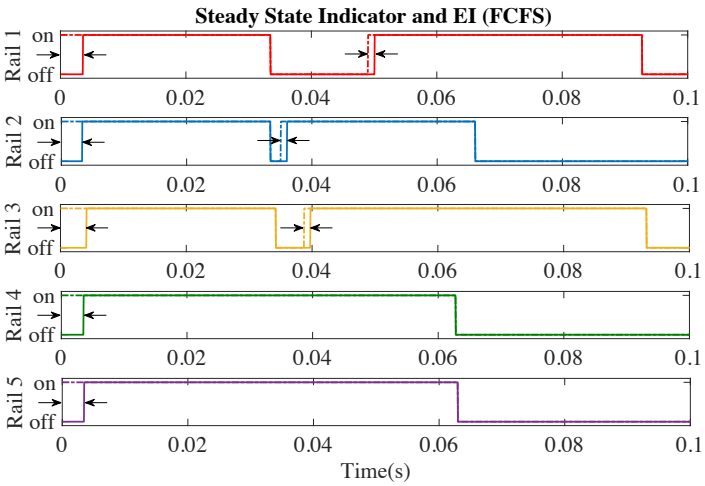
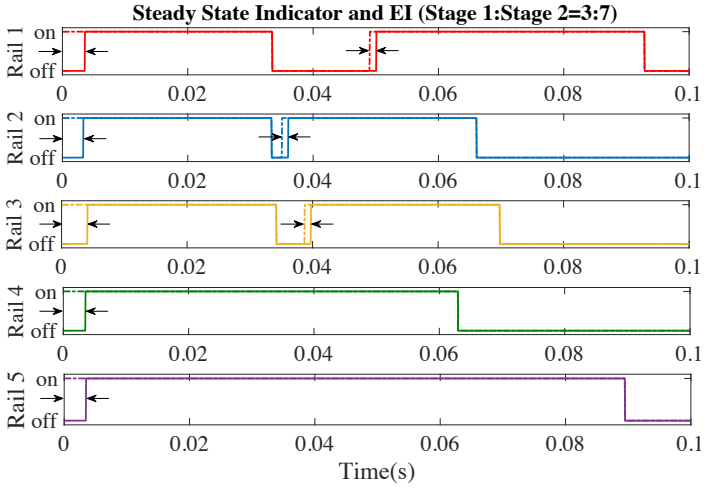


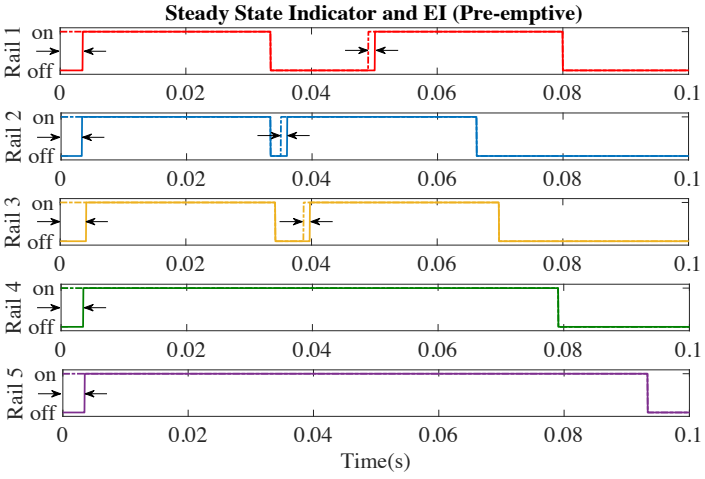
Figure 6.5 Percentage Error above Threshold 1 (eR) that can trigger SI requests



(a)



(b)



(c)

Figure 6.6 Steady State Indicator (SSI) and Waiting/Processing signal (EI) of each rail

Figure 6.6 shows Steady State Indicator (SSI) and waiting/processing signal (EI). According to the descriptions afore about waiting/processing signal and Steady State Indicator, the duration of ‘on’ of waiting/processing signal comprises that of ‘on’ of Steady State Indicator and that of transient responses coping with system variations. In Figure 6.6, dashed lines represent waiting/processing signals and solid ones Steady State Indicator, the areas that arrows suggest are transient responses. Therefore, Steady State Indicator shows time elapsed since rails being ready for SI until the end of the SI. The shorter the time length of its 'on' is, the higher efficiency of the proposed prioritisation approaches could achieve.

Table 6-2 lists the average time length of Steady State Indicator of each rail under different SI task allocation approaches (FCFS, pre-emptive and selectively pre-emptive). As Rail 1, 2 and 3 encountered SI requests two times, and the second one, in all three rails, occurred after the completion of the first SI processes, Steady State Indicator becomes 'on' two times. Hence, the average time length is the mean of the two 'on' of Steady State Indicator. While Rail 4 and 5 have only one SI request separately, the average time length is the 'on' time of Steady State Indicator corresponding to the only-one SI request.

According to Table 6-3 and descriptions of SI processes among five rails, using 'FCFS' may result in the shortest average waiting time, 45.36ms, of all approvers. As to individual rails, approvers with higher priorities can only be identified first when several rails request SI at the same time, while if they ask during others' SI processes, there is not advantages on being identified quicker. Therefore, compared with pre-emptive ways, the variance among the five rails’ pending/processing time, which expresses the difference of the average pending/processing times between rails with higher priorities and those with lower priorities, of FCFS is the smallest.

However, in pre-emptive way, since that SI request from rails with higher priorities are able to interrupt SI processes of rails with lower priorities, Rail 4 and 5 will be pending longer and variance among rails’ average pending/processing times is the largest, compared with that in the other two ways. Besides, the average pending/processing time (51.08ms) of being purely pre-emptive is the longest. Consequently, although rails with higher priorities could be identified with minimized waiting time, Rail 4 and 5 could be wait for unreasonably long. In circumstances that SI requests of Rail 1, 2 and 3 are highly frequent, Rail 4 and 5 might rarely get a chance to be identified.

As the proposed way is featured with being selectively pre-emptive, there are both chances for SI requests from Rails with higher priorities interrupting SI processes of lower priority rails and for SI processes of rails with lower priorities keep taking placing without interruption.

Therefore, compared with those of being purely pre-emptive or following 'FCFS' rule, the average pending/processing time among the five rails of being 'selectively pre-emptive' is moderate, as well as the variance is. Furthermore, as the proposed way allows users to flexibly change stage length ratio (Stage 1: Stage 2), if Stage 1 becomes longer, the chance of SI requests from higher priority rails interrupting SI processes of lower priority rails will become larger. the proposed SI task allocation approach will be more 'pre-emptive' alike. Otherwise, when Stage 2 is longer, the likelihood of SI requests from rails not being identified falling into Stage 2 of rails being identified raises, and SI processes of any rails are uninterruptible in their Stage 2, this scenario, therefore, is more 'FCFS' alike.

Overall, there is a trade-off between minimizing overall pending time and shortening pending time for higher priority rails only. Applying the proposed approach, users may flexibly manipulate the length ratio between Stage 1 and Stage 2 to match their case-by-case requirements and make reasonable balances between being 'FCFS' and being pre-emptive.

Table 6-2 Time Records for Each Rail Dealing with SI Requests

Strategy	**Records**	**Rail 1 (ms)**		**Rail 2 (ms)**		**Rail 3 (ms)**		**Rail 4 (ms)**	**Rail 5 (ms)**
		First	**Second**	**First**	**Second**	**First**	**Second**		
FCFS	Start	3.65	50.15	3.4	36.1	4.2	39.8	3.6	3.55
	end	33.65	92.75	33.4	66.1	34.2	93.4	62.8	63.05
	Duration	30	42.6	30	30	30	53.6	59.2	59.5
Proposal	Start	3.65	50.15	3.4	36.1	4.2	39.8	3.6	3.55
	end	33.65	92.75	33.4	66.1	34.2	69.8	62.8	89.55
	Duration	30	42.6	30	30	30	30	59.2	86
Pre-emptive	Start	3.65	50.15	3.4	36.1	4.2	39.8	3.6	3.55
	end	33.65	80.15	33.4	66.1	34.2	69.8	79.15	93.35
	Duration	30	30	30	30	30	30	75.55	89.8

Table 6-3 Average Processing Duration of Each Rail

Strategy	**Rail 1 (ms)**	**Rail 2 (ms)**	**Rail 3 (ms)**	**Rail 4 (ms)**	**Rail 5 (ms)**	**Overall (ms)**	**Variance**
FCFS	36.3	30	41.8	59.2	59.5	45.36	180.543
Proposal	36.3	30	30	59.2	86	48.3	588.47
Pre-emptive	30	30	30	75.6	89.8	51.08	858.392

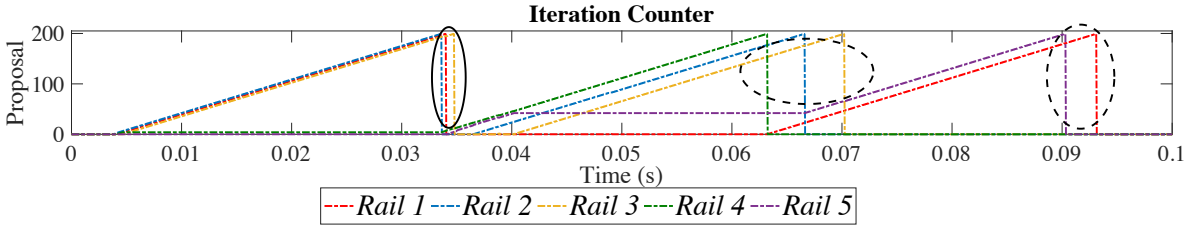


Figure 6.7 The Difference between 'Time Delay' (the solid circle) and 'Phase Differences' (the dashed circles)

As durations of transient responses of rails are different, even if rails start to operate at the same time, they might reach steady state at slightly different time. Consequently, the simultaneous SI among three rails might not be precisely synchronised. Besides, as abrupt changes, which trigger the starts of SI, occur in different time among rails, and it is highly possible that rails being identified would be interrupted by others with higher priorities, there will be 'phase differences', pointed out by the dashed circles, among rails being identified. However, compared with the phase differences caused by various occurrence time of system variances, interruptions of SI processes, etc., time delays, implied with the solid circle, led by reaching steady states at different time are normally insignificant and will not be discussed here.

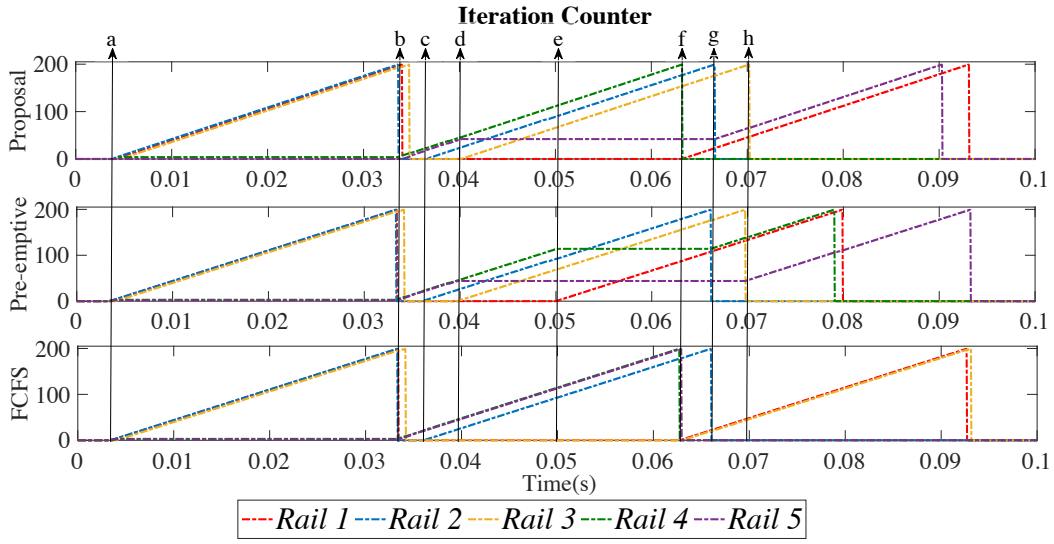


Figure 6.8 The Prioritization and SI Task Allocation Processes by Using Different Coordination Strategies

Table 6-4 The Coordination Records of the Proposed Workflow

Points		**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	
Events		**1, 2, 3** start	**1, 2, 3** end, **4, 5** start	**2** starts	**3** interrupts **5**	none	**1** starts as **4** ends	**5** continues as **2** ends	**3** ends	
Processing	none	**1, 2, 3**	**4, 5**	**2, 4, 5**	**2, 3, 4**	**2, 3, 4**	**1, 2, 3**	**1, 3, 5**	**1, 5**	
Waiting	none	**4, 5**	none	none	**5**	**1, 5**	**5**	none	none	

Table 6-5 The Coordination Records of ‘Pre-emptive’

Points		**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	
Events		**1, 2, 3** start	**1, 2, 3** end, **4, 5** start	**2** starts	**3** interrupts **5**	**1** interrupts **4**	none	**4** continues as **2** ends	**5** continues as **3** ends	
Processing	none	**1, 2, 3**	**4, 5**	**2, 4, 5**	**2, 3, 4**	**1, 2, 3**	**1, 2, 3**	**1, 3, 4**	**1, 4, 5**	
Waiting	none	**4, 5**	none	none	**5**	**4, 5**	**4, 5**	**5**	none	

Table 6-6 The Coordination Records of ‘FCFS’

Points		**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	
Events		**1, 2, 3** start	**1, 2, 3** end, **4, 5** start	**2** starts	none	none	**1, 3** start as **4, 5** end	**2** ends	none	
Processing	none	**1, 2, 3**	**4, 5**	**2, 4, 5**	**2, 4, 5**	**2, 4, 5**	**1, 2, 3**	**1, 3**	**1, 3**	
Waiting	none	**4, 5**	none	none	**3**	**1, 3**	none	none	none	

In Figure 6.8 there are some time points labelled as ‘a, b, …, h’ indicating starts, completions, interruptions and continuations of SI processes of the five rails, the specific events happened at these points are explained in Table 6-4 to Table 6-6. Before SI, the default priority of rails from the highest to the lowest is: Rail 1, 2, 3, 4 and 5. In the three tables, numbers are the numbers of rails. Points locate some important moments in SI processes among five rails in the 0.1s, and ‘Events’ describe the changes at these moments. ‘Processing’ lists the rails being identified between two consecutive moments. The rails waiting for being identified from highest priority to the lowest are listed in ‘Waiting’.

At Point **a**, all five rails have reached steady state and ready for SI, as the first three rails have higher priorities than Rail 4 and 5, their parameters are estimated first.

At Point **b** where SI processes of the first three rails end, Rail 4 and 5 start to be identified.

At Point **c**, after suffering from an abrupt load change, Rail 2 is getting steady state and identified. Its SI process does not interrupt any others since that until this period, the capability of simultaneous SI has not fully exploited. The maximum capability is three and only two rails (Rail 4 and 5) are being identified.

At Point **d** where Rail 3 gets load change and ready for SI, the differences among the three approaches, being selectively pre-emptive, being purely pre-emptive and FCFS, are expressed. Here, four rails, Rail 2, 3, 4 and 5 should be identified while the only three of them can be fulfilled first. Following the ‘First Come First Served’ rule, Rail 3, which has higher priority than Rail 4 and 5 though, should wait until the earliest one of the other three finishes. Oppositely, being pre-emptive results in the SI request of Rail 3 interrupting the SI process of Rail 5 which has the lowest priority. The same interruption will occur in ‘being selectively pre-emptive’, as at this moment the SI processes of Rail 4 and 5 are both in Stage 1, which means they are interruptible. Consequently, after Point **d**, SI are fulfilling among Rail 2, 3 and 4 in being pre-emptive scenarios, while in FCFS routine, SI processes still carry out among Rail 2, 4 and 5.

At Point **e**, there is a load change in Rail 1. In 'being selectively pre-emptive' routine, as SI processes of Rail 2, 3 and 4 have all entered Stage 2 uninterruptible at this moment, they continue even if Rail 1 has the highest priority. In 'being purely pre-emptive', the SI request of Rail 1 stops Rail 4 from being identified while under FCFS rule, SI processes being fulfilling will not be influenced by others. Therefore, after Point e, apart from the rails being estimated are updated to be Rail 1, 2 and 3 with 'being purely pre-emptive', they using the other two approaches would not change.

At Point **f**, in 'being selectively pre-emptive', when SI of Rail 4 ends, Rail 1 starts given that it has the highest priority in the waiting queue in which Rail 1 and 5 are. As to FCFS, SI processes of Rail 4 and 5 end, followed by the two rails, Rail 1 and 3, in waiting queue. There is not variation of rails being identified in 'being pre-emptive'.

At Point **g**, the SI process in Rail 2 in all three scenarios complete. Consequently, in 'being selectively pre-emptive', Rail 5, the only rail in waiting queue, continues after the pending duration from Point **d** to **g**, while in 'being pre-emptive', Rail 4 continues as the pending rails are Rail 4 and 5 and Rail 4 has higher priority.

At Point **h**, in 'being pre-emptive', the completion of SI of Rail 3 triggers the SI continuation of the last rail, Rail 5, pending in the waiting queue. From now on, the waiting queue of all three scenarios become empty and no rail would require SI except those are being identified. Therefore, these ongoing SI processes will continue until they complete at different moments.

In summary, according to the SI task coordination processes among the five rails, and Table 6-2 which lists the average existing times for SI requests, 'FCFS' may achieve the shortest average waiting/processing time for SI requests while high-priority rails won't be coped with earlier. In controversy, 'pre-emptive' could guarantee important rails being identified sooner, although these low-priority rails would wait ages, which causes the longest average waiting/processing time. The most contribution of the proposed workflow is that it could well balance the SI needs from either high-priority rails or low ones, which results in the mediate average waiting/processing time consequently. By manipulating the time length ratio between ***Stage 1*** and ***Stage 2***, the proposed approach can work either 'FCFS' ($\boldsymbol{N_{SI}}$ equals the entire SI length) or 'Pre-emptive' (***Stage 1*** equals the entire SI length) alike.

6.3.2 Comparison of the Proposed Workflow with Different Importance Ratios

This section presents comparisons of simulation results under different Importance Coefficients (IC). When the respects considered for ranking rails in waiting queue are only Application and Error Magnitude, if the importance ratio of Application and Error Magnitude is 3:7, rails which suffer the severer system variations might be scored with higher Urgency

Coefficient (UC) and, therefore, ranked higher in a waiting queue for being identified quicker. Otherwise, if the ratio is Application: Error Magnitude=7:3, the application importance of rails would be mainly considered during ranking rather than the magnitude of error signals. Then the more important fields rails are used in, the frontier they are likely to be ranked in waiting queue. Overall, different importance ratio setups might result in a rail being differently ranked in waiting queue. This section also demonstrates the plan for coping with abrupt system changes during SI processes.

The configurations of times, types and magnitudes of system variations of each rail, as well as transfer function coefficients after these changes, are introduced in Table 6-7. Tracked signals include eR, EG, SSI, Output voltages, iteration counters and identification results/recursive curves.

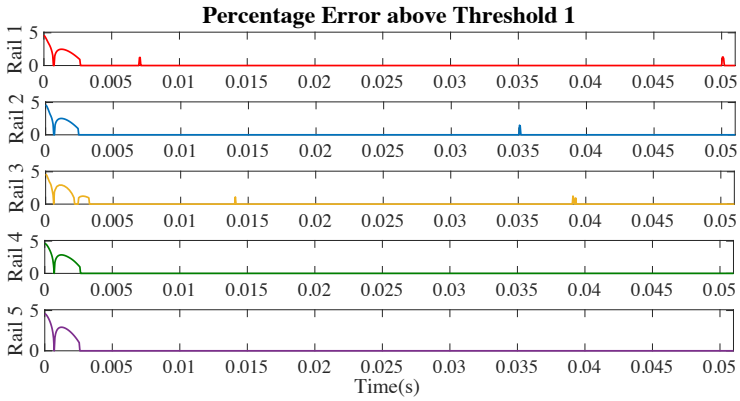
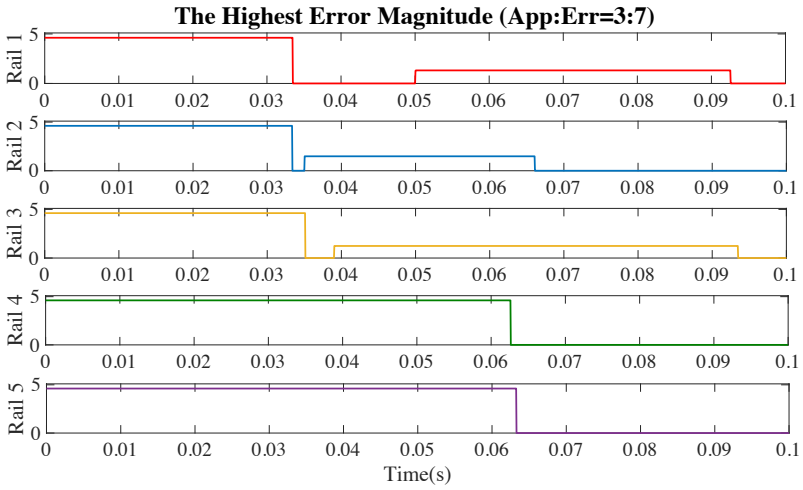


Figure 6.9 Percentage Error above Threshold 1 that can trigger SI requests

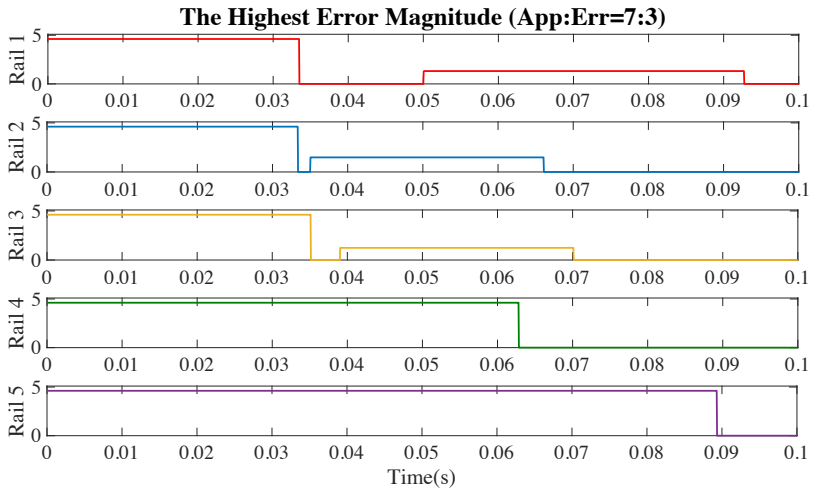
Table 6-7 Transfer Function Coefficients before/after Load Changes of Each Rail

Rail	**Time**	**Variation Types and Resistance Load Values**	**Transfer Function Coefficients**			
			a1	**a2**	**b1**	**b2**
Rail 1	**0s**	Initiation (5Ω)	-1.9348	0.9586	0.1759	0.0624
	7ms	Load Change (2.5Ω)	-1.9150	0.9387	0.1759	0.0619
	50ms	Load Change (5Ω)	-1.9348	0.9586	0.1759	0.0624
Rail 2	**0s**	Initiation (5Ω)	-1.9163	0.9500	0.2258	0.1118
	35ms	Load Change (2.5Ω)	-1.8886	0.9221	0.2252	0.1103
Rail 3	**0s**	Initiation (10Ω)	-1.9066	0.9572	0.3099	0.1955
	14ms	Load Change (5Ω)	-1.8856	0.9358	0.3087	0.1932
	39ms	Load Change (10Ω)	-1.9066	0.9572	0.3099	0.1955
Rail 4	**0s**	Initiation (10Ω)	-1.9596	0.9756	0.1176	0.0420
Rail 5	**0s**	Initiation (1.8Ω)	-1.9414	0.9527	0.1138	-0.0004

eR, the absolute value of percentage control error signals above Threshold 1, is presented in Figure 6.9, in terms of $ln\ (X + 1)$ (X is the magnitude of control error signals), which shows that the error magnitude of converter initiation is much higher than that of load changes. Both of them, however, are able to trigger SI requests.



(a)

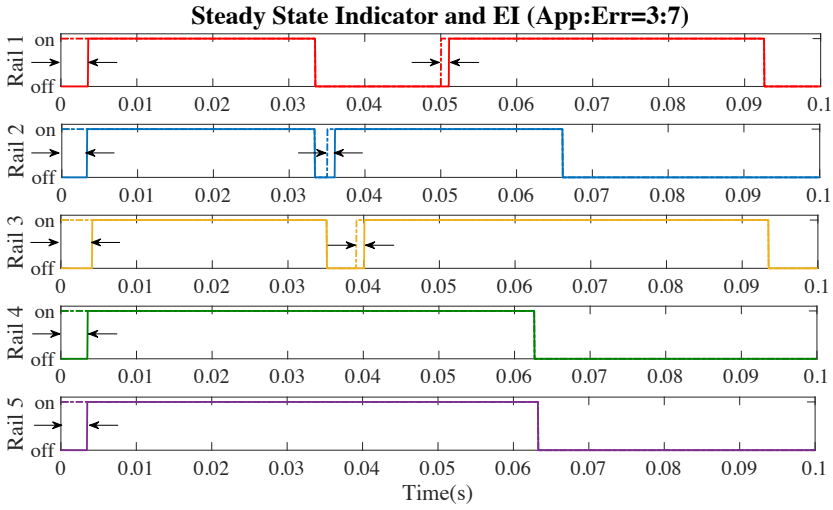


(b)

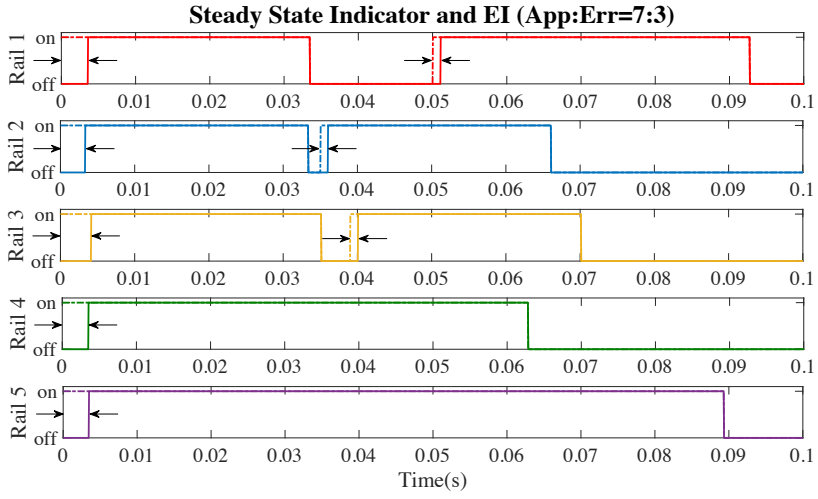
Figure 6.10 The Highest Error Magnitude (EG) of Each Rail

EG, indicating the maximum value of eR, is collected in Figure 6.10 to define system variation severity. Accordingly, EG led by rail initiation of all five rails are the same, 100, while when loads are changed in Rail 1, 2 and 3 by 50%, EG are respectively 2.7, 3.4 and 2.46. Consequently, if Importance Coefficients (IC) in error magnitude respect are the same (EGs are the same), rail prioritization is mainly based on IC in application respect. However, when ICs in error magnitude respect are different, Urgency Coefficients, deciding priorities of each rail, would be affected by importance coefficients of both error magnitude and application.

SSI and *EI*, reflecting time elapsed since SI request has arrived to SI completion, are recorded in Figure 6.11. Figure 6.12, presenting output voltages of five rails, indicates that PRBs are only injected in rails being identified so that rails with pending SI request would not be affected by perturbations (PRBS). Different from [204], which proposes sampling and storing signals with PRBS into a place when the corresponding SI request is still pending in advance for later usage, this strategy, featured of real time PRBS injection, sampling and sampled signal usage (SI), may save memory spaces as it does not require places for storage. Besides, it also avoids the situation that system changes occur after storing signals, which means that the stored signals without update may not produce newest identification results.

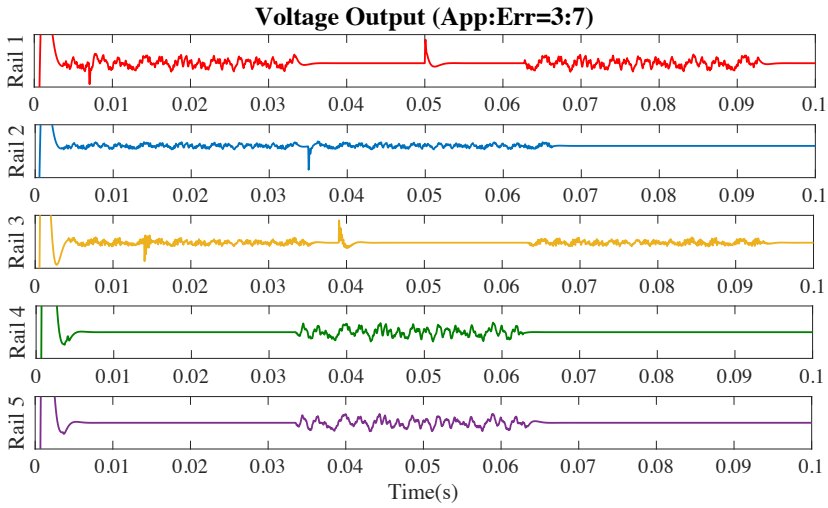


(a)

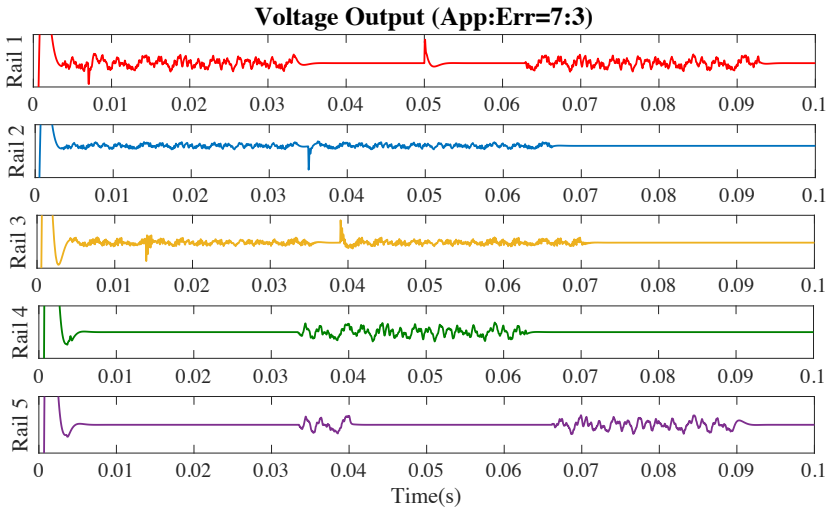


(b)

Figure 6.11 Steady State Indicator (SSI) and Waiting/Processing signal (EI) of each rail



(a)



(b)

Figure 6.12 Voltage Outputs of Each Rail under PRBS Effects

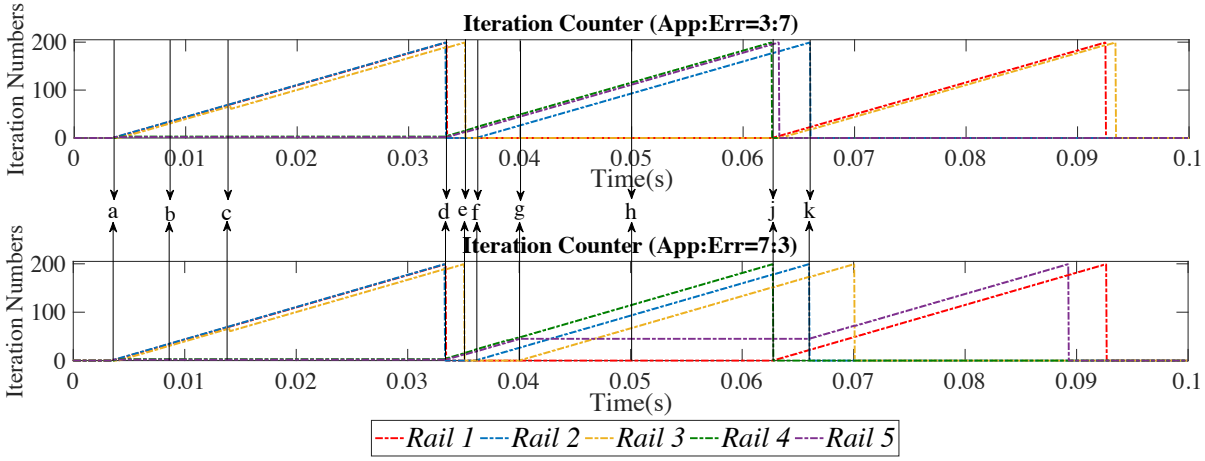


Figure 6.13 The Prioritization and SI Task Allocation Processes from Different Respect Ratios

Figure 6.13 shows the iteration processes of the five rails following the prioritization and SI task allocation rules indicated from Figure 6.1 to Figure 6.4. Same as result demonstrations in **6.3.1**, starts, completions, interruptions and continuations of SI processes of each rail are marked as Point **a**, **b**, …, **k** in Figure 6.13, the events that occur at these time points are explained in Table 6-10 and 6-11. As the magnitudes of control error signals caused by system variations are various, the Importance Coefficient of each rail in 'error magnitude' respect will be dynamically varying throughout the simulated 0.1s, so is Urgency Coefficient (UC) consequently. Table 6-8 and 6-9 list the UCs of each rail in different periods, as well as their ranks.

At Point **a**, all rails are ready for SI after initiations. As the five rails are featured with the same value of EG (100), their IC (Importance Coefficients) in 'error magnitude' respect are respectively their rail numbers, same as those in 'application' respect (default setup). Consequently, the first three rails get identified due to their highest priorities (see Figure 6.13 and Table 6-10), while Rail 4, with higher priority than Rail 5, is listed in front of Rail 5 in waiting queue until Point d.

At Point **b**, at which a load change occurs in Rail 1, since that its SI process is in Stage 1, iteration counter kept accounting without any intervention in Rail 1. While at Point **c** where SI process of Rail 3 has been in Stage 2 (iteration counter is 66 now and Stage 2 is configured from iteration counter equalling 61), the proceeding SI process would be back to the beginning of Stage 2 (iteration counter becomes 61) and continue when load of this rail varies (see Figure 6.13). As a result, SI of Rail 1 and 2, at Point **d**, completed in advance of Rail 3, and then Rail 4 and 5 start to be identified. As of prioritization rules indicated in Figure 6.3, SI completions of Rail 1 and 2 lead to their ICs in 'error magnitude' respect becoming 0.4 and 0.2 respectively and in 'application' respect both 0. Hence, during the period from **d** to **e**, the five rails are ranked from highest to lowest as: Rail 3, Rail 4, Rail 5, Rail 1, Rail 2. At Point **e**, Rail 3 finishes SI. From now on there is spared simultaneous SI capability allowing one rail to be identified, and the rank of the five rails are further changed as shown in Figure 6.13.

At Point **f**, after raising a SI request due to encountering a load change, Rail 2 is ready to be identified. Then, without considering priorities of each rail, SI process of Rail 2 starts, as the process is now taking the spared SI capability that occurred from Point e. At Point **g**, when the number of rails being able to simultaneously identified has been maximised and Rail 3 suffers a load change again, UCs-based priorities of rails would be taken back into account. Now, there are four rails (Rail 2, 3, 4 and 5) containing undertaking or pending SI requests. EG of these rails are respectively 3.4116, 2.4556, 100 and 100, then their corresponding ICs in 'Error

Magnitude' respect are 0.6, 0.4, 1 and 0.8 separately (see Table 6-8). If the respect ratio between 'Application' and 'Error Magnitude' (written as 'App: Err') equals 3:7, as a result of the IC (Importance Coefficient) from 'Error Magnitude' respect occupying more weights for UC calculation, the four rails would be ranked by their UCs, from the highest priority to the lowest, as: Rail 4, Rail 2, Rail 5 and Rail 3 (the corresponding UCs are listed in Table 6-8). Therefore, with the lower priority, Rail 3 would not interrupt the SI process of Rail 5. While when App: Err=7:3, at which IC of 'application' respect has more weights, the priority of Rail 3 would be higher than that of Rail 5 (see Table 6-9). Besides, the SI process of Rail 5 is now in Stage 1, stoppable, therefore, interrupted by SI request of Rail 3.

At Point **h**, corresponding to a load change which causes EG equalling 2.7364, Rail 1 is ready for SI, while rails being identified are all in Stage 2, uninterruptible. Hence, regardless of the priority of Rail 1 at this stage, the SI request of Rail 1 would not interrupt SI processes of the three rails (Rail 2, 4 and 5 in 'App: Err=3: 7', Rail 2, 3 and 4 in 'App: Err=7: 3'). In 'App: Err=3: 7' SI requests of Rail 1 and 3 are pending, as ICs of Rail 1 in both 'error magnitude' and 'application' respects is higher than the two ICs of Rail 3, Rail 1 ranks higher than Rail 3 in waiting queue, it could be identified first once simultaneous SI system is available. While in 'App: Err=7: 3' Rail 3 and 5 are waiting for SI. According to their UCs in Table 6-9, Rail 3 is featured with higher priority.

At Point **j**, in 'App: Err=3: 7', with the SI completions of Rail 4 and 5, two available places of simultaneous identification occur so that rail 1 and 3 start to be identified. In 'App: Err=7: 3', by the end of identifying Rail 4 only one available place is provided. The rail with highest priority in waiting queue (Rail 1), therefore, get the place and start to be identified. Rail 5 still wait until Point k, at which a spare place comes out due to the SI completion of Rail 2.

Table 6-8 Urgency Coefficient Changes from 'App:Err=3:7'

	Rail 1	**Rail 2**	**Rail 3**	**Rail 4**	**Rail 5**	**Queue**
a-d	1, 1, 1	0.8, 0.8, 0.8	0.6, 0.6, 0.6	0.4, 0.4, 0.4	0.2, 0.2, 0.2	1, 2, 3, 4, 5
d-e	0, 0.4, 0.28	0, 0.2, 0.14	1, 1, 1	0.8, 0.8, 0.8	0.6, 0.6, 0.6	3, 4, 5, 1, 2
e-f	0, 0, 0	0, 0, 0	0, 0, 0	1, 0, 0.3	0.8, 0, 0.24	4, 5, 1, 2, 3
f-g	0, 0, 0	1, 0, 0.3	0, 0, 0	0.8, 0, 0.24	0.6, 0, 0.18	2, 4, 5, 1, 3
g-h	0, 0.2, 0.14	1, 0.6, 0.72	0.8, 0.4, 0.52	0.6, 1, 0.88	0.4, 0.8, 0.68	4, 2, 5, 3, 1
h-j	1, 0.4, 0.58	0.8, 0.6, 0.66	0.6, 0.2, 0.32	0.4, 1, 0.82	0.2, 0.8, 0.62	4, 2, 5, 1, 3
j-k	1, 0, 0.3	0.8, 0, 0.24	0.6, 0, 0.18	0, 0, 0	0, 0, 0	1, 2, 3, 4, 5

Table 6-9 Urgency Coefficient Changes from ‘App:Err=7:3

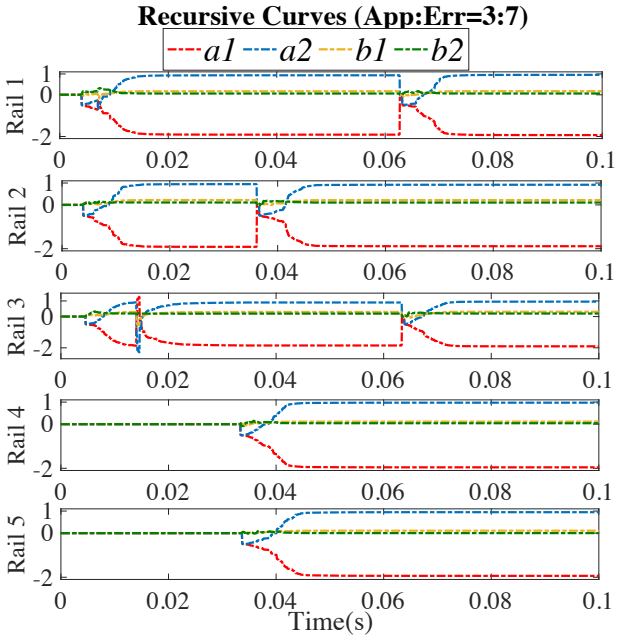
	Rail 1	Rail 2	Rail 3	Rail 4	Rail 5	Queue
a-d	1, 1, 1	0.8, 0.8, 0.8	0.6, 0.6, 0.6	0.4, 0.4, 0.4	0.2, 0.2, 0.2	1, 2, 3, 4, 5
d-e	0, 0, 0	0, 0, 0	1, 0, 0.7	0.8, 0, 0.56	0.6, 0, 0.42	3, 4, 5, 1, 2
e-f	0, 0, 0	0, 0, 0	0, 0, 0	1, 0, 0.7	0.8, 0, 0.56	4, 5, 1, 2, 3
f-g	0, 0, 0	1, 0, 0.7	0, 0, 0	0.8, 0, 0.56	0.6, 0, 0.42	2, 4, 5, 1, 3
g-h	0, 0.2, 0.06	1, 0.6, 0.88	0.8, 0.4, 0.68	0.6, 1, 0.72	0.4, 0.8,0.52	2, 4, 3, 5, 1
h-j	1, 0.4, 0.82	0.8, 0.6, 0.74	0.6, 0.2, 0.48	0.4, 1, 0.58	0.2, 0.8, 0.38	1, 2, 4, 3, 5
j-k	1, 0.6, 0.88	0.8, 0.8, 0.8	0.6, 0.4, 0.54	0, 0.2, 0.06	0.4, 1, 0.58	1, 2, 5, 3, 4

Table 6-10 The Coordination Records of ‘App:Err=3:7’

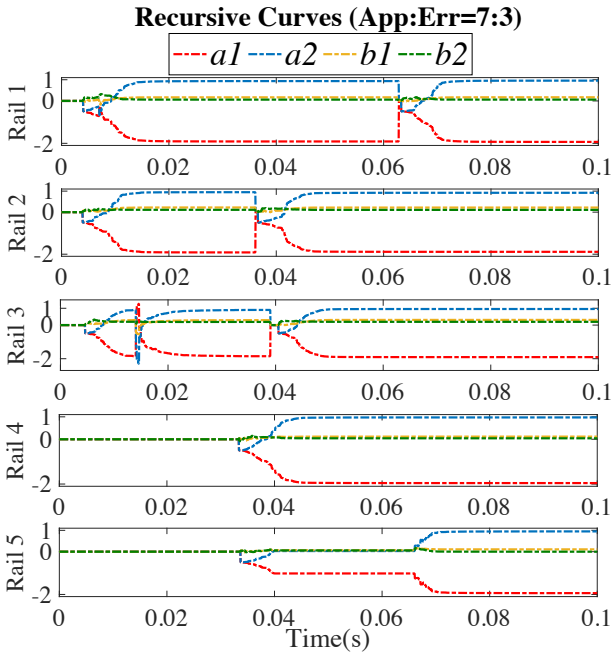
Points		**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**j**	**k**	
Events		**1, 2, 3** start	**1** changes	**3** changes	**4, 5** start as **1, 2** end	**3** ends	**2** starts	**3** changes	**1** changes	**1, 3** start as **4, 5** end	**2** ends	
Processing	none	**1, 2, 3**	**1, 2, 3**	**1, 2, 3**	**3, 4, 5**	**4, 5**	**2, 4, 5**	**2, 4, 5**	**2, 4, 5**	**1, 2, 3**	**1, 3**	
Waiting	none	**4, 5**	**4, 5**	**4, 5**	none	none	none	**3**	**1, 3**	none	none	

Table 6-11 The Coordination Records of ‘App:Err=7:3’

Points		**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**j**	**k**	
Events		**1, 2, 3** start	**1** changes	**3** changes	**4, 5** start as **1, 2** end	**3** ends	**2** starts	**3** changes	**1** changes	**4** ends	**2** ends	
Processing	none	**1, 2, 3**	**1, 2, 3**	**1, 2, 3**	**3, 4, 5**	**4, 5**	**2, 4, 5**	**2, 3, 4**	**2, 3, 4**	**1, 2, 3**	**1, 3, 5**	
Waiting	none	**4, 5**	**4, 5**	**4, 5**	none	none	none	**5**	**1, 5**	none	none	



(a)



(b)

Figure 6.14 System Identification Results of Each Rail

According to Figure 6.14, showing the SI processes of each rail under different prioritization configurations, the proposed workflow is verified to be able to work as it originally designed: Firstly, with different respect ratios, SI processes of lower-priority rails in Stage 1 would be interrupted by SI requests from higher-priority rails. Secondly, the intermediate results during the suspending periods could be saved for later use, instead of being disposed.

6.4 Chapter Summary

This Chapter proposed a dynamic prioritization approach, which can be flexibly adjusted to work both 'FCFS' and 'Pre-emptive' alike. In addition, it could consider various respects for decision-making and save the intermediate results from an interrupted SI process for cost saving. The simulation results verify that the proposed workflow could result in different task allocation ways to match individual requirements.

Chapter 7 Conclusion and Future Work

7.1 Conclusion

Multi-Rail power conversion architectures, in virtue of their flexibility and ease of configuration, have been widely employed in industries. Due to the increasingly complicated working conditions and loads, there have been plenty of studies on adaptive control techniques, applied to achieve robust control of SMPCs suffering from unexpected, frequent or abrupt system variations. Real-time parameter estimation, as a useful tool that adaptive control is based on, has been investigated in many publications, in which the performance evaluation of parameter estimation is typically based on the estimation speed (the convergence time), the estimation accuracy, the computational efforts and the ability of abrupt disturbance rejection. Researchers are inclined to select an ideal algorithm (adaptive filters) used for parameter estimation which can result in the highest estimation accuracy, the shortest convergence time with the least computational costs and the strongest disturbance rejection ability, compared with those algorithms that have been introduced in system identification of power electronics. These algorithms are all investigated in single buck converters so far. When an algorithm is applied on multi-rail architectures with a centralized single controller, the convergence time, estimation accuracy and the disturbance rejection ability would not change, however, the computational burdens will become heavy, increasing proportionately with the addition of rails. For example, if the available computation time is 50μs, the employed processor should finish 64 additions, 109 multiplications, and 1 division in 50μs for single-rail parameter estimation by using RLS. If three rails are simultaneously identified, the computational burden in the 50μs will be increased to 192 additions, 327 multiplications, and 3 divisions. The significant increase in the computational burden in one sampling event could cause the need for advanced processors more computationally capable particularly, resulting in extra investments. Therefore, rather than introducing other algorithms in parameter estimation areas, this paper focuses on the optimization of the estimation processes of multi-rail structures.

The iteration decimation approach is firstly proposed. Through decreasing the iteration frequency of the applied recursive algorithms, e.g. RLS, AP and KF, etc., the iteration cycles of each rails would take place alternatively. After every sampling event, the iteration action is arranged in only one rail instead of in the all rails. However, as the reduced iteration frequency would prolong the estimation time, the convergence-time-related factors will be appropriately adjusted after the investigation of the relation among these factors, the convergence time and the iteration frequency. These factors include the forgetting factor (λ) in RLS, the step size (μ)

in AP and the observation noise variance (r) in KF, etc., and it is known that decreasing λ or r in RLS or KF could raise the estimation speed while a smaller μ would lead to a longer convergence time. Moreover, as the variations of these factors also affect estimation accuracy and the stability of the converged curves, the estimation process is divided into two stages and the factor adjustment caused by the iteration frequency reduction only conducted when the recursive curves have not been converged. Consequently, the computational cost spent on the simultaneous parameter estimation of a multi-rail power architecture during every sampling event can be alleviated from 'the cost of one iteration cycle of a single rail multiplied by the number of rails' to 'the cost of one iteration cycle of a single rail', without compromises of the convergence time and the estimation accuracy.

Moreover, the iteration decimation approach may also enhance the disturbance rejection ability of parameter estimation. As the New Guess in the current iteration is affected by the previously updated ones, the sampled signals including transient responses would result in the New Guesses remotely deflecting from the real values, which would be considered in the next several iterations. As such, removing the transient responses of the sampled data through locally decimating the iteration actions during disturbance rejection could contribute to the moderate recursive curves of parameter estimation.

The CMA/CT reusing approach is secondly proposed for computation burden alleviation, by which the New Guess update, including 5 or 6 update steps in total, is partially conducted. It is noted that updating CMA/CT accounts for more than half of the computational complexity of applying the recursive algorithm. Therefore, re-using the CMA/CT over more than one iteration can reduce the computational efforts, as well as the iteration frequency reduction approach does. Before carrying out this approach, the magnitude difference between the two consecutive CMAs/CTs, and the difference between every other CMAs/CTs are both investigated, which are small enough for the adoption of the substitution. Compared with the iteration decimation approach which needs a careful balance between the convergence-time-related factor and the decimation factor, no adjustment is necessary in the CMA/CT reusing method, which may still save about two third of the computational costs spent in the conventional way. Particularly for the AP, there is almost no difference in computational efforts between the two proposed approaches.

The two approaches can also be combined and Chapter 4 shows how this can be managed, whilst taking stability issues and response to abrupt system variation into consideration. Overall, applying a combination of both methods, it is shown that the computational effort can

effectively be reduced to that observed in a single power converter system, whilst preserving the overall transient behaviour of all converters in a multi-rail structure.

In this project, the two approaches, including their combination, are successfully validated with simulation results by MATLAB/SIMULINK and experimental ones through a platform comprising a Texas Instruments TMS320F28335 DSP and a prototype of a three-rail power conversion architecture.

Another problem existing in System Identification (SI) of multi-rail power conversion architectures is about SI request ranking and responding. When there are multiple rails sending SI requests to the centralized single controller, an efficient request ranking and SI task allocation system is essential. Besides, this system should be able to make the most reasonable priority decisions under a comprehensive consideration. As such, a workflow including prioritization and SI task allocation is proposed in this work which can flexibly configure the respects taken into consideration. The features of the proposed prioritization approaches can be concluded as: 1. Being dynamic (The ranking update frequency equals the iteration frequency of recursive algorithms applied for SI). 2. Being comprehensive (Multiple respects are able to be readily considered into ranking rails in terms of being quantitated with weights). 3. Being selectively 'pre-emptive' (The SI process is divided into 'interruptible' stage and 'uninterruptible' stage to avoid rails with low priorities waiting unreasonably long time which could prolong the average pending time of all rails). 4. Being cost-saving (The intermediate results, corresponding to an interrupted SI process, would be saved instead of be thrown away, for being continuously used once the SI process recovers).

7.2 The Future Work

Both the two computation burden relief approaches and the prioritization workflow, proposed in this thesis, are trying to improve and simplify the process of parameter estimation of multiple buck converters. These methods can also be applied on system identification of any other electronic topologies as long as the topologies can be accurately described with linear models. Furthermore, rather than only being limited in parameter estimation of power converters, the three approaches may be validated in more occasions in which the algorithms (RLS, AP and KF) are employed in the future work.

The limits of the iteration decimation approach include the necessity of carefully tuning the convergence-time-related factors of the applied recursive algorithms, that requires fully investigations about the effects brought by changing the factor on estimation performances and, consequently, heavily restricts the application of the approaches. Therefore, the wider

approaches saving the computational complexity should be continuously proposed for parameter estimation of multi-rail power conversion structures.

Parameter estimation techniques is still mainly used for adaptive control so far, it would be interesting to make estimation results useful in other applications, e.g. health monitoring. Acquiring circuit components based on the transfer function coefficients of the converters is definitely a more efficient way compared with the current approaches to achieve so, and it would be a useful breakthrough in precise control and operation monitoring of converters.

Appendix A: Derivation Processes of RLS and AP Algorithms

RLS is a typical descent-based algorithm that starts from the regularized Newton's recursion. Therefore, $(R_{yu} - R_u\omega_{i-1})$ in (3-2) is still replaced by $u_i^*[y(i) - u_i\omega_{i-1}]$, while R_u is substituted with a better estimate, the exponentially weighted sample average.

$$\widehat{R_u} = \frac{1}{i+1}\sum_{j=0}^{i}\lambda^{i-j}u_j^*u_j \, , for\ some\ scalar\ 0 \ll \lambda \leq 1 \tag{A-1}$$

Assume that λ=1, the above expression for $\widehat{R_u}$ amounts to average all past regressors up to time i. Step-size is further assumed as follows:

$$\epsilon(i) = \lambda^{i+1}\frac{\epsilon}{i+1}, i \geq 0 \tag{A-2}$$

Therefore, the regularized Newton's recursion becomes

$$\omega_i = \omega_{i-1} + \left[\lambda^{i+1}\epsilon I + \sum_{j=0}^{i}\lambda^{i-j}u_j^*u_j\right]^{-1}u_i^*[y_i - u_i\omega_{i-1}] \tag{A-3}$$

According to [185],

$$\Phi_i \triangleq \left(\lambda^{i+1}\epsilon I + \sum_{j=0}^{i}\lambda^{i-j}u_j^*u_j\right) \tag{A-4}$$

It satisfies the recursion that $\Phi_i = \lambda\Phi_{i-1} + u_i^*u_i$, $\Phi_{-1} = \epsilon I$.

Then if $P_i = \Phi_i^{-1} = \lambda^{-1}\left[P_{i-1} - \frac{\lambda^{-1}P_{i-1}u_i^*u_iP_{i-1}}{1+\lambda^{-1}u_iP_{i-1}u_i^*}\right], P_{-1} = \epsilon^{-1}I$, RLS algorithm can be described as follows:

$$\omega_i = \omega_{i-1} + P_iu_i^*[y_i - u_i\omega_{i-1}], i \geq 0 \tag{A-5}$$

AP is another descent-based algorithm derived by replacing the required gradient vector, $(R_{yu} - R_u\omega_{i-1})$ in (3-2) [or (A-6)], and Hessian matrices by different suitable approximations.

$$\omega_i = \omega_{i-1} + \mu\left[R_{yu} - R_u\omega_{i-1}\right], \omega_{-1} = initial\ guess \tag{A-6}$$

For better performance, the regularized Newton's recursion with a constant regularization sequence, i.e.$\epsilon(i) = \epsilon$, and a constant step-size sequence, i.e.$\mu(i) = \mu$, are applied in (A-6), then it can be described as:

$$\omega_i = \omega_{i-1} + \mu[\epsilon I + R_u]^{-1}\left[R_{yu} - R_u\omega_{i-1}\right] \tag{A-7}$$

Instead of replacing R_u and R_{yu} with only current instantaneous values, several previous instantaneous values are required to calculate the mean of these instantaneous values, used for approximation in AP, i.e.,

$$\widehat{R_u} = \frac{1}{N}\left(\sum_{j=i-N+1}^{i}u_j^*u_j\right),\ \widehat{R_{yu}} = \frac{1}{N}\left(\sum_{j=i-N+1}^{i}y(j)u_j^*\right) \tag{A-8}$$

$\widehat{R_u} = \frac{1}{N}U_i^*U_i$, $\widehat{R_{yu}} = \frac{1}{N}U_i^*Y_i$, $U_i \triangleq \begin{bmatrix} u_i \\ u_{i-1} \\ \vdots \\ u_{i-N+1} \end{bmatrix}$, $Y_i \triangleq \begin{bmatrix} y(i) \\ y(i-1) \\ \vdots \\ y(i-N+1) \end{bmatrix}$, where the size of U_i is $N\times M$, of Y_i is $N\times 1$ and of ω $M\times 1$, therefore Newton's recursion becomes

$$\omega_i = \omega_{i-1} + \mu[\epsilon I + U_i{}^* U_i]^{-1} U_i{}^* [Y_i - U_i \omega_{i-1}] \quad \text{(A-9)}$$

Given that $[\epsilon I + u_i{}^* u_i]^{-1} u_i{}^* = u_i{}^* [\epsilon I + u_i u_i{}^*]^{-1}$ [185], AP is described as follows:

$$\omega_i = \omega_{i-1} + \mu U_i{}^* [\epsilon I + U_i U_i{}^*]^{-1} [Y_i - U_i \omega_{i-1}] \quad \text{(A-10)}$$

When $N = 1$, AP can be viewed as ϵ-NLMS. The larger N is (more historically sampled data would be considered in the current iteration cycle), the more accurate the approximation is. However, the computational complexity would increase.

Appendix B: Filter Selection and Test

To reduce electromagnetic interference in SMPCs, additional harmonic filters should be implemented prior to system identification procedure [242]. As the sampling frequency equals the switching frequency, ideally, the sampled output voltages without PRBS effects should be a constant which equal reference values.

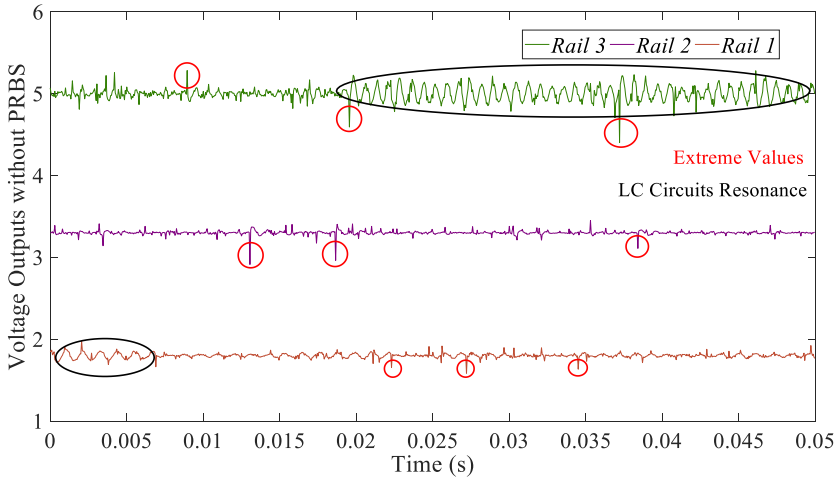
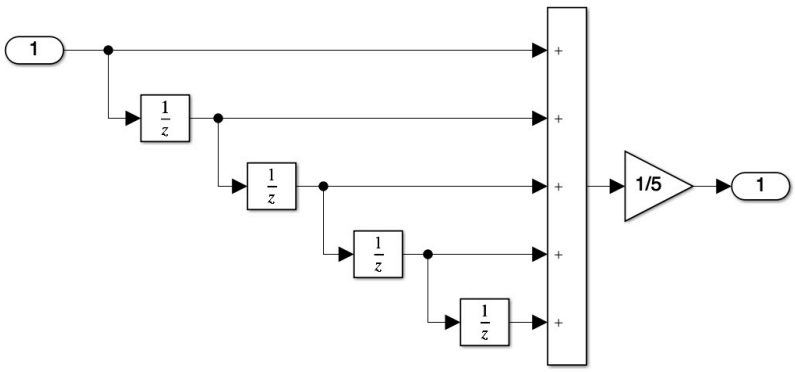
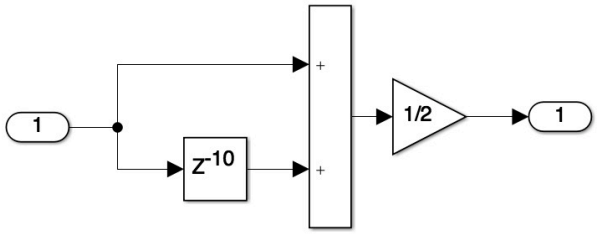


Figure Appendix.1 Two Main Noises in the Practically Sampled Voltage Outputs

Figure Appendix.1 shows the practically sampled voltage outputs of three rails, obviously not straight lines as results of two kinds of noises mainly. One is extreme values throughout the sampling process in three rails, which can be efficiently alleviated by a 5-tap moving average filter (MAF) (see Figure Appendix.2(a)). The other one is a kind of periodic noises that occasionally exists in Rail 1 and Rail 3, removed by Filter B shown in Figure Appendix.2(b). As the periodic noises are all about 1kHz, comprised of 20 samples per period with the sampling frequency of 20kHz, the currently sampled data, $x(i)$, should be numerically equal to the one which has been held for 10 sampling intervals, $x(i-10)$, but be different in signs, which means they can be well counteracted.

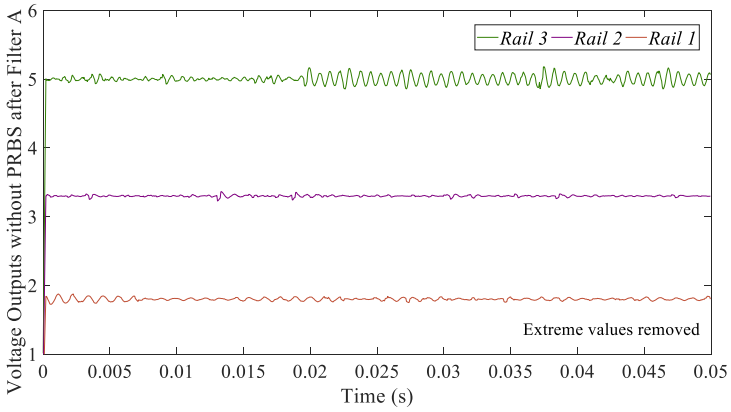


(a)

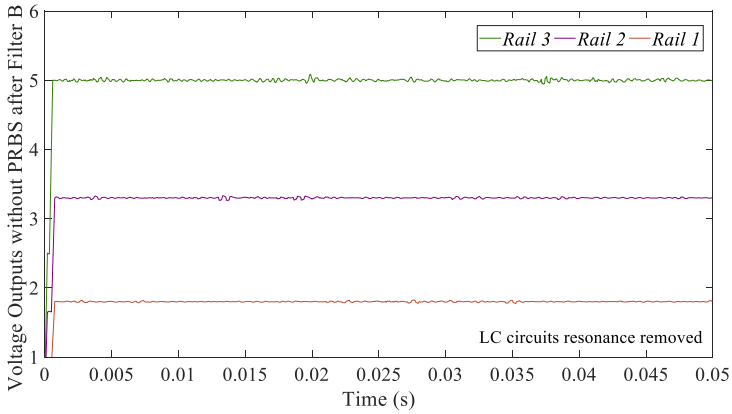


(b)

Figure Appendix.2 Filter A(a) and Filter B(b)



(a)



(b)

Figure Appendix.3 The Sampled Voltage Outputs Filtered by Filter A(a) and then by Filter B(b)

Figure Appendix.3(a) indicates that extreme values in voltage signals of three rails are effectively alleviated after going through Filter A, and after passing Filter B the 1kHz periodic noises are almost all removed (see Figure Appendix.3(b)), hence the sampled signals can be plotted as straight lines as shown in ideal situation.

It is reasonable to presume that the types of noises in sampled voltage signals would not change when PRBS is added into control loop, which means Filter A and B can be well applied

in system identification process. However, as of Figure Appendix.4, the outputs with PRBS feature that: 1: the magnitude of which is higher than that without PRBS injection, 2. a 1kHz periodic wave exists throughout the duration of PRBS injection. As the periodic noises which occasionally occur in Rail 1 and 3 (see Figure Appendix.1) have the same frequency, 1kHz, with the second feature of PRBS effects, applying Filter B to remove noises may also attenuate PRBS effects. Extreme values are filtered during signals passing Filter A (see Figure Appendix.5) while Filter B significantly removes the 1kHz waves, which are both noises and PRBS effects (see Figure Appendix.6). Therefore, Filter B would not be used for noise reduction in parameter estimation process as: 1. it would make PRBS injection in vain, 2. the occasionally occurred periodic noises would not significantly affect identification accuracy, proved by experiment results.

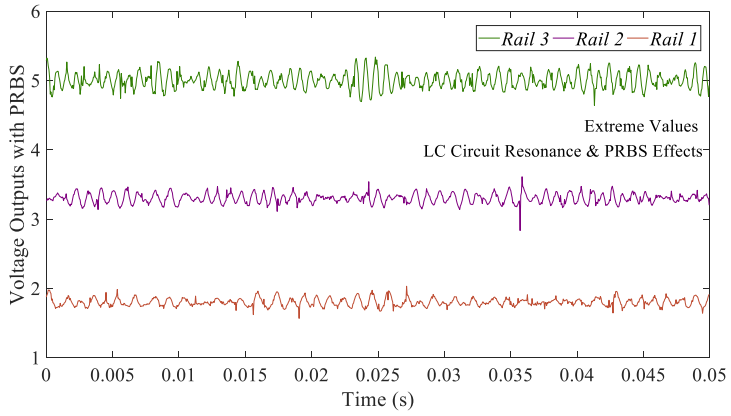


Figure Appendix.4 The output voltages with PRBS in three rails

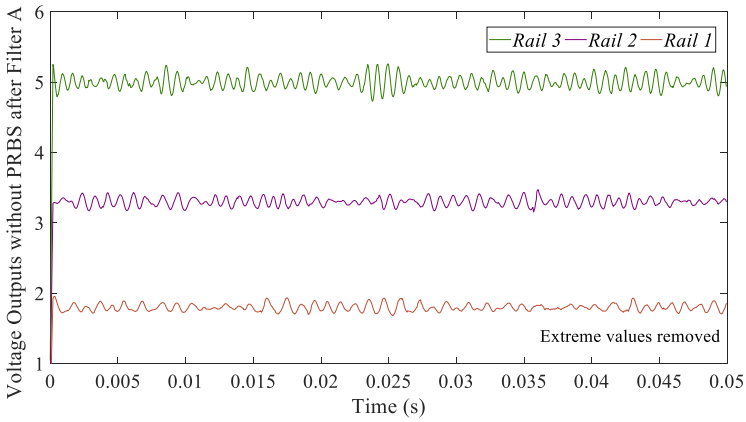


Figure Appendix.5 The output voltages with PRBS after passing Filter A

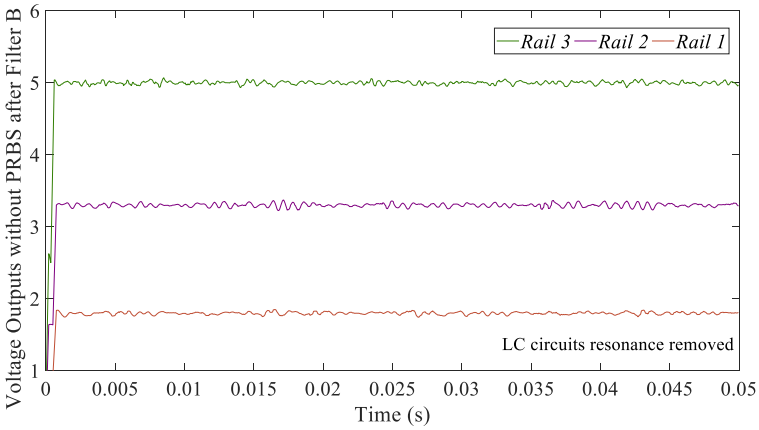


Figure Appendix.6 The output voltages with PRBS after passing Filter B

As the sampled voltage outputs include the afore-mentioned noises, control signals (duty cycle) would also contain extreme values and periodic waves corresponding to those in voltage feedback. To eliminate such noises without any attenuation of PRBS effects, duty cycle and PRBS signals can be separately sampled, then combined after duty cycle is filtered by Filter A and B. For simplifying sampling and filtering process in experiment configuration, however, only Filter A is applied for the control signal with PRBS, as estimation results of using the two filter ways are basically the same.

In summary, only MAF will be applied to filter sampled signals in experiment validation. In Figure Appendix.7, a 5-tap MAF is used for seeking its optimum working condition. Accordingly, when the loads vary from 4Ω to 6Ω, the MAF performs better than in the other load ranges. Appendix.8 shows percentage errors of estimation results of Rail 2 using a 5-tap MAF. Therefore, it is reasonable to presume that there will also be optimum load values individually corresponding to other two rails, and after the consequent investigation, the final filter setup in experiment for the multi-rail power conversion architecture is presented in Chapter 5.

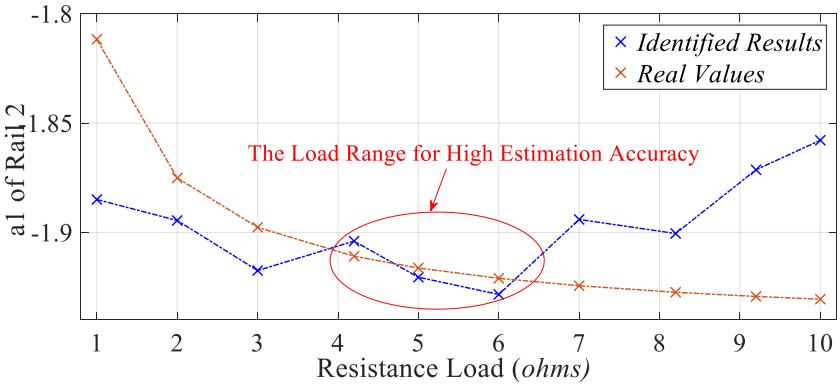


Figure Appendix.7 A 5-tap MAF Working under Different Load Values

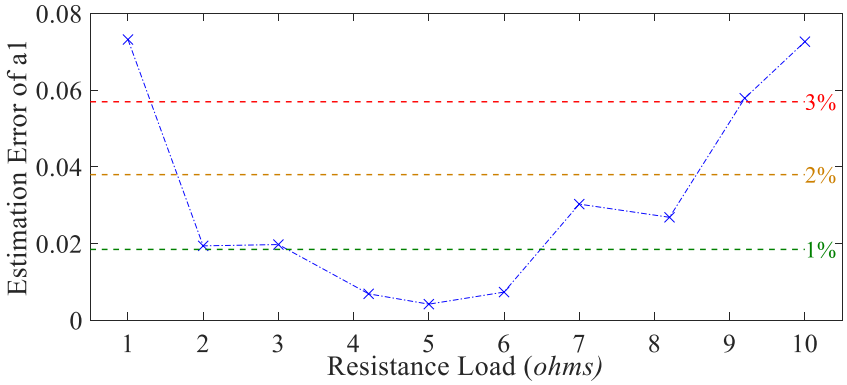


Figure Appendix.8 Percentage Errors of Estimation Results of Rail 2 using A 5-tap MAF

References

[1] R. Miftakhutdinov, L. Sheng, J. Liang, J. Wiggenhorn, and H. Huang, "Advanced control circuit for intermediate bus converter," in *2008 Twenty-Third Annual IEEE Applied Power Electronics Conference and Exposition*, 2008: IEEE, pp. 1515-1521.

[2] R. Miftakhutdinov, "Power distribution architecture for tele-and data communication system based on new generation intermediate bus converter," in *INTELEC 2008 - 2008 IEEE 30th International Telecommunications Energy Conference*, 2008, pp. 1-8.

[3] J. Brown, "Addressing the topologies, converters, and switching devices for intermediate bus architectures," in *2005 European Conference on Power Electronics and Applications*, 2005: IEEE, pp. 9 pp.-P. 9.

[4] R. V. White, "Emerging on-board power architectures," in *Eighteenth Annual IEEE Applied Power Electronics Conference and Exposition, 2003. APEC'03.*, 2003, vol. 2: IEEE, pp. 799-804.

[5] E. Pritchard, L. Mackey, D. Zhu, D. Gregory, and G. Norris, "Modular electric generator rapid deployment DC microgrid," in *2017 IEEE Second International Conference on DC Microgrids (ICDCM)*, 2017: IEEE, pp. 106-110.

[6] B. Aluisio, S. Bruno, L. De Bellis, M. Dicorato, G. Forte, and M. Trovato, "DC-microgrid operation planning for an electric vehicle supply infrastructure," *Applied Sciences,* vol. 9, no. 13, p. 2687, 2019.

[7] B. Haug. "72V Hybrid DC-to-DC Converter Reduces Intermediate Bus Converter Size by up to 50%." Power Electronics. https://www.powerelectronics.com/technologies/power-management/article/21864090/72v-hybrid-dcdc-reduces-intermediate-bus-converter-size-by-up-to-50 (accessed Jan. 29, 2018).

[8] S. Davis. "42V Quad Monolithic Synchronous Step-Down Regulator." Power Electronics. https://www.powerelectronics.com/technologies/regulators/article/21862796/42v-quad-monolithic-synchronous-stepdown-regulator (accessed Aug. 10, 2015).

[9] R.-J. Wai, Y.-F. Lin, and Y.-K. Liu, "Design of adaptive fuzzy-neural-network control for a single-stage boost inverter," *IEEE Transactions on Power Electronics,* vol. 30, no. 12, pp. 7282-7298, 2015.

[10] G. E. Pitel and P. T. Krein, "Real-time system identification for load monitoring and transient handling of Dc-Dc supplies," in *2008 IEEE Power Electronics Specialists Conference*, 2008: IEEE, pp. 3807-3813.

[11] Y. Yin *et al.*, "Advanced Control Strategies for Dc DC Buck Converters With Parametric Uncertainties via Experimental Evaluation," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 67, pp. 5257-5267, 2020.

[12] L. Liu, W. X. Zheng, and S. Ding, "An Adaptive SOSM Controller Design by Using a Sliding-Mode-Based Filter and its Application to Buck Converter," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 67, pp. 2409-2418, 2020.

[13] M. Jovanovic, "Power conversion technologies for computer, networking, and telecom power systems-past, present and future," in *International Power Conversion & Drive Conference (IPCDC) St. Petersburg, Russia*, 2011, pp. 156-161.

[14] G. A. Dumont and M. Huzmezan, "Concepts, methods and techniques in adaptive control," in *American Control Conference, 2002. Proceedings of the 2002*, 2002, vol. 2: IEEE, pp. 1137-1150.

[15] M. Al-Greer, M. Armstrong, M. Ahmeid, and D. Giaouris, "Advances on System Identification Techniques for DC–DC Switch Mode Power Converter Applications," *IEEE Transactions on Power Electronics,* vol. 34, no. 7, pp. 6973-6990, 2018.

[16] N. Beohar, V. N. Malladi, D. Mandal, S. Ozev, and B. Bakkaloglu, "Online Built-In Self-Test of High Switching Frequency DC DC Converters Using Model Reference Based System Identification Techniques," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 65, pp. 818-831, 2018.

[17] R.-G. Li and H. N. Wu, "Iterative Approach With Optimization-Based Execution Scheme for Parameter Identification of Distributed Parameter Systems and its Application in Secure Communication," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 67, pp. 3113-3126, 2020.

[18] F. Hafiz, A. Swain, E. M. Mendes, and L. A. Aguirre, "MultiObjective Evolutionary Approach to Grey-Box Identification of Buck Converter," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 67, no. 6, pp. 2016-2028, 2020.

[19] A. Pedross-Engel, H. Schumacher, and K. Witrisal, "Modeling and identification of ultra-wideband analog multipliers," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 65, no. 1, pp. 283-292, 2017.

[20] Y. Du, F. Liu, J. Qiu, and M. Buss, "A Semi-Supervised Learning Approach for Identification of Piecewise Affine Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 67, pp. 3521-3532, 2020.

[21] M. Shirazi, R. Zane, and D. Maksimovic, "An autotuning digital controller for DC–DC power converters based on online frequency-response measurement," *IEEE Transactions on Power Electronics,* vol. 24, no. 11, pp. 2578-2588, 2009.

[22] B. Miao, R. Zane, and D. Maksimovic, "System identification of power converters with digital control through cross-correlation methods," *IEEE Transactions on Power Electronics,* vol. 20, no. 5, pp. 1093-1099, 2005.

[23] T. Roinila, T. Helin, M. Vilkko, T. Suntio, and H. Koivisto, "Circular correlation based identification of switching power converter with uncertainty analysis using fuzzy density approach," *Simulation Modelling Practice and Theory,* vol. 17, no. 6, pp. 1043-1058, 2009.

[24] C. Wang, M. Armstrong, S. Gadoue, and P. Missailidis, "System identification of a DC-DC converter system using a Fast Affine Projection algorithm," in *7th IET International Conference on Power Electronics, Machines and Drives (PEMD 2014)*.

[25] B. Li and K. Low, "Low sampling rate online parameters monitoring of DC–DC converters for predictive-maintenance using biogeography-based optimization," *IEEE Transactions on Power Electronics,* vol. 31, no. 4, pp. 2870-2879, 2016.

[26] M. Kanzian, H. Gietler, C. Unterrieder, M. Agostinelli, M. Lunglmayr, and M. Huemer, "Low-Complexity State-Space-Based System Identification and Controller Auto-Tuning Method for Multi-Phase DC–DC Converters," *IEEE Transactions on Industry Applications,* vol. 55, no. 2, pp. 2076-2087, 2018.

[27] H. J. Tan, S. C. Chan, J. Q. Lin, and X. Sun, "A New Variable Forgetting Factor-Based Bias-Compensated RLS Algorithm for Identification of FIR Systems With Input Noise and Its Hardware Implementation," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 67, no. 1, pp. 198-211, 2019.

[28] M. Algreer, M. Armstrong, and D. Giaouris, "Active online system identification of switch mode DC–DC power converter based on efficient recursive DCD-IIR adaptive filter," *IEEE transactions on power electronics,* vol. 27, no. 11, pp. 4425-4435, 2012.

[29] Y. Zhang and Q. Xu, "Adaptive Sliding Mode Control With Parameter Estimation and Kalman Filter for Precision Motion Control of a Piezo-Driven Microgripper," *IEEE Transactions on Control Systems Technology,* vol. 25, no. 2, pp. 728-735, 2017.

[30] J. Liu, Y. V. Zakharov, and B. Weaver, "Architecture and FPGA design of dichotomous coordinate descent algorithms," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 56, no. 11, pp. 2425-2438, 2009.

[31] M. Ahmeid, M. Armstrong, M. Al-Greer, and S. Gadoue, "Computationally efficient self-tuning controller for dc–dc switch mode power converters based on partial update Kalman filter," *IEEE Transactions on Power Electronics,* vol. 33, no. 9, pp. 8081-8090, 2017.

[32] D. Baba. "Benefits of a multiphase buck converter." Power Management. https://www.ti.com/lit/an/slyt449/slyt449.pdf?ts=1615321484969&ref_url=https%253A%252F%252Fwww.google.com%252F. (accessed Mar, 2020).

[33] R. Miftakhutdinov, "Power distribution architecture for tele-and data communication system based on new generation intermediate bus converter," in *INTELEC 2008-2008 IEEE 30th International Telecommunications Energy Conference*, 2008: IEEE, pp. 1-8.

[34] F. C. Lee and Q. Li, "High-frequency integrated point-of-load converters: Overview," *IEEE Transactions on Power Electronics,* vol. 28, no. 9, pp. 4127-4136, 2013.

[35] X. Lyu *et al.*, "Composite Modular Power Delivery Architecture for Next-Gen 48V Data Center Applications," in *2018 1st Workshop on Wide Bandgap Power Devices and Applications in Asia (WiPDA Asia)*, 2018: IEEE, pp. 343-350.

[36] B. Haug. "72V Hybrid DC-to-DC Converter Reduces Intermediate Bus Converter Size by up to 50%." https://www.analog.com/media/en/analog-dialogue/volume-52/number-1/72-v-hybrid-dc-to-dc-reduces-intermediate-bus-converter-size.pdf (accessed Feb, 2018)).

[37] J. Brown and V. Siliconix, "Point of Load Converters—The Topologies Converters and Switching Devices Required for Efficient Conversion," in *PCIM conference*, 2002.

[38] L. Brush, "Distributed power architecture demand characteristics," in *Nineteenth Annual IEEE Applied Power Electronics Conference and Exposition, 2004. APEC'04.*, 2004, vol. 1: IEEE, pp. 342-345.

[39] E. R. Diaz, X. Su, M. Savaghebi, J. C. Vasquez, M. Han, and J. M. Guerrero, "Intelligent dc microgrid living laboratories-a chinese-danish cooperation project," in *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, 2015: IEEE, pp. 365-370.

[40] J. Mooney, M. Halton, and A. E. Mahdi, "Specialized digital signal processor for control of multi-rail/multi-phase high switching frequency power converters," in *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2010: IEEE, pp. 2207-2211.

[41] "TPS62770 Multi-Rail DC/DC Converter For Wearable Applications." TEXAS INSTRUMENTS. https://www.ti.com/document-viewer/TPS62770/datasheet/pin_configuration_and_functions - SLVSCX02188 (accessed Feb, 2021)).

[42] "TPS653850-Q1 Multirail Power Supply for Microcontrollers in Safety-Relevant Applications." TEXAS INSTRUMENTS. https://www.ti.com/lit/ds/symlink/tps65381-q1.pdf?ts=1590492873017 (accessed June, 2017).

[43] S. Yang, A. Bryant, P. Mawby, D. Xiang, L. Ran, and P. Tavner, "An industry-based survey of reliability in power electronic converters," *IEEE transactions on Industry Applications,* vol. 47, no. 3, pp. 1441-1451, 2011.

[44] P. Singh, "Power MOSFET failure mechanisms," in *INTELEC 2004. 26th Annual International Telecommunications Energy Conference*, 2004: IEEE, pp. 499-502.

[45] M. Ciappa, "Selected failure mechanisms of modern power modules," *Microelectronics reliability,* vol. 42, no. 4-5, pp. 653-667, 2002.

[46] E. Wu, E. Nowak, J. Aitken, W. Abadeer, L. Han, and S. Lo, "Structural dependence of dielectric breakdown in ultra-thin gate oxides and its relationship to soft breakdown modes and device failure," in *International Electron Devices Meeting 1998. Technical Digest (Cat. No. 98CH36217)*, 1998: IEEE, pp. 187-190.

[47] C. Chen, D. Labrousse, S. Lefebvre, M. Petit, C. Buttay, and H. Morel, "Study of short-circuit robustness of SiC MOSFETs, analysis of the failure modes and comparison with BJTs," *Microelectronics Reliability,* vol. 55, no. 9-10, pp. 1708-1713, 2015.
[48] R. Orsagh, D. Brown, M. Roemer, T. Dabnev, and A. Hess, "Prognostic health management for avionics system power supplies," in *2005 IEEE Aerospace Conference*, 2005: IEEE, pp. 3585-3591.
[49] V. Smet *et al.*, "Ageing and failure modes of IGBT modules in high-temperature power cycling," *IEEE transactions on industrial electronics,* vol. 58, no. 10, pp. 4931-4941, 2011.
[50] J.-G. Loquet, J.-P. David, S. Duzellier, D. Falguere, and T. Nuns, "Simulation of heavy-ion-induced failure modes in nMOS cells of ICs," *IEEE Transactions on Nuclear Science,* vol. 48, no. 6, pp. 2278-2284, 2001.
[51] J. R. Celaya, A. Saxena, P. Wysocki, S. Saha, and K. Goebel, "Towards prognostics of power MOSFETs: Accelerated aging and precursors of failure," NATIONAL AERONAUTICS AND SPACE ADMINISTRATION MOFFETT FIELD CA AMES RESEARCH …, 2010.
[52] S. Saha, J. R. Celaya, V. Vashchenko, S. Mahiuddin, and K. F. Goebel, "Accelerated aging with electrical overstress and prognostics for power MOSFETs," in *IEEE 2011 EnergyTech*, 2011: IEEE, pp. 1-6.
[53] J. R. Celaya, A. Saxena, C. S. Kulkarni, S. Saha, and K. Goebel, "Prognostics approach for power MOSFET under thermal-stress aging," in *2012 Proceedings Annual Reliability and Maintainability Symposium*, 2012: IEEE, pp. 1-6.
[54] M. Makdessi, A. Sari, P. Venet, P. Bevilacqua, and C. Joubert, "Accelerated ageing of metallized film capacitors under high ripple currents combined with a DC voltage," *IEEE Transactions on Power Electronics,* vol. 30, no. 5, pp. 2435-2444, 2014.
[55] K. Abdennadher, P. Venet, G. Rojat, J.-M. Rétif, and C. Rosset, "A real-time predictive-maintenance system of aluminum electrolytic capacitors used in uninterrupted power supplies," *IEEE Transactions on Industry Applications,* vol. 46, no. 4, pp. 1644-1652, 2010.
[56] H. Wang and F. Blaabjerg, "Reliability of capacitors for DC-link applications in power electronic converters—An overview," *IEEE Transactions on Industry Applications,* vol. 50, no. 5, pp. 3569-3578, 2014.
[57] K. Harada, A. Katsuki, and M. Fujiwara, "Use of ESR for deterioration diagnosis of electrolytic capacitor," *IEEE Transactions on Power Electronics,* vol. 8, no. 4, pp. 355-361, 1993.
[58] W. Minford, "Accelerated life testing and reliability of high K multilayer ceramic capacitors," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology,* vol. 5, no. 3, pp. 297-300, 1982.
[59] J. D. Prymak and J. Bergenthal, "Capacitance monitoring while flex testing," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part A,* vol. 18, no. 1, pp. 180-186, 1995.
[60] G. F. Franklin, M. Workman, and J. D. Powell, "Digital control of dynamic systems," 1980.
[61] W. Levine, "Control System Fundamentals," 1999.
[62] K. P. A. Packard, and R. Horowitz, "Dynamic systems and feedback class notes," Dept. Mech. Eng., Univ. California, Berkeley, CA, USA, Spring 2005.
[63] N. Shimkin, "Nonlinear Control Systems," 2008.
[64] Y. Liu, X. Wu, J. J. Zhu, and J. Lew, "Omni-directional mobile robot controller design by trajectory linearization," *Proceedings of the 2003 American Control Conference, 2003.,* vol. 4, pp. 3423-3428 vol.4, 2003.

[65] M. C. Mickle, R. Huang, and J. J. Zhu, "Unstable, nonminimum phase, nonlinear tracking by trajectory linearization control," *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.,* vol. 1, pp. 812-818 Vol.1, 2004.
[66] G. W. Wester and R. D. Middlebrook, "Low-Frequency Characterization of Switched dc-dc Converters," *IEEE Transactions on Aerospace and Electronic Systems,* vol. AES-9, pp. 376-385, 1973.
[67] R. Middlebrook and S. Cuk, "A general unified approach to modelling switching-converter power stages," *1970 IEEE Power Electronics Specialists Conference,* pp. 18-34, 1976.
[68] R. Middlebrook, "Small-signal modeling of pulse-width modulated switched-mode power converters," 1988.
[69] X. Yue, X. Wang, and F. Blaabjerg, "Review of Small-Signal Modeling Methods Including Frequency-Coupling Dynamics of Power Converters," *IEEE Transactions on Power Electronics,* vol. 34, pp. 3313-3328, 2019.
[70] Y. Qiu, M. Xu, K. Yao, J. Sun, and F. Lee, "The multi-frequency small-signal model for buck and multiphase interleaving buck converters," *Twentieth Annual IEEE Applied Power Electronics Conference and Exposition, 2005. APEC 2005.,* vol. 1, pp. 392-398 Vol. 1, 2005.
[71] Y. Qiu, M. Xu, J. Sun, and F. Lee, "A Generic High-Frequency Model for the Nonlinearities in Buck Converters," *IEEE Transactions on Power Electronics,* vol. 22, pp. 1970-1977, 2007.
[72] R. Middlebrook, "Modeling current-programmed buck and boost regulators," *IEEE Transactions on Power Electronics,* vol. 4, pp. 36-52, 1989.
[73] F. Lee, Y. Yu, and M. Mahmoud, "A unified analysis and design procedure for a standardized control module for dc-dc switching regulators," *1980 IEEE Power Electronics Specialists Conference,* pp. 284-301, 1980.
[74] R. Middlebrook, "Topics in Multiple-Loop Regulators and Current-Mode Programming," *IEEE Transactions on Power Electronics,* vol. PE-2, pp. 109-124, 1987.
[75] A. E. Aroudi, M. Debbat, R. Giral, G. Olivar, L. Benadero, and E. Toribio, "Bifurcations in Dc-dc Switching converters: Review of Methods and Applications," *Int. J. Bifurc. Chaos,* vol. 15, pp. 1549-1578, 2005.
[76] R. Erickson and D. Maksimovj "Fundamentals of Power Electronics," 1997.
[77] S. Banerjee and G. Verghese, "Nonlinear Phenomena in Power Electronics," 2001.
[78] C. Tse and M. D. Bernardo, "Complex behavior in switching power converters," 2002.
[79] J. Li and F. Lee, "New Modeling Approach and Equivalent Circuit Representation for Current-Mode Control," *IEEE Transactions on Power Electronics,* vol. 25, pp. 1218-1230, 2010.
[80] J. Sun, Y. Qiu, M. Xu, and F. Lee, "High-Frequency Dynamic Current Sharing Analyses for Multiphase Buck VRs," *IEEE Transactions on Power Electronics,* vol. 22, pp. 2424-2431, 2007.
[81] X. Yue, D. Boroyevich, F. Lee, F. Chen, R. Burgos, and F. Zhuo, "Beat Frequency Oscillation Analysis for Power Electronic Converters in DC Nanogrid Based on Crossed Frequency Output Impedance Matrix Model," *IEEE Transactions on Power Electronics,* vol. 33, pp. 3052-3064, 2018.
[82] X. Yue, F. Zhuo, S. Yang, Y. Pei, and H. Yi, "A Matrix-Based Multifrequency Output Impedance Model for Beat Frequency Oscillation Analysis in Distributed Power Systems," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 4, pp. 80-92, 2016.

[83] X. Yue, Y. Zhu, S. Yang, Y. Chen, F. Zhuo, and Y. Pei, "A modulation and sampling based modeling method for the nonlinearities of power converters and its application analysis," *2015 17th European Conference on Power Electronics and Applications (EPE'15 ECCE-Europe),* pp. 1-10, 2015.
[84] N. Wereley, "Analysis and control of linear periodically time varying systems," 1990.
[85] N. Wereley and S. R. Hall, "Linear Time Periodic Systems: Transfer Function, Poles, Transmission Zeroes and Directional Properties," *1991 American Control Conference,* pp. 1179-1184, 1991.
[86] N. M. Wereley and S. R. Hall, "Frequency response of linear time periodic systems," *29th IEEE Conference on Decision and Control,* pp. 3650-3655 vol.6, 1990.
[87] X. Yue, X. Wang, F. Blaabjerg, D. Boroyevich, R. Burgos, and F. Lee, "Wideband small-signal input dq admittance modeling of six-pulse diode rectifiers," *2018 IEEE Applied Power Electronics Conference and Exposition (APEC),* pp. 1981-1988, 2018.
[88] X. Yue, D. Boroyevich, R. Burgos, and F. Zhuo, "Modeling and analysis for input characteristics of line-frequency rectifiers," *2016 IEEE Energy Conversion Congress and Exposition (ECCE),* pp. 1-8, 2016.
[89] J. Sun and H. Liu, "Sequence Impedance Modeling of Modular Multilevel Converters," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 5, pp. 1427-1443, 2017.
[90] K. Guesmi and A. Hamzaoui, "On the modelling of DC-DC converters: An enhanced approach," *International Journal of Numerical Modelling-electronic Networks Devices and Fields,* vol. 24, pp. 36-57, 2011.
[91] A. Khoudiri, K. Guesmi, and D. Mahi, "Spectral decomposition based approach for DC–DC converters modeling," *International Journal of Electrical Power & Energy Systems,* vol. 61, pp. 288-297, 2014.
[92] A. Khoudiri, K. Guesmi, and D. Mahi, "DC-DC converters modelling approach using spectral decomposition theorem," *IFAC Proceedings Volumes,* vol. 45, no. 21, pp. 548-553, 2012.
[93] A. Khoudiri, K. Guesmi, and D. Mahi, "DC-DC Converters Modelling Approach Using Spectral Decomposition Theorem," *IFAC Proceedings Volumes,* vol. 45, pp. 548-553, 2012.
[94] S. Tian, F. Lee, Q. Li, and Y. Yan, "Unified Equivalent Circuit Model and Optimal Design of V^{2} Controlled Buck Converters," *IEEE Transactions on Power Electronics,* vol. 31, pp. 1734-1744, 2016.
[95] Y. Yan, F. Lee, P. Mattavelli, and S. Tian, "Small Signal Analysis of V2 Control Using Equivalent Circuit Model of Current Mode Controls," *IEEE Transactions on Power Electronics,* vol. 31, pp. 5344-5353, 2016.
[96] Y. Yan, F. Lee, and P. Mattavelli, "Unified Three-Terminal Switch Model for Current Mode Controls," *IEEE Transactions on Power Electronics,* vol. 27, pp. 4060-4070, 2012.
[97] M. M. Peretz and S. Ben-Yaakov, "Time Domain Identification of PWM Converters for Digital Controllers Design," *2007 IEEE Power Electronics Specialists Conference,* pp. 809-813, 2007.
[98] B. Hutchings. "SMPS Buck Converter Design Example." https://www.microchip.com/stellent/groups/SiteComm_sg/documents/Training_Tutorials/en528032.pdf (accessed Feb, 2020)).
[99] S. Manurkar, "Controller Design for Buck Converter Step-by-Step Approach," 2014.
[100] M. Hallworth and S. Shirsavar, "Microcontroller-Based Peak Current Mode Control Using Digital Slope Compensation," *IEEE Transactions on Power Electronics,* vol. 27, pp. 3340-3351, 2012.

[101] A. I. Bogdan and N. Bizon, "Voltage-mode control of the DC-DC power converter - a short review," *2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI),* pp. E-27-E-32, 2015.
[102] S. Kapat and P. Krein, "A Tutorial and Review Discussion of Modulation, Control and Tuning of High-Performance DC-DC Converters Based on Small-Signal and Large-Signal Approaches," *IEEE Open Journal of Power Electronics,* vol. 1, pp. 339-371, 2020.
[103] J. Chow and P. Kokotovi "A decomposition of near-optimum regulators for systems with slow and fast modes," 1976.
[104] S. Busquets-monge *et al.*, "Power converter design optimization," *IEEE Industry Applications Magazine,* vol. 10, pp. 32-38, 2004.
[105] A. Kislovski, R. Redl, and N. Sokal, "Dynamic analysis of switching-mode DC/DC converters," 1991.
[106] L. Corradini, A. Bjeletif , R. Zane, and D. Maksimovj "Fully digital hysteretic modulator for DC-DC switching converters," *2009 IEEE Energy Conversion Congress and Exposition,* pp. 3312-3319, 2009.
[107] V. Raviraj and P. Sen, "Comparative study of proportional-integral, sliding mode and fuzzy logic controllers for power converters," *IAS '95. Conference Record of the 1995 IEEE Industry Applications Conference Thirtieth IAS Annual Meeting,* vol. 2, pp. 1651-1655 vol.2, 1995.
[108] G. Escobar, R. Ortega, H. Sira-RamÌrez, J. Vilain, and I. Zein, "An experimental comparison of several nonlinear controllers for power converters," *IEEE Control Systems Magazine,* vol. 19, pp. 66-82, 1999.
[109] A. R. Nikhar, S. Apte, and R. Somalwar, "Review of various control techniques for DC-DC interleaved boost converters," *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC),* pp. 432-437, 2016.
[110] Y. Alsmadi, V. Utkin, M. A. Haj-ahmed, and L. Xu, "Sliding mode control of power converters: DC/DC converters," *International Journal of Control,* vol. 91, pp. 2472 - 2493, 2018.
[111] D. Q. Zhang and S. Panda, "Chattering-free and fast-response sliding mode controller," 1999.
[112] B. M. David, "A Review of Sliding Mode Control Of DC-DC Converters," 2015.
[113] J. Ni, L. Liu, C. Liu, and X. Hu, "Chattering-Free Time Scale Separation Sliding Mode Control Design with Application to Power System Chaos Suppression," *Mathematical Problems in Engineering,* vol. 2016, pp. 1-14, 2016.
[114] J. M. Carrasco, J. Quero, F. P. Ridao, M. Perales, and L. Franquelo, "Sliding mode control of a DC/DC PWM converter with PFC implemented by neural networks," *IEEE Transactions on Circuits and Systems I-regular Papers,* vol. 44, pp. 743-749, 1997.
[115] J. V. Sawant and D. Ginoya, "dSPACE DS1104 based Proportional Integral Sliding Mode Controller for Continuous Conduction Mode Buck Boost Converter," 2012.
[116] S.-C. Tan, Y. M. Lai, M. K. H. Cheung, and C. K. Tse, "On the practical design of a sliding mode voltage controlled buck converter," *IEEE Transactions on Power Electronics,* vol. 20, pp. 425-437, 2005.
[117] A. Ghosh, M. Prakash, S. Pradhan, and S. Banerjee, "A comparison among PID, Sliding Mode and internal model control for a buck converter," *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society,* pp. 1001-1006, 2014.
[118] Q. Shui-sheng, "Analysis and Experimental Study of Proportional-Integral Sliding Mode Control for DC/DC Converter," 2005.

[119] P. Juli·n, A. Oliva, P. Mandolesi, and H. Chiacchiarini, "Output discrete feedback control of a DC-DC buck converter," *ISIE '97 Proceeding of the IEEE International Symposium on Industrial Electronics,* pp. 925-930 vol.3, 1997.
[120] H. Sira-RamÌrez, R. Perez-Moreno, R. Ortega, and M. Garcia-Esteban, "Passivity-based controllers for the stabilization of Dc-to-Dc Power converters," *Autom.,* vol. 33, pp. 499-513, 1997.
[121] H. Sira-RamÌrez, "Sliding motions in bilinear switched networks," 1987.
[122] H. Sira-RamÌrez, G. Escobar, and R. Ortega, "On passivity-based sliding mode control of switched DC-to-DC power converters," *Proceedings of 35th IEEE Conference on Decision and Control,* vol. 3, pp. 2525-2526 vol.3, 1996.
[123] M. G. Shankar and P. R. R. ME, "DESIGN FPGA BASED MULTI RAIL DC-DC CONVERTER AND MAC ARCHITECTURE BY USING MSP 430 PROCESSOR," 2014.
[124] K.-Y. Lee, C.-A. Yeh, and Y.-S. Lai, "Design and implementation of fully digital controller for non-isolated-point-of-load converter with high current slew rate," in *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, 2006: IEEE, pp. 2605-2610.
[125] R. Priewasser, M. Agostinelli, S. Marsili, and M. Huemer, "Digital controller design for point-of-load DC-DC converters with variable switching frequency," *Electronics letters,* vol. 47, no. 6, pp. 374-375, 2011.
[126] Y. Yan, F. C. Lee, and P. Mattavelli, "Dynamic performance comparison of current mode control schemes for Point-of-Load Buck converter application," in *2012 Twenty-Seventh Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2012: IEEE, pp. 2484-2491.
[127] D. S. Padimiti and M. Ferdowsi, "Review of digital control techniques for automotive DC-DC converters," *2005 IEEE Vehicle Power and Propulsion Conference,* pp. 5 pp.-, 2005.
[128] J. Fiser and P. ZÌtek, "PID Controller Tuning via Dominant Pole Placement in Comparison with Ziegler-Nichols Tuning," *IFAC-PapersOnLine,* vol. 52, pp. 43-48, 2019.
[129] K. ≈str^m and T. H‰gglund, "New Tuning Methods for PID Controllers," 1995.
[130] D. Rivera, M. Morari, and S. Skogestad, "Internal model control: PID controller design," *Industrial & Engineering Chemistry Process Design and Development,* vol. 25, pp. 252-265, 1986.
[131] M. Campi, A. Lecchini, and S. Savaresi, "Virtual reference feedback tuning (VRFT): a new direct approach to the design of feedback controllers," *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187),* vol. 1, pp. 623-629 vol.1, 2000.
[132] Z. Wei and Y. Ma-ying, "Comparison of auto-tuning methods of PID controllers based on models and closed-loop data," *Proceedings of the 33rd Chinese Control Conference,* pp. 3661-3667, 2014.
[133] M. M. F. S. Algreer, "Microprocessor based signal processing techniques for system identification and adaptive control of DC-DC converters," Thesis (Ph. D.)--University of Newcastle upon Tyne, 2012., Newcastle upon Tyne, 2012.
[134] M. M. Peretz and S. Ben-Yaakov, "Time-domain identification of pulse-width modulated converters," *Iet Power Electronics,* vol. 5, pp. 166-172, 2012.
[135] B. Miao, R. Zane, and D. Maksimovj "Automated Digital Controller Design for Switching Converters," *2005 IEEE 36th Power Electronics Specialists Conference,* pp. 2729-2735, 2005.

[136] Y. Liu, E. Meyer, and X. Liu, "Recent Developments in Digital Control Strategies for DC/DC Switching Power Converters," *IEEE Transactions on Power Electronics,* vol. 24, pp. 2567-2577, 2009.
[137] H. Al-Atrash and I. Batarseh, "Digital controller design for a practicing power electronics engineer," *APEC 07 - Twenty-Second Annual IEEE Applied Power Electronics Conference and Exposition,* pp. 34-41, 2007.
[138] H. J. Y. Duan, "Digital controller design for switchmode power converters," in *APEC '99. Fourteenth Annual Applied Power Electronics Conference and Exposition. 1999 Conference Proceedings (Cat. No.99CH36285)*, Dallas, TX, USA, vol. 2, pp. 967-973, doi: 10.1109/APEC.1999.750486.
[139] S. E. Beid, S. Doubabi, and M. Chaoui, "Adaptive control of PWM Dc-to-Dc converters operating in continuous conduction mode," *2007 Mediterranean Conference on Control & Automation,* pp. 1-5, 2007.
[140] V. V. Chalam, "Adaptive Control Systems: Techniques and Applications," 1987.
[141] M. Algreer, "Microprocessor based signal processing techniques for system identification and adaptive control of DC-DC converters," 2012.
[142] M. M. Peretz and S. Ben-Yaakov, "Time-Domain Design of Digital Compensators for PWM DC-DC Converters," *IEEE Transactions on Power Electronics,* vol. 27, pp. 284-293, 2012.
[143] V. Bob·l, J. Bˆhm, J. Fessl, and J. Mach, ek, "Digital Self-tuning Controllers: Algorithms, Implementation and Applications," 2005.
[144] A. Kelly and K. Rinne, "Control of dc-dc converters by direct pole placement and adaptive feedforward gain adjustment," *Twentieth Annual IEEE Applied Power Electronics Conference and Exposition, 2005. APEC 2005.,* vol. 3, pp. 1970-1975 Vol. 3, 2005.
[145] G. Cimini, G. Ippoliti, S. Longhi, G. Orlando, and M. Pirro, "Synchronous buck converter control via robust periodic pole assignment," *2014 European Control Conference (ECC),* pp. 1921-1926, 2014.
[146] A. Prodic, D. Maksimovj , and R. Erickson, "Design and implementation of a digital PWM controller for a high-frequency switching DC-DC power converter," *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.37243),* vol. 2, pp. 893-898 vol.2, 2001.
[147] Z. Zhao and A. Prodic, "Limit-Cycle Oscillations Based Auto-Tuning System for Digitally Controlled DC DC Power Supplies," *IEEE Transactions on Power Electronics,* vol. 22, pp. 2211-2222, 2007.
[148] V. Yousefzadeh, N. Wang, Z. Popovj , and D. Maksimovj "A digitally controlled DC/DC converter for an RF power amplifier," *IEEE Transactions on Power Electronics,* vol. 21, pp. 164-172, 2006.
[149] A. Prodic and D. Maksimovi "Design of a digital PID regulator based on look-up tables for control of high-frequency DC-DC converters," *2002 IEEE Workshop on Computers in Power Electronics, 2002. Proceedings.,* pp. 18-22, 2002.
[150] Y. Niu, Y. Gao, and S. Guo, "A Dual-mode digital controller for switching mode power supply," *2011 4th International Conference on Power Electronics Systems and Applications,* pp. 1-7, 2011.
[151] S. Guo, Y. Gao, Y. Xu, X. Lin-Shi, and B. Allard, "Digital PWM controller for high-frequency low-power DC-DC switching mode power supply," *2009 IEEE 6th International Power Electronics and Motion Control Conference,* pp. 1340-1346, 2009.
[152] X. Lin-Shi, F. Morel, B. Allard, D. Tournier, J. Retif, and S. Guo, "A Digital-Controller Parameter-Tuning Approach, Application to a Switch-Mode Power Supply," *2007 IEEE International Symposium on Industrial Electronics,* pp. 3356-3361, 2007.

[153] R. Janga and S. Malaji, "Digitally controlled Active Clamp Forward Converter with small signal discrete-time modeling," *2014 International Conference on Computer Communication and Informatics,* pp. 1-6, 2014.
[154] S. Abe, M. Ogawa, T. Zaitsu, S. Obata, M. Shoyama, and T. Ninomiya, "Power-stage frequency response cancellation of DC-DC converter with digital control," *SPEEDAM 2010,* pp. 44-49, 2010.
[155] K. Cave-Ayland, V. Becerra, B. Potter, and S. Shirsavar, "Choosing the most suitable analogue redesign method for forward-type digital power converters," *2012 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES),* pp. 1-6, 2012.
[156] M. Hagen and V. Yousefzadeh, "Applying Digital Technology to PWM Control-Loop Designs," 2008.
[157] Z. Zhao, "Design and Practical Implementation of Digital Auto-tuning and Fast-response Controllers for Low-power Switch-mode Power Supplies," 2008.
[158] S. Padhee, U. C. Pati, and K. Mahapatra, "Modelling switched mode DC-DC converter using system identification techniques: A review," *2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS),* pp. 1-6, 2016.
[159] A. Congiu, E. Bodano, and M. Barbaro, "A $\Delta \Sigma$ Dithering-Amplification-Based Identification Technique for Online SMPS," *IEEE Transactions on Power Electronics,* vol. 31, pp. 5992-6001, 2016.
[160] N. Beohar, V. N. Malladi, D. Mandal, S. Ozev, and B. Bakkaloglu, "Online built-in self-test of high switching frequency dc–dc converters using model reference based system identification techniques," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 65, no. 2, pp. 818-831, 2017.
[161] M. Bhardwaj, S. Choudhury, R. Poley, and B. Akin, "Online Frequency Response Analysis: A Powerful Plug-in Tool for Compensation Design and Health Assessment of Digitally Controlled Power Converters," *IEEE Transactions on Industry Applications,* vol. 52, pp. 2426-2435, 2016.
[162] M. Shirazi, J. Morroni, A. Dolgov, R. Zane, and D. Maksimovj "Integration of Frequency Response Measurement Capabilities in Digital Controllers for Dc DC Converters," *IEEE Transactions on Power Electronics,* vol. 23, pp. 2524-2535, 2008.
[163] P. Kshirsagar, D. Jiang, and Z. Zhang, "Implementation and evaluation of online system identification of electromechanical systems using adaptive filters," *2014 IEEE Energy Conversion Congress and Exposition (ECCE),* pp. 2477-2484, 2014.
[164] S. Dusmez, M. Bhardwaj, L. Sun, and B. Akin, "A software frequency response analysis method to monitor degradation of power MOSFETs in basic single-switch converters," *2016 IEEE Applied Power Electronics Conference and Exposition (APEC),* pp. 505-510, 2016.
[165] J. Morroni, A. Dolgov, M. Shirazi, R. Zane, and D. Maksimovi "Online Health Monitoring in Digitally Controlled Power Converters," *2007 IEEE Power Electronics Specialists Conference,* pp. 112-118, 2007.
[166] M. Shirazi, R. Zane, and D. Maksimovi "An Autotuning Digital Controller for Dc DC Power Converters Based on Online Frequency-Response Measurement," *IEEE Transactions on Power Electronics,* vol. 24, pp. 2578-2588, 2009.
[167] A. Barkley, R. Dougal, and E. Santi, "Adaptive control of power converters using Digital Network Analyzer Techniques," *2011 Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC),* pp. 1824-1832, 2011.
[168] A. Barkley and E. Santi, "Improved Online Identification of a DC DC Converter and Its Control Loop Gain Using Cross-Correlation Methods," *IEEE Transactions on Power Electronics,* vol. 24, pp. 2021-2031, 2009.

[169] D. Martin, E. Santi, and A. Barkley, "Wide bandwidth system identification of AC system impedances by applying pertubations to an existing converter," *2011 IEEE Energy Conversion Congress and Exposition,* pp. 2549-2556, 2011.
[170] T. Roinila, M. Vilkko, and T. Suntio, "Frequency-Response Measurement of Switched-Mode Power Supplies in the Presence of Nonlinear Distortions," *IEEE Transactions on Power Electronics,* vol. 25, pp. 2179-2187, 2010.
[171] T. Roinila, M. Vilkko, and T. Suntio, "Fast Loop Gain Measurement of a Switched-Mode Converter Using a Binary Signal With a Specified Fourier Amplitude Spectrum," *IEEE Transactions on Power Electronics,* vol. 24, pp. 2746-2755, 2009.
[172] J. Liu, C.-H. Yang, and C. Tsai, "Correlation-based system identification of digitally controlled SMPS," *2011 IEEE Ninth International Conference on Power Electronics and Drive Systems,* pp. 1149-1152, 2011.
[173] N. Beohar *et al.*, "Disturbance-free BIST for loop characterization of DC-DC buck converters," *2015 IEEE 33rd VLSI Test Symposium (VTS),* pp. 1-6, 2015.
[174] A. Davoudi, N. Kong, H. Behjati, M. D. Hagen, and E. Oettinger, "Automated system identification and controller tuning for digitally controlled dc-dc converters," *Iet Power Electronics,* vol. 5, pp. 765-772, 2012.
[175] A. Fujimori and S. Ohara, "Parameter identification of continuous-time systems using iterative learning control," *International Journal of Control, Automation and Systems,* vol. 9, pp. 203-210, 2011.
[176] L. Corradini, P. Mattavelli, and D. Maksimovic, "Robust Relay-Feedback Based Autotuning for DC-DC Converters," *2007 IEEE Power Electronics Specialists Conference,* pp. 2196-2202, 2007.
[177] W. Stefanutti, P. Mattavelli, S. Saggini, and M. Ghioni, "Autotuning of Digitally Controlled Dc DC Converters Based on Relay Feedback," *IEEE Transactions on Power Electronics,* vol. 22, pp. 199-207, 2007.
[178] L. Corradini, P. Mattavelli, W. Stefanutti, and S. Saggini, "Simplified Model Reference-Based Autotuningfor Digitally Controlled SMPS," *IEEE Transactions on Power Electronics,* vol. 23, pp. 1956-1963, 2008.
[179] C. Wang, M. Armstrong, S. Gadoue, and P. Missailidis, "System identification of a DC-DC converter system using a Fast Affine Projection algorithm," 2014.
[180] P. Nagarjuna and M. Ramireddy, "SYSTEM IDENTIFICATION USING AN AFFINE COMBINATION OF TWO LMS ADAPTIVE FILTERS," 2012.
[181] K. Rasadia and K. Parmar, "Adaptive Filter Analysis for System Identification Using Various Adaptive Algorithms," *International journal of engineering research and technology,* vol. 1, 2012.
[182] B. Johansson and M. Lenells, "Possibilities of obtaining small-signal models of DC-to-DC power converters by means of system identification," *INTELEC. Twenty-Second International Telecommunications Energy Conference (Cat. No.00CH37131),* pp. 65-75, 2000.
[183] A. Simpkins, "System Identification: Theory for the User, 2nd Edition (Ljung, L.; 1999) [On the Shelf]," *IEEE Robotics & Automation Magazine,* vol. 19, pp. 95-96, 2012.
[184] H. Gietler, C. Unterrieder, A. Berger, R. Priewasser, and M. Lunglmayr, "Low-complexity, high frequency parametric system identification method for switched-mode power converters," in *Applied Power Electronics Conference and Exposition (APEC), 2017 IEEE*, 2017: IEEE, pp. 2004-2009.
[185] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.
[186] S. A. Ghauri and M. Sohail, "System identification using LMS, NLMS and RLS," *2013 IEEE Student Conference on Research and Developement,* pp. 65-69, 2013.

[187] C. Wang, M. Armstrong, and S. Gadoue, "System identification and adaptive control of a DC-DC converter using a current balancing ON/OFF control technique for optimal transient performance," in *Power Electronics and Applications (EPE'15 ECCE-Europe), 2015 17th European Conference on*, 2015: IEEE, pp. 1-10.
[188] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman Filter and Its Application," *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS),* pp. 74-77, 2015.
[189] M. Ahmeid, M. Armstrong, S. Gadoue, M. Al-Greer, and P. Missailidis, "Real-Time Parameter Estimation of DC–DC Converters Using a Self-Tuned Kalman Filter," *IEEE Transactions on Power Electronics,* vol. 32, no. 7, pp. 5666-5674, 2017.
[190] L. Cao and H. Schwartz, "Analysis of the Kalman filter based estimation algorithm: an orthogonal decomposition approach," *Autom.,* vol. 40, pp. 5-19, 2004.
[191] A. Barkley and E. Santi, "Online Monitoring of Network Impedances Using Digital Network Analyzer Techniques," *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition,* pp. 440-446, 2009.
[192] K. Abdennadher, P. Venet, G. Rojat, J. Retif, and C. Rosset, "A Real-Time Predictive-Maintenance System of Aluminum Electrolytic Capacitors Used in Uninterrupted Power Supplies," *IEEE Transactions on Industry Applications,* vol. 46, pp. 1644-1652, 2010.
[193] M. Vogelsberger, T. Wiesinger, and H. Ertl, "Life-Cycle Monitoring and Voltage-Managing Unit for DC-Link Electrolytic Capacitors in PWM Converters," *IEEE Transactions on Power Electronics,* vol. 26, pp. 493-503, 2011.
[194] A. Amaral and A. Cardoso, "On-line fault detection of aluminium electrolytic capacitors, in step-down DC-DC converters, using input current and output voltage ripple," *Iet Power Electronics,* vol. 5, pp. 315-322, 2012.
[195] G. M. Buiatti, J. Martin-Ramos, C. R. Garcia, A. Amaral, and A. Cardoso, "An Online and Noninvasive Technique for the Condition Monitoring of Capacitors in Boost Converters," *IEEE Transactions on Instrumentation and Measurement,* vol. 59, pp. 2134-2143, 2010.
[196] A. Radic, A. Straka, and A. Prodic, "Synchronized Zero-Crossing-Based Self-Tuning Capacitor Time-Constant Estimator for Low-Power Digitally Controlled DC DC Converters," *IEEE Transactions on Power Electronics,* vol. 29, pp. 5106-5110, 2014.
[197] H. Ma, X. Mao, N. Zhang, and D. Xu, "Parameter Identification of Power Electronic Circuits Based on Hybrid Model," *2005 IEEE 36th Power Electronics Specialists Conference,* pp. 2855-2860, 2005.
[198] Y. Zhao, B. Huang, H. Su, and J. Chu, "Output error method for identification of multiple-model process with transition," *2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP),* pp. 72-77, 2011.
[199] F. Alonge, F. D'Ippolito, F. M. Raimondi, and S. Tumminaro, "Nonlinear modeling of DC/DC converters using the Hammerstein's approach," *IEEE transactions on power electronics,* vol. 22, no. 4, pp. 1210-1221, 2007.
[200] V. Valdivia, A. Barrado, A. Laazaro, P. Zumel, C. Raga, and C. Fern·ndez, "Simple Modeling and Identification Procedures for Black-Boò Behavioral Modeling of Power Converters Based on Transient Response Analysis," *IEEE Transactions on Power Electronics,* vol. 24, pp. 2776-2790, 2009.
[201] F. Alonge, F. d Ippolito, F. Raimondi, and S. Tumminaro, "Nonlinear Modeling of DC/DC Converters Using the Hammerstein's Approach," *IEEE Transactions on Power Electronics,* vol. 22, pp. 1210-1221, 2007.

[202] F. Alonge, R. Rabbeni, M. Pucci, and G. Vitale, "Identification and robust control of a quadratic DC/DC boost converter by Hammerstein model," *2014 IEEE Energy Conversion Congress and Exposition (ECCE),* pp. 3355-3362, 2014.
[203] J. A. Oliver, R. Prieto, J. A. Cobos, O. Garcia, and P. Alou, "Hybrid wiener-hammerstein structure for grey-box modeling of dc-dc converters," in *Applied Power Electronics Conference and Exposition, 2009. APEC 2009. Twenty-Fourth Annual IEEE*, 2009: IEEE, pp. 280-285.
[204] M. Al-Greer, M. Armstrong, and J. Xu, "Coordinate descent auto-tuning architecture for multi rail DC-DC switch mode power converters," in *2018 IEEE 19th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2018: IEEE, pp. 1-7.
[205] A. Marchetto, M. M. Islam, W. Asghar, A. Susi, and G. Scanniello, "A multi-objective technique to prioritize test cases," *IEEE Transactions on Software Engineering,* vol. 42, no. 10, pp. 918-940, 2015.
[206] D. T. Ahmed and M. A. Hossain, "Dynamic prioritization of multi-sensor feeds for resource limited surveillance systems," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, 2012: IEEE, pp. 412-416.
[207] T. Pimpalkhute and S. Pasricha, "An application-aware heterogeneous prioritization framework for NoC based chip multiprocessors," in *Fifteenth International Symposium on Quality Electronic Design*, 2014: IEEE, pp. 76-83.
[208] K. J. M. Bautista, M. G. R. Sotelo, and A. I. C. Llanos, "A prioritization system for failures of a vital signs monitor," in *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2018: IEEE, pp. 1-4.
[209] F. L. Ţiplea and C. Bocăneală, "Resource relocation in workflow nets with time, resource, and task priority constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 44, no. 7, pp. 953-965, 2014.
[210] N. Nasser, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE transactions on wireless communications,* vol. 12, no. 4, pp. 1448-1459, 2013.
[211] X. Wang and T. Liu, "Research on Subsystem Hybrid Scheduling and Priority Inversion Based uCOS-II," in *2012 Fifth International Conference on Intelligent Networks and Intelligent Systems*, 2012: IEEE, pp. 117-121.
[212] W. Song, Q. Guo, H. Chen, and W. Hu, "Target Priority Discriminant optimization," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2019: IEEE, pp. 1646-1650.
[213] K. Abd, K. Abhary, and R. Marian, "A fuzzy decision support system for selecting the optimal scheduling rule in robotic flexible assembly cells," *Australian Journal of Multi-Disciplinary Engineering,* vol. 9, no. 2, pp. 125-132, 2013.
[214] A. Maoudj, B. Bouzouia, A. Hentout, and R. Toumi, "Multi-agent approach for task allocation and scheduling in cooperative heterogeneous multi-robot team: Simulation results," in *2015 IEEE 13th international conference on industrial informatics (INDIN)*, 2015: IEEE, pp. 179-184.
[215] V. Jain, S. Agarwal, and K. Goswami, "Dynamic multilevel priority packet scheduling design for WSN," in *2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)*, 2014: IEEE, pp. 86-90.
[216] S. She-wen, "Priority-based soft real-time task scheduling in TinyOS," *Microcomputer Information,* no. 8, p. 7, 2010.
[217] Z. Li, H. Wan, Y. Deng, and M. Gu, "Work-in-Progress: A Flattened Priority Framework for Mixed-Criticality Real-Time Systems," in *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2018: IEEE, pp. 85-88.

[218] H. Sibanda and B. Nleya, "Model analysis of dynamic priority segmented burstification and deflection routing," in *2017 IEEE AFRICON*, 2017: IEEE, pp. 325-331.
[219] N. Bansal and M. Dutta, "Performance evaluation of task scheduling with priority and non-priority in cloud computing," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, 2014: IEEE, pp. 1-4.
[220] T. Cakar, H. R. Yazgan, and R. Koker, "Parallel robot Scheduling with genetic algorithms," *Parallel Manipulators, New Developments,* 2008.
[221] Z. Li, M. Harman, and R. M. Hierons, "Search algorithms for regression test case prioritization," *IEEE Transactions on software engineering,* vol. 33, no. 4, pp. 225-237, 2007.
[222] N. B. Satyendra and N. N. Prasad, "Functional Sorting And Prioritization Of Approval Queues Using Multi Objective Optimizationtechniques," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2019: IEEE, pp. 1-5.
[223] A. S. Cacciapuoti, M. Caleffi, F. Marino, and L. Paura, "On the route priority for cognitive radio networks," *IEEE Transactions on Communications,* vol. 63, no. 9, pp. 3103-3117, 2015.
[224] D. Marijan, "Multi-perspective regression test prioritization for time-constrained environments," in *2015 IEEE International Conference on Software Quality, Reliability and Security*, 2015: IEEE, pp. 157-162.
[225] Y. Zhao, Q. Wang, W. Wang, D. Jiang, and Y. Liu, "Research on the priority-based soft real-time task scheduling in TinyOS," in *2009 International Conference on Information Technology and Computer Science*, 2009, vol. 1: IEEE, pp. 562-565.
[226] K. R. Walcott, M. L. Soffa, G. M. Kapfhammer, and R. S. Roos, "Timeaware test suite prioritization," in *Proceedings of the 2006 international symposium on Software testing and analysis*, 2006, pp. 1-12.
[227] M. Salehie, S. Li, L. Tahvildari, R. Dara, S. Li, and M. Moore, "Prioritizing requirements-based regression test cases: A goal-driven practice," in *2011 15th European Conference on Software Maintenance and Reengineering*, 2011: IEEE, pp. 329-332.
[228] M. J. Arafeen and H. Do, "Test case prioritization using requirements-based clustering," in *2013 IEEE sixth international conference on software testing, verification and validation*, 2013: IEEE, pp. 312-321.
[229] C. D. Nguyen, A. Marchetto, and P. Tonella, "Test case prioritization for audit testing of evolving web services using information retrieval techniques," in *2011 IEEE International Conference on Web Services*, 2011: IEEE, pp. 636-643.
[230] S. Yoo, M. Harman, P. Tonella, and A. Susi, "Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge," in *Proceedings of the eighteenth international symposium on Software testing and analysis*, 2009, pp. 201-212.
[231] X. Wang and H. Zeng, "Dynamic test case prioritization based on multi-objective," in *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2014: IEEE, pp. 1-6.
[232] D. M. Kilgour, Y. Chen, and K. W. Hipel, "Multiple criteria approaches to group decision and negotiation," in *Trends in multiple criteria decision analysis*: Springer, 2010, pp. 317-338.
[233] M. Adriyendi, "Multi-attribute decision making using simple additive weighting and weighted product in food choice," 2015.

[234] M. M. Islam, A. Marchetto, A. Susi, and G. Scanniello, "A multi-objective technique to prioritize test cases based on latent semantic indexing," in *2012 16th European Conference on Software Maintenance and Reengineering*, 2012: IEEE, pp. 21-30.
[235] A. Maoudj, B. Bouzouia, A. Hentout, A. Kouider, and R. Toumi, "Distributed multi-agent approach based on priority rules and genetic algorithm for tasks scheduling in multi-robot cells," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016: IEEE, pp. 692-697.
[236] S. Chakraborty, "Applications of the MOORA method for decision making in manufacturing environment," *The International Journal of Advanced Manufacturing Technology,* vol. 54, no. 9-12, pp. 1155-1166, 2011.
[237] K. Lin, H. Zhao, Z.-y. Yin, and Y.-g. Bi, "An adaptive double ring scheduling strategy based on tinyos," *Journal-Northeastern University Natural Science,* vol. 28, no. 7, p. 985, 2007.
[238] M. Yu, S. Xiahou, and X. Li, "A survey of studying on task scheduling mechanism for TinyOS," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008: IEEE, pp. 1-4.
[239] P. Navrátil and V. Bobál, "Recursive identification algorithms library," in *Proceedings17th International Conference on Process Control*, 2009, pp. 516-523.
[240] J. Morroni, L. Corradini, R. Zane, and D. Maksimovic, "Adaptive tuning of switched-mode power supplies operating in discontinuous and continuous conduction modes," *IEEE Transactions on Power Electronics,* vol. 24, no. 11, pp. 2603-2611, 2009.
[241] M. Ahmeid, "Real-time system identification and self-tuning control of DC-DC power converter using Kalman Filter approach," Thesis (Ph. D.)--Newcastle University, 2017., Newcastle upon Tyne, England, 2017.
[242] O. Trescases, G. Wei, A. Prodic, and W. Tung, "An EMI Reduction Technique for Digitally Controlled SMPS," *IEEE Transactions on Power Electronics,* vol. 22, pp. 1560-1565, 2007.