



Microprocessor Based Signal Processing Techniques for System Identification and Adaptive Control of DC-DC Converters

Maher Mohammed Fawzi Saber Algreer

B.Sc., M.Sc.

A thesis submitted for the degree of

Doctor of Philosophy

May, 2012

School of Electrical, Electronic and

Computer Engineering

Newcastle University

United Kingdom

ABSTRACT

Many industrial and consumer devices rely on switch mode power converters (SMPCs) to provide a reliable, well regulated, DC power supply. A poorly performing power supply can potentially compromise the characteristic behaviour, efficiency, and operating range of the device. To ensure accurate regulation of the SMPC, optimal control of the power converter output is required. However, SMPC uncertainties such as component variations and load changes will affect the performance of the controller. To compensate for these time varying problems, there is increasing interest in employing real-time adaptive control techniques in SMPC applications. It is important to note that many adaptive controllers constantly tune and adjust their parameters based upon on-line system identification. In the area of system identification and adaptive control, Recursive Least Square (RLS) method provide promising results in terms of fast convergence rate, small prediction error, accurate parametric estimation, and simple adaptive structure. Despite being popular, RLS methods often have limited application in low cost systems, such as SMPCs, due to the computationally heavy calculations demanding significant hardware resources which, in turn, may require a high specification microprocessor to successfully implement. For this reason, this thesis presents research into lower complexity adaptive signal processing and filtering techniques for on-line system identification and control of SMPCs systems.

The thesis presents the novel application of a Dichotomous Coordinate Descent (DCD) algorithm for the system identification of a dc-dc buck converter. Two unique applications of the DCD algorithm are proposed; system identification and self-compensation of a dc-dc SMPC. Firstly, specific attention is given to the parameter estimation of dc-dc buck SMPC. It is computationally efficient, and uses an infinite

impulse response (IIR) adaptive filter as a plant model. Importantly, the proposed method is able to identify the parameters quickly and accurately; thus offering an efficient hardware solution which is well suited to real-time applications. Secondly, new alternative adaptive schemes that do not depend entirely on estimating the plant parameters is embedded with DCD algorithm. The proposed technique is based on a simple adaptive filter method and uses a one-tap finite impulse response (FIR) prediction error filter (PEF). Experimental and simulation results clearly show the DCD technique can be optimised to achieve comparable performance to classic RLS algorithms. However, it is computationally superior; thus making it an ideal candidate technique for low cost microprocessor based applications.

DEDICATION

To my loving parents and my wife Israa

ACKNOWLEDGMENTS

I would like to acknowledge everyone for those who made this work possible to complete. First and foremost, I would like to thank God that gives me the patience to complete this work, praise to God.

I would also like to express my deep sincere gratitude towards all my supervisors Dr. Matthew Armstrong, Dr. Damian Giaouris, and Dr. Petros Missailidis for their support, patient guidance and encouragement during my doctoral research. The successful achievement of this work would not be complete without their support. I would like to extend my thanks to Dr. Matthew Armstrong for his amicable nature that he has provided for positive free stress collaboration and for sharing his expertise in practical design. Honestly, he has an exemplary role that always presented kinds words for encouragement.

My acknowledgments also go to my friends and colleagues at PEDM Lab, for their collaboration. My thanks also go to Bassim Jassim for sharing his knowledge on power electronics. In addition, I would like to thank the academic and technicians staff in EECE for their cooperation. My thank towards the head of school Prof. Bayan Sharif for his collaboration during my life in Newcastle city. I thank Mrs Gillian Webber and Deborah Alexander for help in all the administrative work.

I am also grateful indebted to Dr. Yuriy Zakharov of the University of York, for his valuable comments and advice received from him on the DCD algorithm.

I would like to gratefully appreciate the Ministry of Higher Education, from my home country IRAQ, for the financial support during this research, without their sponsorship, I could not complete this work.

Finally, my deepest appreciation to my father and mother, for their love and continues support they provide me through my entire life. I am always imagining my parent happiness when I will be successes in PhD to encourage myself progressing more. I owe all that I have. I would like to warmly thank all my brothers, my lovely sister Moroj and my sons Mohib and Majd, they give me the power to complete this work and give me endless morale support. Last but most important, to say thanks to my wife ISRAA, you always encourage me, given me the strength and enthusiasm to complete this research, she always face the same tension and frustration that I had during my work. This project would not be complete without her understanding and love.

TABLE OF CONTENTS

ABSTRACT	II
DEDICATION.....	IV
ACKNOWLEDGMENTS	V
LIST OF FIGURES	XII
LIST OF TABLES	XIX
LIST OF ACRONYMS	XX
LIST OF SYMBOL.....	XXII
Chapter 1 INTRODUCTION AND SCOPE OF THE THESIS	1
1.1 Introduction	1
1.2 Scope and Contribution of the Thesis	4
1.3 Publications Arising from this Research	6
1.4 Layout of the Thesis	7
1.5 Notations	8
Chapter 2 DC-DC SWITCH MODE POWER CONVERTERS MODELLING AND CONTROL.....	9
2.1 Introduction	9

2.2 DC-DC Circuit Topologies and Operation.....	9
2.2.1 DC-DC Buck Converter Principle of Operation	11
2.3 DC-DC Buck Converter Modelling	11
2.4 Model Simulation	14
2.5 Buck State Space Average Model	15
2.6 Discrete Time Modelling of Buck SMPC	17
2.7 Digital Control Architecture for PWM DC-DC Power Converters	18
2.7.1 Digital Voltage Mode Control	20
2.8 Digital Proportional-Integral-Derivative Control.....	22
2.8.1 Digital Control for Buck SMPC Based on PID Pole-Zero Cancellation ...	25
2.8.2 Pole Placement PID Controller for DC-DC Buck SMPC.....	31
2.9 Chapter Summary	37
Chapter 3 SYSTEM IDENTIFICATION, ADAPTIVE CONTROL AND ADAPTIVE FILTER PRINCIPLES -A LITERATURE REVIEW	38
3.1 Introduction	38
3.2 Introduction to System Identification.....	38
3.3 Parametric and Non-Parametric Identification.....	40
3.4 Model Structures for Parametric Identification	43
3.5 Parametric Identification Process	46
3.6 Adaptive Control and Adaptive Filter Applications.....	47
3.7 Adaptive Control Structures	48
3.8 Adaptive Filter Techniques	49

3.9 Literature Review on System Identification and Adaptive Control for DC-DC Converters	53
3.9.1 Non-Parametric System Identification Techniques and Adaptive Control for SMPC	53
3.9.2 Parametric Estimation Techniques and Adaptive Control for SMPC.....	55
3.9.3 Independent Adaptive Control Technique for SMPC.....	60
3.10 Chapter Summary	62
Chapter 4 SYSTEM IDENTIFICATION OF DC-DC CONVERTER USING A RECURSIVE DCD-IIR ADAPTIVE FILTER	63
4.1 Introduction	63
4.2 System Identification of DC-DC Converter Using Adaptive IIR DCD-RLS Algorithm	65
4.3 Adaptive System Identification	67
4.4 Least Square Parameters Estimation	68
4.5 Conventional RLS Estimation.....	70
4.6 Normal Equations Solution Based On Iterative RLS Approach	72
4.6.1 Exponentially Weighted RLS Algorithm (ERLS)	74
4.7 Coordinate Descent and Dichotomous Coordinate Descent Algorithms	76
4.7.1 Dichotomous Coordinate Descent Algorithm.....	80
4.8 Pseudo-Random Binary Sequence and Persistence Excitation	82
4.9 Discrete Time Modelling of DC-DC Converter and Adaptive IIR Filter	86
4.9.1 Equation Error IIR Adaptive Filter	88
4.10 Parameter Estimation Metrics and Validation.....	91

4.11 Model Example and Simulation Results	92
4.12 Adaptive Forgetting Strategy	101
4.12.1 Fuzzy RLS Adaptive Method for Variable Forgetting Factor	101
4.13 Simulation Test.....	106
4.14 Chapter Summary	110
Chapter 5 ADAPTIVE CONTROL OF A DC-DC SWITCH MODE POWER CONVERTER USING A RECURSIVE FIR PREDICTOR	112
5.1 Introduction	112
5.2 Self-Compensation of a DC-DC Converter Based on Predictive FIR	113
5.3 Auto-Regressive / Process Generation, Identification.....	114
5.3.1 Relationship between Forward Prediction Error Filter and AR Identifier	116
5.3.2 One-Tap Linear FIR Predictor for PD Compensation	120
5.4 Least Mean Square Algorithm.....	121
5.5 Simulation Results.....	124
5.5.1 Reference Voltage Feed-Forward Adaptive Controller	125
5.5.2 Voltage Control Using Adaptive PD+I Controller	128
5.6 Robustness and Stability Analysis for the Proposed Adaptive PD+I Controller	134
5.7 Chapter Summary	138
Chapter 6 MICROPROCESSOR APPLICATION BASED SYNCHRONOUS DC-DC SWITCH MODE POWER CONVERTER-EXPERIMENTAL RESULTS	139
6.1 Introduction	139

6.2 Microprocessor Control Platform.....	139
6.2.1 Microprocessor Code Development.....	141
6.3 System Hardware Description and Microprocessor Setup.....	142
6.4 System Identification Using DCD-RLS / Experimental Validation.....	148
6.5 Realisation of the Converter Model	155
6.6 Adaptive Controller / Experimental Validation	157
6.7 Complexity Reduction.....	162
6.8 Chapter Summary	165
Chapter 7 CONCLUSION AND FUTURE WORK	166
7.1 Conclusion.....	166
7.2 Future Work	169
APPENDIX A	172
DERIVATION OF RLS ALGORITHM BASED ON MATRIX INVERSION LEMMA	172
APPENDIX B	175
SCHEMATIC CIRCUIT OF THE SYNCHRONOUS BUCK CONVERTER	175
APPENDIX C	178
SIMULINK MODEL OF THE PROPOSED STRUCTURES	178
REFERENCES	180

LIST OF FIGURES

Fig. 1.1 Dual core microprocessor and digital control architecture for SMPCs	3
Fig. 2.1 Most common dc-dc converter topologies, a: buck converter, b: boost converter, c: buck-boost converter	10
Fig. 2.2 Buck converter circuit configuration, a: <i>On</i> state interval, b: <i>Off</i> state interval.....	13
Fig. 2.3 Open loop steady state output voltage	14
Fig. 2.4 Open loop steady state inductor current.....	15
Fig. 2.5 Digital voltage mode control architecture of DC-DC SMPC.....	20
Fig. 2.6 Two-poles / Two-zeros IIR digital controller	22
Fig. 2.7 Digital PID compensator.....	23
Fig. 2.8 Frequency response of the compensated and uncompensated dc-dc buck SMPC	28
Fig. 2.9 Power stage root locus	28
Fig. 2.10 PID compensator root locus	29
Fig. 2.11 Total loop gains root locus.....	29
Fig. 2.12 Transient response of the PID controller, a: output voltage, b: inductor current, c: load current. Load current change between 0.66 A and 1.32 A every 5 ms	30
Fig. 2.13 Closed loop control of the buck SMPC	31

Fig. 2.14 Frequency response of the compensated and uncompensated dc-dc buck SMPC	34
Fig. 2.15 Transient response of the pole-placement PID controller, a: output voltage, b: inductor current, c: load current. Load current change between 0.66 A and 1.32 A every 5 ms	35
Fig. 2.16 Loop-gain comparison between pole-placement and pole-zero PID controllers.....	36
Fig. 2.17 Comparison of transient response results between pole-placement and pole-zero PID controllers. Repetitive load current change between 0.66 A and 1.32 A every 5 ms	36
Fig. 3.1 General block diagram of parametric identification	39
Fig. 3.2 General linear model transfer function	41
Fig. 3.3 Parametric identification model structures	44
Fig. 3.4 Parametric identification flowchart.....	46
Fig. 3.5 Adaptive model reference structure	48
Fig. 3.6 Self-tuning controller block-diagram.....	49
Fig. 3.7 An adaptive filter structure	50
Fig. 3.8 Adaptive Filter structures, a: system identification, b: signal prediction, c: inverse modelling, d: noise cancellation	52
Fig. 4.1 The proposed closed loop adaptive IIR identification method using DCD-RLS algorithm.....	65
Fig. 4.2 Adaptive system identification block diagram.....	67
Fig. 4.3 Closed loop operation of conventional RLS algorithm based matrix inversion lemma	71
Fig. 4.4 Nine-bits single period PRBS	84

Fig. 4.5 Nine-bits shift register with XOR feedback for 511 maximum length PRBS generation	84
Fig. 4.6 Ideal auto-correlation of an infinite period of PRBS	85
Fig. 4.7 Single period 9-bit auto-correlation of PRBS	86
Fig. 4.8 System identification based on adaptive IIR filter using output error block diagram.....	88
Fig. 4.9 System identification based on adaptive IIR filter using equation error block diagram.....	90
Fig. 4.10 The procedure of system identification.....	93
Fig. 4.11 Identification sequence, a: output voltage during ID, b: voltage model parameters ID, c: voltage error prediction, d. ID enable signal	95
Fig. 4.12 Tap-weights estimation for IIR filter using DCD-RLS and classical RLS methods; compared with calculated model	97
Fig. 4.13 Prediction error signals, a: classical RLS, b: DCD-RLS	97
Fig. 4.14 Parameters estimation error, a: classical RLS, b: DCD-RLS	98
Fig. 4.15 Tap-weights estimation DCD-RLS at $Nu = 4$ and classical RLS	99
Fig. 4.16 Tap-weights estimation DCD-RLS and CD algorithms	99
Fig. 4.17 Frequency responses for control-to-output transfer of function; estimated and calculated model.....	100
Fig. 4.18 The proposed system identification structure for a dc-dc converter based on RLS fuzzy AFF	102
Fig. 4.19 General block diagram of the fuzzy logic system.....	103
Fig. 4.20 Fuzzy logic input and output membership functions, a: e_p^2 , b: Δe_p^2 , c: λ	105

Fig. 4.21 Parameters estimation of control-to-output voltage transfer of a dc-dc converter at load changes from 5-to-1 Ω using DCD-RLS algorithm at a: $\lambda = 0.7$, b: $\lambda = 0.99$, c: fuzzy AFF	108
Fig. 4.22 Prediction error signal during initial start-up and at load change	109
Fig. 4.23 Forgetting factor at initial start-up and at load change	109
Fig. 5.1 Adaptive PD+I controller using one tap DCD-RLS PEF	113
Fig. 5.2 Reconstruction of white noise.....	114
Fig. 5.3 AR process generator	115
Fig. 5.4 AR process identifier	116
Fig. 5.5 One step ahead forward predictor	117
Fig. 5.6 Forward prediction error filter	118
Fig. 5.7 Prediction error filter.....	118
Fig. 5.8 AR analyser, a: matched Inverse MA filter, b: one tap adaptive PEF, c: two tap adaptive PEF filter. The dotted line is the estimated output and the solid line is the actual input	119
Fig. 5.9 Closed loop LMS system block diagram	124
Fig. 5.10 Reference voltage feed-forward: Comparison of transient response between LMS and DCD-RLS. Repetitive load change between 0.66 A and 1.32 A every 5 ms	125
Fig. 5.11 Zoomed adaptation of gain (K_d) and tap-weight (w_1) in the two stage adaptive linear predictor for different step-size values	127
Fig. 5.12 Transient response of the proposed adaptive controller, a: output voltage, b: inductor current, c: load current change between 0.66A and 1.32 A every 5 ms .	129
Fig. 5.13 Error signal behaviour during adaptation process, a: loop error (e_L), b: prediction error (e_{p1}). Load current change between 0.66 A and 1.32 A every 5 ms	130

Fig. 5.14 Transient response of the proposed adaptive PD+I controller using DCD-RLS or LMS. Load current change between 0.66 A and 1.32 A every 5 ms.....	131
Fig. 5.15 Transient response of the proposed adaptive controller during load current change between 0.66 A and 1.32 A every 5 ms, a: output capacitance $C = 150 \mu\text{F}$ and $L = 220 \mu\text{H}$, b: $C = 660 \mu\text{F}$ and $L = 220 \mu\text{H}$, c: output inductor $L = 100 \mu\text{H}$ and $C = 330 \mu\text{F}$	133
Fig. 5.16 Comparison of transient response results between the proposed adaptive PD+I using DCD-RLS and pole-zero PID control. Repetitive load current change between 0.66 A and 1.32 A every 5 ms	134
Fig. 5.17 Frequency response of the PD + I compensator and the compensated / uncompensated open loop gains.....	135
Fig. 5.18 Closed loop scheme of voltage mode control for SMPC.....	136
Fig. 5.19 Sensitivity functions of the PD+I controller	137
Fig. 5.20 Margins on Nyquist plot	137
Fig. 6.1 TMS320F28335 <i>eZdsp</i> Architecture [129].....	140
Fig. 6.2 Hardware platform setup.....	142
Fig. 6.3 Block diagram of the synchronous dc-dc buck converter based on microprocessor	143
Fig. 6.4 a: TMS320F28335™ DSP platform, b: the synchronous dc-dc buck converter circuit	145
Fig. 6.5 PWM waveforms in open loop circuit test, a: duty ratio 50 % , b: duty ratio 33 %	147
Fig. 6.6 Leading DCD-RLS algorithm flowchart.....	148
Fig. 6.7 Experimental output voltage waveform when identification enabled. (ac coupled).....	149

Fig. 6.8 Experimental output voltage and persistence excitation signal (duty signal + Δ_{PRBS}) results during ID, based on sampled data collected from DSP.....	151
Fig. 6.9 Experimental tap-weights estimation for IIR filter with DCD-RLS and classical RLS methods; compared with the calculated model	152
Fig. 6.10 Experimental prediction error results, a: conventional RLS, b: DCD-RLS	152
Fig. 6.11 Experimental parameters estimation error, a: classical RLS, b: DCD-RLS	153
Fig. 6.12 Experimental learning curves comparison results of conventional RLS against DCD-RLS at different iteration values	154
Fig. 6.13 Experimental sampled data collected from DSP, a: output voltage, b: control signal (duty signal + Δ_{PRBS}).....	155
Fig. 6.14 Model errors comparison between third/second order output error and equation error model	156
Fig. 6.15 Transient response of PID controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient.....	159
Fig. 6.16 Transient response of adaptive PD+I DCD-RLS controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient	160
Fig. 6.17 Transient response of adaptive PD+I LMS controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient.....	161
Fig. 6.18 Load transient response at significant change in load current, with two stage DCD-DCD adaptive controller and hybrid DCD-LMS adaptive controller....	163

Fig. 6.19 Transient response of hybrid DCD-RLS:LMS ($\mu = 1$) adaptive controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient 164

LIST OF TABLES

Table 4.1 Conventional RLS algorithm based matrix inversion lemma	71
Table 4.2 Iteratively solving for auxiliary equations	74
Table 4.3 ERLS algorithm using auxiliary equations	76
Table 4.4 Exact line search algorithm description	77
Table 4.5 Cyclic CD algorithm description.....	79
Table 4.6 Leading CD algorithm description	79
Table 4.7 Cyclic DCD algorithm description.....	80
Table 4.8 Leading DCD algorithm description	82
Table 4.9 Bit cell setup for different MLBS generation.....	84
Table 4.10 Discrete time control-to-output transfer function identification	100
Table 4.11 The rule base for the forgetting factor (λ)	106
Table 5.1 LMS algorithm operation	124
Table 6.1 Prototyped synchronous buck converter parameters.....	144

LIST OF ACRONYMS

AC	Alternating Current
ADC	Analogue -to-Digital Converter
AR	Auto-Regressive
ARMA	Auto Regressive Moving Average Model
CCM	Continuous Conduction Mode
CCS	Code Composer Studio
CD	Coordinate Descent
CPU	Central Processing Unit
DAC	Digital-to-Analogue Converter
DC	Direct Current
DCD	Dichotomous Coordinate Descent
DCM	Discontinuous Conduction Mode
DPWM	Digital Pulse Width Modulation
DSP	Digital Signal Processor
ERLS	Exponentially Weighted Recursive Least Square
FFT	Fast Fourier Transform
FIR	Finite Impulse Response

FL	Fuzzy logic
FPGA	Field Programmable Gate Array
IC	Integrated Circuit
IDE	Integrated Development Environment
IIR	Infinite Impulse Response
LCO	Limit Cycle Oscillation
LMS	Least Mean Square
LS	Least Square
LTI	Linear Time Invariant
MA	Moving Average
MLBS	Maximum Length Pseudo Binary Sequence
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
MSE	Mean Square Error
PD	Proportional-Derivative
PEF	Prediction Error Filter
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PRBS	Pseudo Random Binary Sequence
RLS	Recursive Least Square
SI	System Identification
SMPC	Switch Mode Power Converter
ZOH	Zero-Order-Hold

LIST OF SYMBOL

μ	Step size
C	Capacitor
$d(n)$	Control signal
e_p	Prediction error
f_o	Corner frequency
f_s	Sampling frequency
i_L	Inductor current
i_o	Load current
K_D	Derivative gain
K_I	Integral gain
K_P	Proportional gain
L	Inductor
M_p	Maximum overshoot
Q	Quality Factor
t_r	Time rise
T_{sw}	Switching time

v_C	Capacitor voltage
V_{in}	Input voltage
v_L	Inductor voltage
v_o	Output voltage
V_{ref}	Reference voltage
\hat{w}	Estimated filter weight
\hat{y}	Estimated output
Δ_{PRBS}	PRBS amplitude
θ	Parameters vector
λ	Forgetting factor
φ	Regression vector

Chapter 1

INTRODUCTION AND SCOPE OF THE THESIS

1.1 Introduction

Many classical control schemes for switch mode power converters (SMPCs) suffer from inaccuracies in the design of the controller. This may be due to poor knowledge of the load characteristics, or unexpected external disturbances in the system. In addition, SMPC uncertainties such as component tolerances, unpredictable load changes, changes in ambient conditions, and ageing effects, all affect the performance of the controller over time [1, 2]. Consequently, greater consideration should be given to the design of the controller to accommodate these uncertainties in the system. Therefore, an intermediate process is required to explicitly determine the parameters of the power converter and to estimate the dynamic characteristics of the SMPC. This process can be achieved by system identification algorithms. Also, in SMPC applications, often a classical Proportional-Integral-Derivative (PID) controller is employed using fixed controller gains. In such systems, the fixed control loop is unable to consider parameter changes that may occur during the normal operation of the plant. Ultimately, this limits the stability margins, robustness, and dynamic performance of the control system [3].

For this reason, more advanced auto-tuning and adaptive digital controllers are now playing an increasingly important role in SMPC systems. With the advent of developments in digital control techniques, intelligent and advanced control algorithms can now readily be incorporated into the digital based systems to significantly improve the overall dynamic performance of the process. On-line

identification, system monitoring, adaptive and self-tuning controllers are some of the most attractive features of digital control systems. These intelligent algorithms, which are well suited to SMPC applications, allow more optimised control designs to be realised [2, 4] and can rapidly adjust controller settings in response to system parameter variation. Clearly, an accurate model is required (transfer function, state space), and therefore excellent estimation of plant parameters is essential. Here, the controller tuning is based upon on-line system identification techniques, and therefore a discrete time transfer function of the SMPCs is necessary for control design [5, 6]. This is particularly true in most adaptive and self-tuning controllers which require system identification to update the control parameters. The fundamental principle of system identification and parameter estimation is to evaluate the parameters within a transfer function which has an analogous arrangement to the actual plant to be controlled. However, system identification and adaptive controllers are not fully exploited in low cost, low power SMPCs due to the heavy computational burden they place upon the microprocessor platform. Complex algorithms often require higher performance hardware to implement and this is usually cost prohibitive in applications such as SMPCs [7]. Therefore, there is a requirement to further research and develop cost effective, computationally light identification and adaptation methods which offer accurate estimation performance.

Recent developments in digital hardware; including microprocessors, microcontrollers, digital signal processors (DSP) and field programmable gate arrays (FPGA), provide the ability to design and implement a complex system at higher sampling rate, such as adaptive and self-tuning controllers. However, the execution time of adaptive algorithms is dependent upon several factors: processor architectures, memory, data/address bus widths, clock rate, *etc.*

Fortunately, the industrial electronics companies have been attempting to release adaptive and self-tuning controllers in SMPC applications. The scheme of these controllers is based upon real time identification and system monitoring of SMPCs, using new microprocessor architecture; including multiprocessor cores (Fig. 1.1). As shown in Fig. 1.1, the digitally controlled block-diagram of SMPCs is classified as a mixed signal system. In this structure two kinds of signals are used: analogue/digital

or discrete signal. The analogue system consists of dc-dc power stage circuit, sensing/signal conditioning circuit, and gate drive circuit. The digital system consists of digital compensator, a digital-pulse-width-modulation (DPWM) circuit and an analogue-to-digital converter (ADC) that provides an interface between the digital and analogue domains.

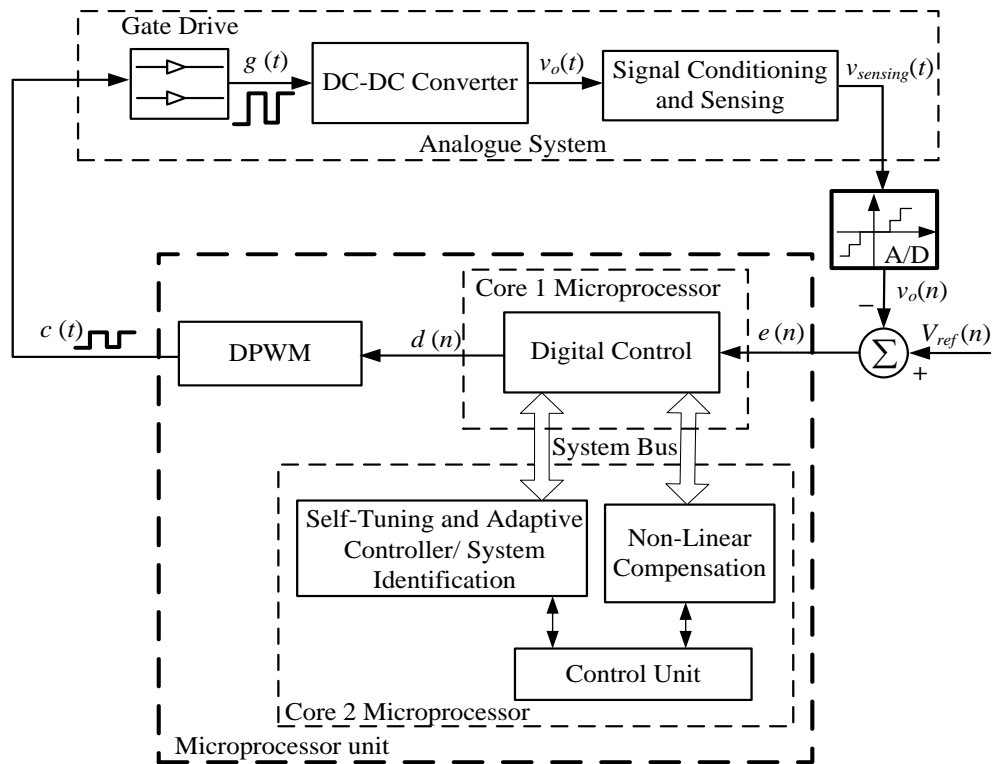


Fig. 1.1 Dual core microprocessor and digital control architecture for SMPCs

The new configuration of Power Electronics Management (PEM) will increase the performance of the microprocessor without increasing the power consumption. Here, the tasks are divided between the two processor cores. The first microprocessor core is designed for simple control regulation such as a conventional digital PID control. The second microprocessor core provides advanced control implementation, for instance adaptive and system identification algorithm. In some PEM units, non-linear control techniques have also been introduced in the second microprocessor core to further improve the transient characteristics of the system. As illustrative examples, “*POWERVATION*[®]” creates a dual core PEM-IC (PV3002). This IC is capable of tuning the controller gains at load current variations, and at circuit parameters change

(output capacitor/inductor) based on cycle-by-cycle output voltage monitoring. The PV3002 includes several analogue circuits, DSPs, and a reduced instruction set computing (RISC) microprocessor [8, 9]. “*INTERSIL Zilker Labs*” designed a digital adaptive controller IC, namely the ZL6100. This processor can compensate the feedback loop automatically to produce optimal controller performance during output load changes. A non-linear controller utilises this architecture to further improve the dynamic response in the event of abrupt load change [10]. In another example from *TEXAS INSTRUMENTS (TI)*, the attractive features of system identification have been used for the purpose of monitoring the performance of SMPCs, and to update the feedback control loop. In this device (UCD9240) non-linear gains have been augmented to further improve the dynamic behaviour of the system [11, 12].

1.2 Scope and Contribution of the Thesis

Recent advances in microprocessor technology and continual price improvement now allows for more advanced signal processing algorithms to be implemented in many industrial and commercial products, cost and complexity are clearly a major concern. For this reason, the aim of this thesis is to research new practical solutions for system identification and adaptive control that can easily be developed in low complexity systems, whilst maintaining the performance of conventional algorithms. Particular attention is given to parametric estimation and self-compensator design of switch mode dc-dc power converters. In this thesis, the work is applied to a small synchronous dc-dc buck converter. However, the proposed techniques are transferable to other applications.

In order to quickly and accurately identify the system dynamics of a SMPC, a new adaptive method known as Dichotomous Coordinate Descent-Recursive-Least-Square (DCD-RLS) algorithm is proposed. An equation error IIR adaptive filter scheme is developed along with the DCD-RLS algorithm for system modelling of dc-dc SMPC. The design and implementation of the proposed DCD-RLS technique is presented in detail, and results are compared and verified against classical techniques (RLS). A major conclusion from the work is that the DCD-RLS can achieve similar estimation performance to the classic RLS technique, but with a lighter computational burden on the microprocessor platform. The proposed scheme has successfully been presented

by the author in [13]. In addition, an enhancement on the scheme is suggested by employing a variable forgetting factor based fuzzy logic algorithm for the identification of the SMPC parameters. The concept of this scheme is presented in the thesis and the advantages it delivers are discussed. The simulation results for the proposed adaptive forgetting factor with fuzzy logic scheme has been published by the author in [14].

System identification is a first step to developing adaptive and self-tuning controllers. Therefore, the computation complexity of these structures is typically very high. Furthermore, in order to achieve a good quality, dynamic closed loop control system, the unknown parameters of the plant should be estimated quickly and accurately. With these issues in mind; this thesis presents a new alternative adaptive scheme that does not depend entirely on estimating the plant parameters. This scheme is based on adaptive signal processing techniques which are suitable for both prediction/identification and controller adaptation. Importantly, and explained in detail in this thesis, the method the use of an adaptive prediction error filter (PEF) as a main control in the feedback loop. A two stage/one-tap FIR adaptive PEF is placed in parallel with a conventional integral controller to produce an adaptive Proportional-Derivative + Integral (PD+I) controller. The DCD-RLS algorithm is incorporated into the PD+I controller for real time estimation of the PEF tap-weights and for reducing the computational complexity of the classical RLS algorithms for efficient hardware implementation. Simulation and experiments results of the proposed scheme have been published by the author in [15, 16]. The mathematical analysis and concept of using an adaptive PEF for adaptive control, and the relationship between an adaptive PEF and a Proportional-Derivative (PD) controller, are clearly described by the author in the thesis and have been published in [17].

In summary, the main objectives and contributions of this research are:

- To propose a novel method, based on the DCD algorithm, for on-line system identification of dc-dc converters.
- Application of the DCD-RLS algorithm to reduce computation complexity compared to classical methods (RLS).

- To develop an equation error IIR adaptive filter for system modelling of dc-dc converters based upon the DCD-RLS algorithm.
- To apply an adaptive forgetting factor strategy to track the time varying parameters of SMPCs using a fuzzy logic approach.
- To develop a new adaptive controller for SMPCs based upon an FIR prediction error filter using DCD-RLS and LMS adaptive algorithms.
- To experimentally assess the performance of the proposed adaptive DCD-RLS algorithm using a Texas Instruments TMS320F28335 DSP platform and synchronous dc-dc buck converter.

1.3 Publications Arising from this Research

The research in this thesis has resulted in number of journals and international conference publications. These articles are listed below:

- 1- **M. Algreer**, M. Armstrong, and D. Giaouris, “Active On-Line System Identification of Switch Mode DC-DC Power Converter Based on Efficient Recursive DCD-IIR Adaptive Filter”, *IEEE Transactions on Power Electronics*, vol.27, pp.4425-4435, Nov. 2012.
- 2- **M. Algreer**, M. Armstrong, and D. Giaouris, “Adaptive PD+I Control of a Switch Mode DC-DC Power Converter Using a Recursive FIR Predictor”, *IEEE Transactions on Industry Applications*, vol.47, pp.2135-2144, Oct. 2011.
- 3- **M. Algreer**, M. Armstrong, and D. Giaouris, “Predictive PID Controller for DC-DC Converters Using an Adaptive Prediction Error Filter,” in *Proc. IET International Conf. on Power Electron., Machines and Drives, PEMD 2012*, vol. 2012, Bristol, United Kingdom.
- 4- **M. Algreer**, M. Armstrong, and D. Giaouris, “Adaptive Control of a Switch Mode DC-DC Power Converter Using a Recursive FIR Predictor,” in *Proc. IET International Conf. on Power Electron., Machines and Drives, PEMD 2010*, vol. 2010, Brighton, United Kingdom.

- 5- **M. Algreer**, M. Armstrong, and D. Giaouris, "System Identification of PWM DC-DC Converters during Abrupt Load Changes," in *Proc. IEEE Industrial Electron. Conf., IECON'09*, 2009, pp. 1788 – 1793, Porto, Portugal.

1.4 Layout of the Thesis

The thesis is organised into 7 chapters as follow:

Chapter 2 presents the modelling and control of dc-dc power converters. This includes the common circuit topologies of dc-dc converters with more emphasis on operation and circuit configuration of buck dc-dc switch mode power converters. It also provides details on derivation of the continuous state space model, followed by details on average and discrete models of buck dc-dc converter. A digital voltage mode control structure is introduced in this chapter; sub-circuit blocks are also explained. In the digital control section, two techniques of digital compensator are discussed including the pole-zero cancellation method and pole-placement approach. The modelling and control in this chapter will be used to evaluate the proposed algorithms.

Chapter 3 provides details on the principles and techniques used in system identification. Different common models of parametric estimation techniques are also demonstrated. In addition, it outlines basic information on adaptive control and adaptive filter techniques. Recent publications on system identification/adaptive control techniques for dc-dc SMPCs are also reviewed in this chapter.

Chapter 4 presents details on the derivation of the classical LS and RLS algorithms. In addition, it briefly explains the system identification paradigm based adaptive filter technique. The proposed on-line system identification scheme for SMPC is also described in this chapter. This is followed by in-depth analyses and derivation of the new DCD-RLS adaptive algorithm along with equation error IIR adaptive filter structure. Each sub block in the on-line system identification structure is explained. Furthermore, Chapter 4 explores a new adaptive forgetting factor based fuzzy logic system to detect and estimate the fast change in the system via sudden change in prediction error signal. The new identification schemes in this chapter are comprehensively tested and validate through simulations.

Chapter 5 presents the proposed adaptive controller. The first part of this chapter provides details on the principle of how an adaptive PEF filter can be employed as a central controller in the feedback loop of a closed loop system. Following this, an overview of auto-regressive and moving average filters is presented along with the derivation of the Least-Mean-Square (LMS) adaptive algorithm. In addition, Chapter 5 demonstrates the effectiveness of the DCD-RLS adaptive algorithm to improve the dynamics performance of the proposed adaptive scheme. Robustness and stability analysis of the proposed controller is discussed. Extensive simulation results that compare the proposed adaptive control based upon DCD-RLS with classical LMS are provided in this chapter.

Chapter 6 focuses on the experimental validation of the developed adaptive algorithms using a high speed microprocessor board. It provides an overview on the architecture of the selected digital signal processor platform. In addition, this chapter explains the practical circuit diagram of the constructed dc-dc buck converter and the experimental setup. Importantly, Chapter 6 concentrates on practical evaluation of the proposed system identification algorithm and adaptive controller structure. It also provides a comparison between the obtained experimental results of the proposed scheme using the DCD-RLS algorithm and the classical RLS/LMS algorithms, as well as with the conventional digital PID controller.

Finally, Chapter 7 presents the conclusion drawn for this thesis and it summarises possible suggestions for future work.

1.5 Notations

In this thesis the matrices and vectors are represented by bold upper case and bold lower case characters respectively. As an illustrative example, \mathbf{R} and \mathbf{r} . The elements of the matrix and vector are denoted as $R_{i,i}$ and r_i . The i -th column of \mathbf{R} is denoted as $\mathbf{R}^{(i)}$. Finally, variable n is used as a time index, for instance $\mathbf{\beta}(n)$ is the vector $\mathbf{\beta}$ at time instant n .

Chapter 2

DC-DC SWITCH MODE POWER CONVERTERS

MODELLING AND CONTROL

2.1 Introduction

DC-DC SMPCs are extensively used in a wide range of electrical and electronic systems, with varying power levels (typically mW-MW applications). Some illustrative examples are power supplies in personal/laptop computers, telecommunications devices, motor drives, and aerospace systems. These applications require SMPCs with a high performance voltage regulation during static and dynamic operations, high efficiency, low cost, small size/lightweight, and reliability [18-20]. The main role of dc-dc converters is to convert the unregulated DC input voltage into a different regulated level of DC output voltage. In general, a dc-dc converter can be described as an analogue power processing device that contains a number of passive components combined with semiconductor devices (diodes and electronics switches) to produce a regulated DC output voltage that has a different magnitude from the DC input voltage. Some examples refer to the power supply of the microprocessor and other integrated circuits that require a low regulated DC voltage between 3.3 V and 5 V. This voltage is resultant from the reduction of the high DC voltage generated from an AC-to-DC power rectifier [18].

2.2 DC-DC Circuit Topologies and Operation

Configuring the components of dc-dc converters in different ways will lead to the forming of various power circuit topologies (Fig. 2.1). All of the circuit topologies have the same types of components including capacitor (C), inductor (L), load resistor

(R_o), and the lossless semiconductor components. The selection of the topology is mainly dependent on the desired level of regulated voltage, since the dc-dc converters are applied to produce a regulated DC voltage with a DC level different from the input DC voltage. This level can be higher or lower than the DC input voltage. However, the most widely used SMPCs are known as: buck converter, boost converter and buck-boost converter. A dc-dc buck converter is configured to generate a DC output voltage lower than the input voltage, Fig. 2.1(a). Conversely, a dc-dc boost converter is utilised to provide a DC output higher than the applied input voltage, as shown in Fig. 2.1(b). Finally, a dc-dc buck-boost converter is able to produce two levels of DC output voltage; these levels can either be lower or higher than the DC input voltage [19]. See Fig. 2.1(c).

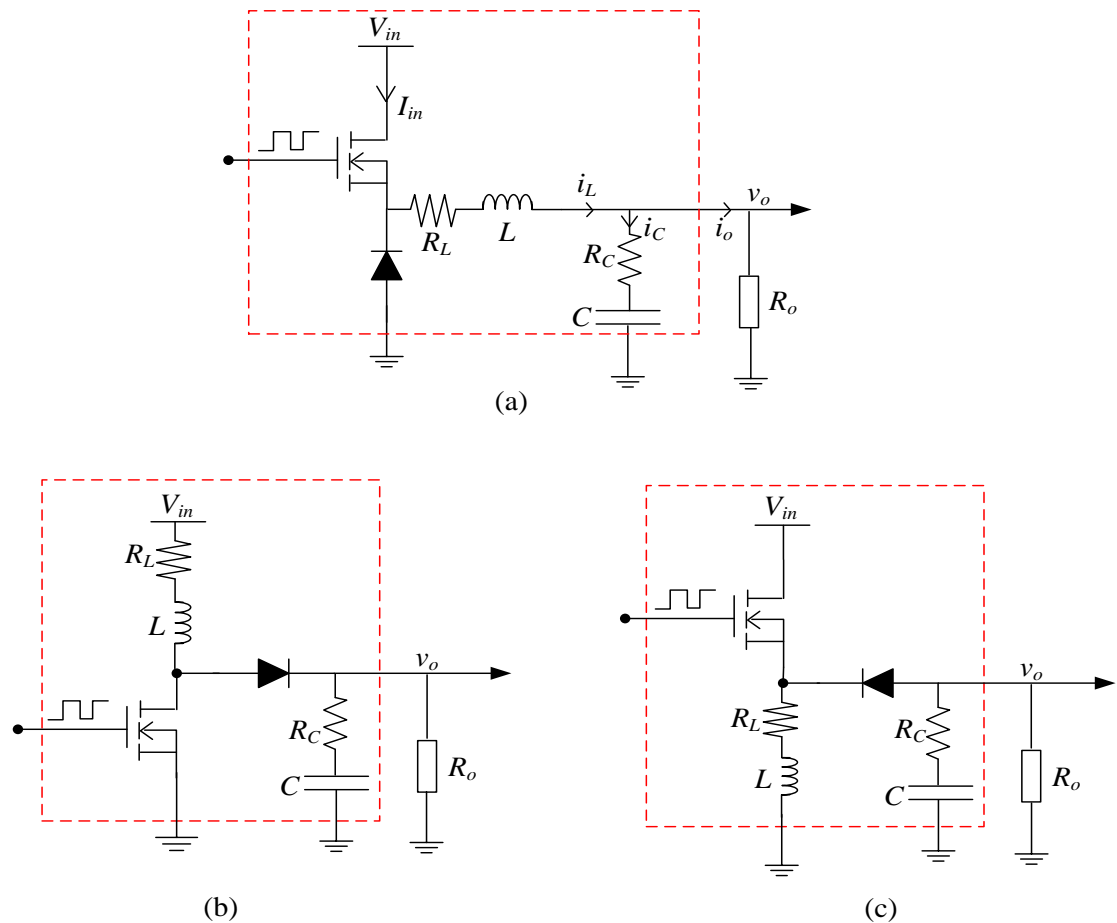


Fig. 2.1 Most common dc-dc converter topologies, a: buck converter, b: boost converter, c: buck-boost converter

2.2.1 DC-DC Buck Converter Principle of Operation

The buck converter is employed to step down the input voltage (V_{in}) into a lower output voltage (V_o). This can be achieved by controlling the operation of the power switches (*e.g.* MOSFET), usually by using a PWM signal. Accordingly, the states of the switch (*On/Off*) are changed periodically with a period equal to T_{sw} (switching period) and conversion ratio (duty-cycle) equal to D . The level of the converted DC voltage is based on the magnitude of the applied input voltage and the duty ratio. During the steady-state, the duty cycle is calculated by $D = V_o / V_{in}$ [19]. Then, the L - C low pass filter removes the switching harmonics from the applied input signal. In practice, to deliver a smooth DC voltage to the connected load, the selected corner frequency of this filter should be much lower than the switching frequency (f_{sw}) of the buck converter [18]. This corner frequency is defined as:

$$f_o = \frac{1}{2\pi\sqrt{LC}} \quad (2.1)$$

Two switching states are apparent during each switch period. The first state is when the switch is *On* and the diode is *Off*. At this state, the input voltage will pass energy to the load through the inductor and the storage elements start to charge. The second state is when the switch is *Off* and the diode is *On*; then the stored energy will discharge through the diode. This operation is known as a Continuous Conduction Mode (CCM). In CCM the inductor current will not drop to zero during switching states, whilst in second operation mode which is Discontinuous Conduction Mode (DCM), the inductor current drops to zero before the end of the switching interval. As a result, a third switching state is introduced during the switching period. In this state, the inductor current drops and remains at zeros while both the diode and switch are *Off* during the operation interval [19].

2.3 DC-DC Buck Converter Modelling

In order to design an appropriate feedback controller, it is essential to define the model of the system. Accordingly, this section presents the details of analysis and modelling of the dc-dc converter. This research focuses on modelling and control of the synchronous dc-dc buck converter, as this topology is widely used in industrial

and commercial products [10, 12]. In synchronous dc-dc buck converter the free-wheel diode is replaced by another MOSFET device. A point of load (POL) converter is one of the applications that utilises this kind of topology. As previously mentioned, there are two intervals per switching cycle. The switching period is defined as the sum of the *On* and *Off* intervals ($T_{sw} = T_{on} + T_{off}$). The ratio of the T_{on} interval to the switch period is known as the duty ratio or duty cycle ($D = T_{on} / T_{sw}$). In the steady-state operation, the output voltage can be computed in terms of duty cycle. The buck dc-dc converter produces a lower output voltage compared with the input voltage (2.2). As expressed in (2.2), the variation of the output voltage magnitude is controlled by the T_{on} duration or duty cycle value. The PWM signal is used to control the output voltage level [19].

$$V_o = \frac{T_{on}}{T_{sw}} V_{in} = DV_{in} \quad (2.2)$$

During the T_{on} duration, the circuit diagram of the buck converter can simply be depicted as in Fig. 2.2(a). A set of differential equations are derived to describe this period of operation:

$$L \frac{di_L}{dt} = V_{in} - (R_L + R_C)i_L - v_C + R_C i_o \quad (2.3)$$

$$C \frac{dv_C}{dt} = i_L - i_o = i_L - \frac{v_o}{R_o} \quad (2.4)$$

$$v_o = R_C(i_L - i_o) + v_C = R_C C \frac{dv_C}{dt} + v_C \quad (2.5)$$

Generally, the dc-dc converter model is defined by state-space matrices [21]:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}_1 \mathbf{x}(t) + \mathbf{B}_1 V_{in} \\ y &= \mathbf{C}_1 \mathbf{x}(t) + \mathbf{E}_1 V_{in} \end{aligned} \quad (2.6)$$

Here, \mathbf{A}_1 , \mathbf{B}_1 , \mathbf{C}_1 , and \mathbf{E}_1 are the system matrices/vectors during the *On* interval, y is the output, and $\mathbf{x}(t)$ is the capacitor voltage and inductor current state vector: $\mathbf{x} = [v_C \quad i_L]^T$.

By substituting equation (2.4) into (2.5) and solving with respect to the output voltage (v_o), the output vector can be written in state space matrix form as:

$$\mathbf{y} = \mathbf{C}_1 \mathbf{x} = \begin{bmatrix} \frac{R_o}{R_o + R_C} & \frac{R_o R_C}{R_o + R_C} \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} \quad (2.7)$$

Now, inserting equations (2.3)-(2.5) into (2.6), the *On* state space matrix \mathbf{A}_1 and vector \mathbf{B}_1 can be expressed as [21]:

$$\mathbf{A}_1 = \begin{bmatrix} \frac{-1}{C(R_o + R_C)} & \frac{1}{C} \frac{R_o}{(R_o + R_C)} \\ \frac{-R_o}{(R_o + R_C)L} & \frac{-1}{L} \left(R_L + \frac{R_o R_C}{R_o + R_C} \right) \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \quad (2.8)$$

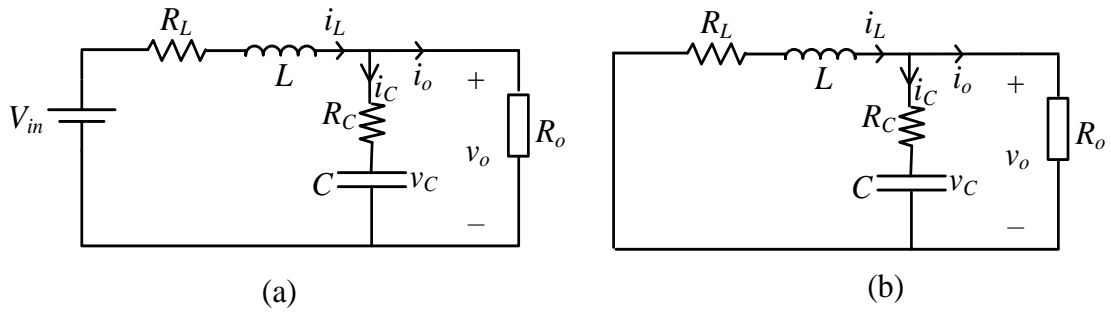


Fig. 2.2 Buck converter circuit configuration, a: *On* state interval, b: *Off* state interval

In the second interval, during the T_{off} duration, the system equations of the buck converter have the same form with T_{on} interval. The only difference between the *On* and *Off* duration is the \mathbf{B} vector (2.9). Fig. 2.2(b) presents the circuit diagram of the buck converter during the *Off* interval.

$$\mathbf{A}_1 = \mathbf{A}_2, \mathbf{B}_1 = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}, \mathbf{B}_2 = 0$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \begin{bmatrix} \frac{R_o}{R_o + R_C} & \frac{R_o R_C}{R_o + R_C} \end{bmatrix}, \mathbf{E}_1 = \mathbf{E}_2 = 0 \quad (2.9)$$

Finally, the state space matrices during *Off* duration can be written as:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}_2\mathbf{x}(t) + \mathbf{B}_2V_{in} \\ y &= \mathbf{C}_2\mathbf{x}(t) + \mathbf{E}_2V_{in}\end{aligned}\quad (2.10)$$

2.4 Model Simulation

To investigate the behaviour of the aforementioned buck model, the derived differential equations presented in section (2.3) have been simulated using MATLAB/Simulink. The power load of the designed dc-dc buck converter is for 5 W operations. The following circuit parameters are used: $L = 220 \mu\text{H}$, $C = 330 \mu\text{F}$, $R_o = 5 \Omega$, $R_L = 63 \text{ m}\Omega$, $R_C = 25 \text{ m}\Omega$, $V_{in} = 10 \text{ V}$, and the switching frequency is 20 kHz. These parameters are calculated using design notes available from Microchip^(TM) [22]. Fig. 2.3 and Fig. 2.4 shows the open loop output voltage and inductor current at 33% duty-cycle and $V_{in} = 10 \text{ V}$. As displayed in the waveforms of Fig. 2.3 and Fig. 2.4, the steady state DC output voltage and the inductor are evidently content periodic ripples that are repeated at each switching period. Normally, the power stage elements (L , C) determine the magnitude of the ripple as shown in the waveforms.

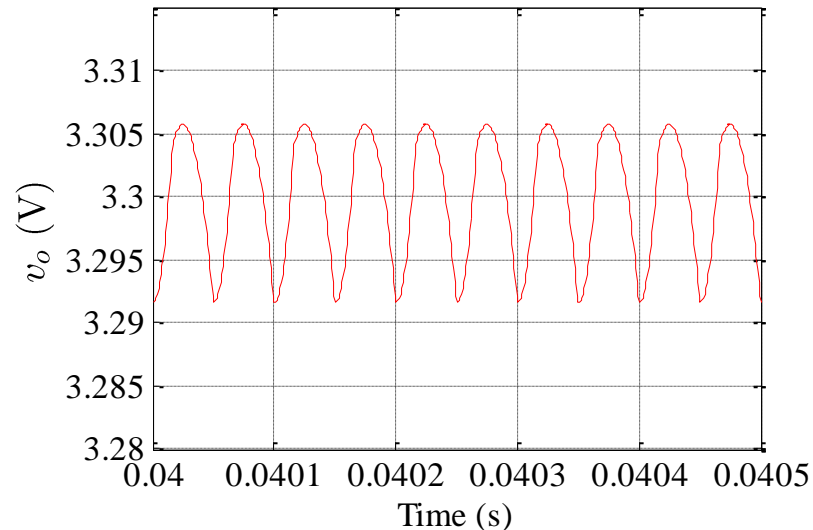


Fig. 2.3 Open loop steady state output voltage

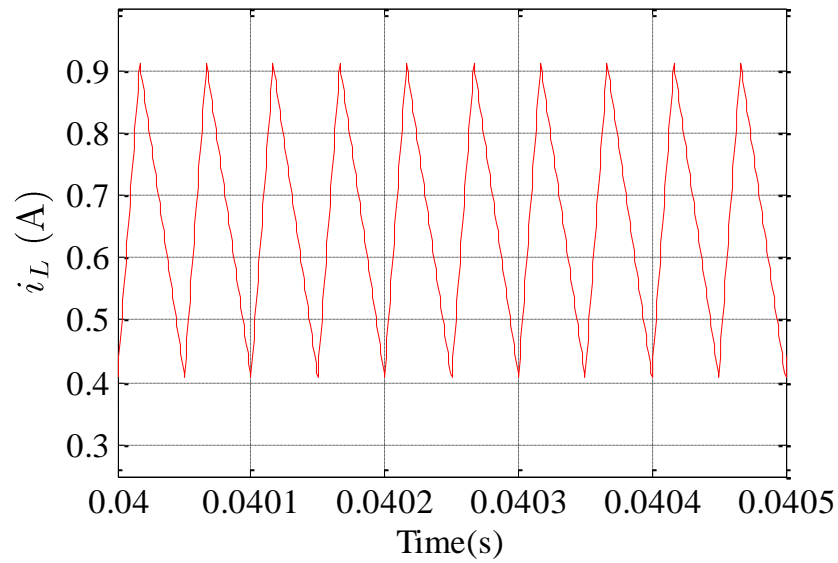


Fig. 2.4 Open loop steady state inductor current

2.5 Buck State Space Average Model

The state space average model is the most common approach to obtain the linear time invariant (LTI) system of SMPC. The strategy starts by averaging the converter's waveforms (inductor current and capacitor voltage) over one switching period to produce the equivalent state space model. In this way, the switching ripples in the inductor current and capacitor voltage waveforms will be removed [23]. As demonstrated in the previous section, there are two LTI differential equations to describe the operation of buck dc-dc converter (*On* and *Off* intervals). By averaging these two state intervals, the state space average model can be obtained. This is achieved by multiplying the *On* interval (2.6) by $d(t)$ and the *Off* interval (2.10) by *Off* time duration [$d'(t) = 1 - d(t)$]. This yields the following state space average model [18]:

$$\dot{\mathbf{x}} = (d(t)\mathbf{A}_1 + [1 - d(t)]\mathbf{A}_2)\mathbf{x}(t) + (d(t)\mathbf{B}_1 + [1 - d(t)]\mathbf{B}_2)V_{in} \quad (2.11)$$

$$\mathbf{y}(t) = (d(t)\mathbf{C}_1 + [1 - d(t)]\mathbf{C}_2)\mathbf{x}(t) + (d(t)\mathbf{E}_1 + [1 - d(t)]\mathbf{E}_2)V_{in}$$

where, d denotes the *On* time length.

Once the average state space model of the buck converter is defined, it is possible to apply the Laplace transform for obtaining the frequency domain linear time model.

This model is essential in the linear feedback control design, such as the root locus control approach. In voltage mode control of the SMPC, the control-to-output voltage transfer function (2.12) [24, 25] plays the important role of describing the locations of poles/zeros for optimal voltage response. The control-to-output model can be computed by applying the Laplace transform to the small signal average model of SMPC in equation (2.11) and then solving the system with respect to output voltage. This research is primarily focused to utilise this model in the system identification and the power converter control design.

$$G_{dv}(s) = \frac{V_{in}(CR_C s + 1)}{s^2 LC \left(\frac{R_o + R_C}{R_o + R_L} \right) + s \left(R_C C + C \left(\frac{R_o R_L}{R_o + R_L} \right) + \frac{L}{R_o + R_L} \right) + 1} \quad (2.12)$$

As expressed in (2.12) the control-to-output transfer function of the buck SMPC exhibits a general form of second order transfer function and generally it can be written as [1, 18]:

$$G_{dv}(s) = G_o \frac{1 + \frac{s}{w_{zesr}}}{1 + \frac{s}{Qw_o} + \left(\frac{s}{w_o} \right)^2} \quad (2.13)$$

where, the corner frequency (w_o) of the buck converter, the quality factor (Q), the zero frequency (w_{zesr}), and the dc gain (G_o) can be defined as follows [26]:

$$w_o = \sqrt{\frac{R_o + R_L}{LC(R_o + R_C)}} \quad (2.14)$$

$$Q = \frac{1}{w_o \left(R_C C + \frac{L}{R_o + R_L} + \frac{R_o R_L C}{R_o + R_L} \right)}$$

$$G_o = V_{in} = \frac{V_o}{D}$$

$$w_{zesr} = \frac{1}{CR_C}$$

From equation (2.13), it can be observed that the control-to-output voltage transfer function of the buck converter contains two poles and one zero. The locations of the poles as well as the dynamic behaviour of the dc-dc converter are mainly dependent upon the quality factor (Q) and the angular resonant frequency (ω_o) of the converter. In the time domain, the quality factor gives indication of the amount of overshoot that occurs during a transient response. This factor is inversely related to the damping ratio (ζ) of the system [27, 28]:

$$M_p \approx e^{-\frac{\pi}{2Q} \sqrt{1 - \frac{1}{4Q^2}}}, \quad Q = \frac{1}{2\zeta} \quad (2.15)$$

Here, M_p is the maximum peak value.

It is worth noting that a non-negligible resistance of the output capacitor (R_C) of the dc-dc converter introduces a zero in the control-to-output voltage transfer function of the SMPC as given in (2.13). The location of this zero has a negative impact on the dynamic behaviour of the SMPC. In order to cancel the effect of this zero and improve the system performance, a constant pole in the control loop may be added. This pole can be placed at the same value as the ESR zero.

2.6 Discrete Time Modelling of Buck SMPC

In order to derive the discrete model of SMPC, the continuous time dynamic model in (2.6) and (2.10) should first be defined. Then, by sampling the states of the converter at each time instant, the continuous time differential equations are transformed into a discrete time model. A discrete time model is necessary for digital implementation of the algorithms. In the literature, different techniques have been proposed for discrete time modelling of dc-dc converters and for obtaining the control-to-output transfer function [21, 29]. However, these techniques including the direct transformation methods (bilinear transformation, zero-order-hold transformation, pole-zero matching transformation, *etc.*) from s -to- z domain are generally describe the buck SMPC as a second order IIR filter (2.16), for example the literature that have been presented in [1, 5, 21, 30-33].

$$G_{dv}(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}, N = M = 2 \quad (2.16)$$

However, a zero-order-hold (ZOH) transformation approach is preferred for discrete time modelling of the control-to-output transfer function (2.17). Practically, the sampled data signals are acquired based on sample and hold process followed by A/D operation. In addition, the control signal remains constant (held) during the sampling interval and is modified at the beginning of each updated cycle [30]. Therefore, both the control and output signals are based on ZOH operation. Consequently, a ZOH transformation method is utilised in this work. The authors in [30] and [31] use the ZOH transformation method to model the $G_{dv}(z)$ and then to be used in the system identification process. Recently, system identification techniques have been extensively used in dc-dc converters for discrete time modelling of small signal control-to-output transfer function. This is typically accomplished by superimposing the duty command with a small amplitude signal. The frequency components and the amplitude are then estimated through different identification methods. Finally, the frequency response control-to-output LTI transfer function can be constructed. Other approaches involve by directly identifying the z -domain transfer function using different parametric identification techniques such as the RLS algorithm.

$$G_{dv}(z) = (1 - z^{-1})Z\left\{\frac{G_{dv}(s)}{s}\right\} \quad (2.17)$$

2.7 Digital Control Architecture for PWM DC-DC Power Converters

Digital controllers have been increasingly used in different fields and have recently become widely utilised in the control design of SMPCs. The use of digital controllers can significantly improve the performance characteristic of dc-dc converters for several reasons. Firstly, digital controllers provided more flexibility in the design compared with the analogue controllers. Secondly, they can be implemented with a small number of passive components, which reduce the size and cost of design. Also, digital controllers have low sensitivity on external disturbances and system parameter

variations. In addition, digital controllers are easy and fast to design, as well as to modify or change the control structures or algorithms. Furthermore, it enables advance control algorithms to be implemented, such as non-linear control, adaptive control, and system identification algorithms. Finally, programmability; the algorithms can easily be changed and reprogrammed [34-36]. On the other hand, the power processing speed is faster in an analogue controller than in a digital controller; this is due to the limitations in the microprocessors speeds. Furthermore, the system bandwidth is higher in analogue design compared with the digital design. In addition, no quantisation effects are considered in analogue systems [37, 38]. However, in order to stabilise the output voltage at the desired level, the control signal must be varied and accommodate any changes in the system, such as load changes or the variations in the input voltage. This can be performed by designing an appropriate feedback controller for appropriate control signal generation.

There are two common control structures applied in the closed loop control design of the dc-dc power converters: voltage mode control and current-mode control. Digital voltage mode controllers are mostly used and preferred in the industry over current-mode controllers [10-12]. This is because the current-mode controllers require an additional signal condition circuit, consisting of a high speed current sensor; in consequence this will incur extra costs to the system [39]. In addition, a voltage mode control is simple to design. Therefore, this research will concentrate on the design and implementation of the digital voltage mode control for SMPCs.

2.7.1 Digital Voltage Mode Control

As illustrated in Fig. 2.5, the digitally controlled voltage mode scheme of SMPCs is divided into six sub circuit blocks. These circuits are categorised into two parts. The first part defines as an analogue system, including the dc-dc power processor stage, the gate drive, and the sensing/signal conditioning circuits. The second part classifies as the digital system, which is represented by the digital controller, and DPWM. The ADC block can be described as a mixed signal device.

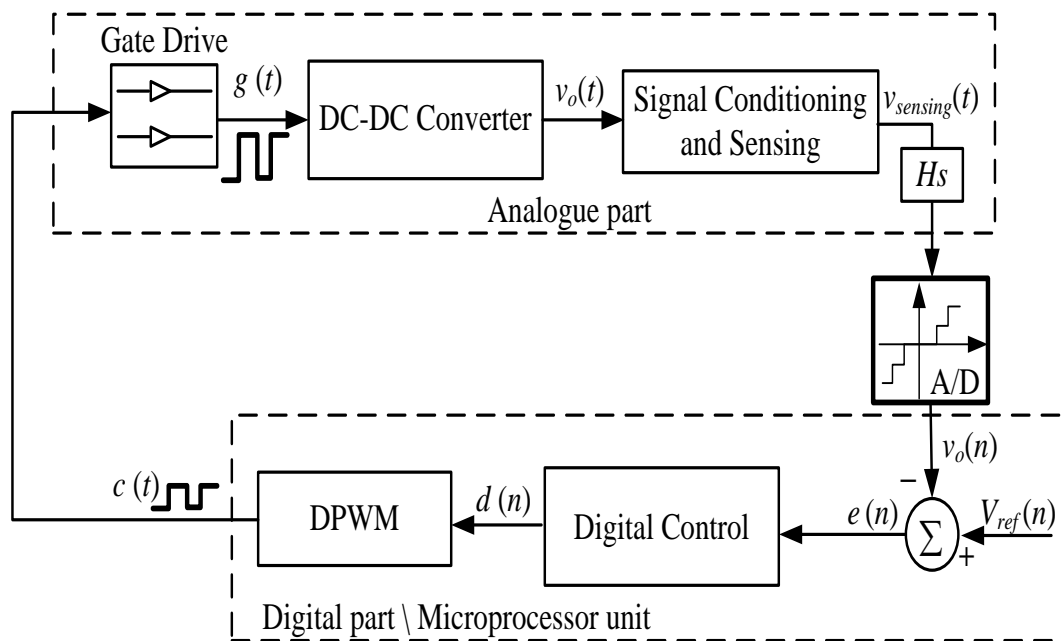


Fig. 2.5 Digital voltage mode control architecture of DC-DC SMPC

The output voltage generated from the dc-dc power converter is firstly sensed and scaled via a commonly used resistive voltage divider circuit with gain factor equal to H_s . Hence, any sensed voltage higher than the ADC full dynamic scale must be attenuated by a factor to be processed within the desired range. Other signal conditioning circuits can also be considered for suitable interfacing with ADCs. This includes different analogue circuits such as buffer circuits with wide bandwidth operation. An anti-aliasing filter is often used to filter the frequency content in the output voltage that is above half of the ADC sampling frequency (Nyquist criteria) [11]. Typically, this would be a low-pass filter.

The sensed output voltage ($v_{sensing}$) is digitised by the ADC. In digital control design for SMPC, there are two factors that must first be considered for the appropriate selection of an ADC:

1) The A/D number of bits or A/D resolution. This is important to the static and dynamic response of the controlled voltage of SPMC. The A/D resolution has to be less than the allowed variation in the sensed output voltage [40, 41].

2) The conversion time is an important factor in the selection of ADC as it dictates the maximum sampling rate of the ADC. In digitally controlled SMPCs, the conversion time is required to be small enough to achieve a fast response and high dynamic performance. Typically, the sampling time of an ADC is chosen to be equal to the switching frequency of the SMPC. This will ensure that the control signal is updated at each switching cycle.

The digital reference signal, $V_{ref}(n)$ is compared with the scaled sampled output voltage, $v_o(n)$. The resultant error voltage signal, $e(n)$, is then processed by the digital controller via its signal processing algorithm. A second order IIR filter is used as a linear controller that governs the output voltage of the SMPC as described in (2.18) and shown in Fig. 2.6 [20]. Generally, this IIR filter performs as a digital PID compensator as a central controller in the feedback loop. Both non-linear control and intelligent control techniques can also be applied for the digital control of SMPCs [24, 42-45].

$$G_c(z) = \frac{\sum_{i=0}^N q_i z^{-i}}{1 - \sum_{k=1}^M s_k z^{-k}} \quad (2.18)$$

However, the control signal, $d(n)$, is then computed on cycle-by-cycle basis. The desired duty ratio, $c(t)$, of the PWM is produced by comparing the discrete control signal with the discrete ramp signal; in the digital domain it is represented as a digital counter. Here, the DPWM performs as an interface circuit between the digital and analogue domains of the digitally controlled architecture within the SMPC,

simulating the purpose of the digital-to-analogue converter (DAC). Finally, the generated *On/Off* command signal across the DPWM is amplified by the gate drive circuit. The output of the gate signal is then used to activate the power switches of the SMPC.

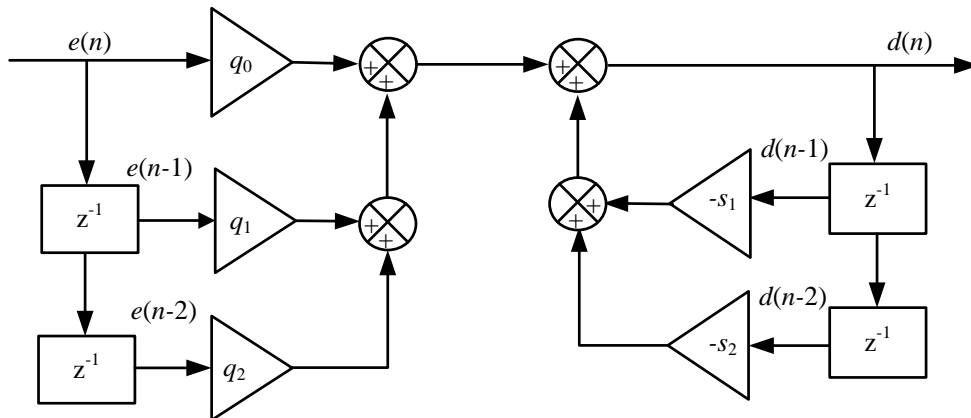


Fig. 2.6 Two-poles / Two-zeros IIR digital controller

It is worth noting that a high resolution DPWM is essential for the digital control of SMPCs. This will lead to accurate voltage regulation and avoid the limit cycle oscillation phenomenon. Limit cycles are defined as non-linear phenomena that occur in digital control of dc-dc converters during steady-state periods. In accordance to [46], the undesirable limit cycle oscillations in digitally controlled dc-dc converters can be avoided when the DPWM resolution is greater than the ADC resolution. Therefore, in order to eliminate the limit cycle oscillations, the resolution of DPWM has to be at least one bit greater than the ADC resolution [46]. Also, care is required in the selection of the integral gain in PID controllers, as excessively high values of integral gain can cause limit cycle oscillations around the steady-state value. For more rigorous details and analysis of the limit cycle oscillation, the reader should refer to the work presented by Peterchev and Sanders [46].

2.8 Digital Proportional-Integral-Derivative Control

The digital PID controller is well known and it is commonly used in control loop design of SMPCs [47]. This is because the PID control parameters are easy to tune and the designed controller is easy to implement. Generally, the discrete PID controller can be described as given by (2.19) [48]. Here, the PID controller is in

parallel form structure, where the control action is divided into three control signals as shown in (2.20) and the PID gains can be tuned independently.

$$G_c(z) = \frac{D(z)}{E(z)} = K_P + K_I \frac{1}{1-z^{-1}} + K_D(1-z^{-1}) \quad (2.19)$$

$$d(n) = d_P(n) + d_I(n) + d_D(n) \quad (2.20)$$

where:

$$\begin{aligned} d_P(n) &= K_P e(n) \\ d_I(n) &= K_I e(n) + d_I(n-1) \\ d_D(n) &= K_D [e(n) - e(n-1)] \end{aligned} \quad (2.21)$$

The variables K_P , K_I , and K_D , are the proportional-integral-derivative gains of PID controller, $e(n)$ is the error signal [$e(n) = V_{ref}(n) - v_o(n)$], and $d(n)$ is the control action. From (2.19), the discrete time domain of the PID controller can be described as shown in Fig. 2.7 and given in (2.22) and (2.23):

$$d(n) = d(n-1) + q_0 e(n) + q_1 e(n-1) + q_2 e(n-2) \quad (2.22)$$

$$\begin{aligned} q_0 &= K_P + K_I + K_D \\ q_1 &= -(K_P + 2K_D) \\ q_2 &= K_D \end{aligned} \quad (2.23)$$

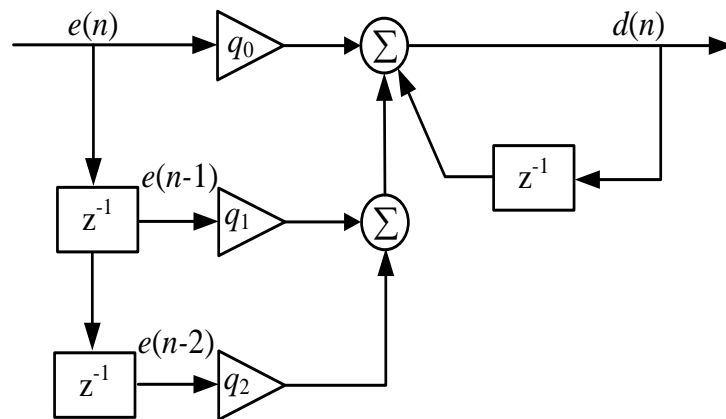


Fig. 2.7 Digital PID compensator

System performance, loop bandwidth, phase margin and gain margin are determined based on PID coefficients. For example, decreasing the steady-state error is achieved by the integral gain (K_I). However, the integral part will add a pole at the origin to the open loop transfer function of the system. This pole requires more consideration in the control loop design to ensure the system stability. In the frequency domain, the integral part acts as a low-pass filter, which makes the system less susceptible to noise. However, it adds a phase-lag to the system, which reduces the phase margin of the control loop, thus more oscillations can be observed in the output response [49, 50]. Therefore, the derivative part should be introduced in the control loop to increase the phase margin (phase-lead). This in turn leads to an improvement to the stability of the system and enhance the dynamic performance [51]. The derivative controller is responsible for the rate of change of the error signal. For instance, if the sensed output voltage of SMPC reaches the desired set point quickly, then the derivative part slows the rate of the change in the output control action [49]. Therefore, the derivative part can be considered as an intelligent part of the PID controller. However, the derivative part is more sensitive to the noise in the system [51], therefore the derivation of the error signal will amplify the noise in the control loop. Now, the proportional gain makes the output of the PID controller respond to any change of the error signal. For example, a small change in the error signal at high value of K_P results in a large change in the control action. In summary, the PID controller has the same scheme functionality of a phase lead-lag compensator [49, 50].

The parameters of the PID controller can be determined directly or indirectly. In the direct method, the discrete time model of SMPC and the PID controller are used, thus all the calculations are obtained in the z -domain. Therefore, a more accurate control loop can be achieved, where the errors related to the transformation approximation from the s -to- z domains are avoided in this approach [1, 5]. In the indirect approach, a continuous time domain of SMPC is utilised and the PID controller is designed in the s -domain. Different transformation methods can be applied to transfer the PID controller from the continuous domain to the discrete domain (s -to- z), such as the bilinear transform method, the backward Euler method, and the pole-zero cancellation method. However, inaccuracy in system performance

will be increased using this technique. This is due to the transformation approximation from s -to- z domain [52]. Duan *et al.* [37], demonstrated a systematic evaluation approach to compare the performance of the PID controller for SMPCs using four types of discretisation methods. The direct design approach has also been compared with the indirect method. It was discovered that that the direct method provides better performance compared with the indirect method. A similar conclusion was demonstrated by Al-Atrash and Batarseh [53]. In this research we are interested in two approaches that are commonly used in the digitally controlled design of SMPCs: the pole-zero matching approach [40, 54-56], which provides a simple discrete time difference equation [52], and the systematic pole placement method [47, 57-59].

2.8.1 Digital Control for Buck SMPC Based on PID Pole-Zero Cancellation

The design method presented in this section follows the same procedure demonstrated in [26, 54, 55]. The design starts from the continuous model of the buck dc-dc converter as described in (2.13). In order to cancel the two poles of power converter in (2.13), two zeros should be placed exactly at the same frequencies defined by w_o of the dc-dc power converter as given in equation (2.24). For simplicity of design, we assumed that $R_C = 0$:

$$G_c(s) = G_{co} \frac{1 + \frac{s}{Qw_o} + \left(\frac{s}{w_o}\right)^2}{s} \quad (2.24)$$

Therefore, the overall loop gain is reduced to only one pole at origin together with the dc gain:

$$L(s) = \frac{G_o G_{co}}{s} \quad (2.25)$$

From (2.24), it can be deduced that the design of the PID compensator using a pole-zero cancellation technique requires the precise knowledge of the power converter parameters, such as the quality factor and converter corner frequency [33]. This can be one of the drawbacks of this method, where the effect of any change in

the dc-dc converter parameters will directly influence the PID coefficients and in turn to the overall control loop. Therefore, an accurate parameters estimation is required for adequate control design [54]. For this reason, the authors in [33, 54, 55] choose the quality factor as a fixed value. As shown in (2.15), the quality factor is related to the damping factor (ζ). For an effective damping response, the damping factor is varied between 0.6 and 1.0 [28]. The resonant frequency of PID zeros in (2.24) is approximated to be at the same value of power converter corner frequency. As a result, the compensator zeros are assigned close to the converter poles; this will ensure system robustness. As a result, the overall loop gain can be written as [33]:

$$L(s) = G_o G_{co} \frac{\left(1 + \frac{2\xi s}{w_z} + \frac{s^2}{w_z^2}\right)}{s \left(1 + \frac{s}{Qw_o} + \frac{s^2}{w_o^2}\right)} \quad (2.26)$$

Here, G_{co} is the dc gain. This gain is selected to satisfy design requirements such as phase margin and gain margin. The root-locus method can be used to find G_{co} [55]. From [33, 54], the dc gain can be determined directly based on the desired loop bandwidth (2.27); in practice, the bandwidth chosen will be $f_b = f_s/10$ [33, 54].

$$G_{co} = \frac{2\pi f_b}{G_o} \quad (2.27)$$

Finally, by using the pole-zero matching transformation method the discrete PID gains described in (2.22) can be determined (*i.e.* q_0 , q_1 , and q_2).

2.8.1.1 Simulation design of a buck SMPC based on PID pole-zero Cancellation

In order to evaluate the PID cancellation method, the digital voltage mode control of a synchronous dc-dc buck SMPC circuit is simulated (Fig. 2.5). The circuit parameters of the buck converter are as follows: $L = 220 \mu\text{H}$, $C = 330 \mu\text{F}$, $R_o = 5 \Omega$, $R_L = 63 \text{ m}\Omega$, $R_C = 25 \text{ m}\Omega$, $V_{in} = 10 \text{ V}$, the switching frequency is $f_{sw} = 20 \text{ kHz}$, $H_s = 0.5$, and the sampling time $T_s = 50 \mu\text{s}$ ($f_s = f_{sw}$). A damping response of $\zeta = 0.7$ with the zero centre frequency is chosen as $w_z \approx w_o \approx 3723.5 \text{ rad / s}$. The damping factor and the zero centre frequency are then substituted into equation (2.24) to determine

the control transfer function. The s -to- z based MATLAB pole-zero matched method is used to obtain the discrete PID controller coefficients as written in equation (2.28). Accordingly, the PID gains (Fig. 2.7) are optimally tuned to: $q_0 = 4.127$, $q_1 = -7.184$, and $q_2 = 3.182$.

$$d(n) = d(n-1) + 4.127e(n) - 7.184e(n-1) + 3.182e(n-1) \quad (2.28)$$

Fig. 2.8 displays the frequency response of the controlled system; here it shown that the phase margin of the compensated system is 41.1° and the gain margin is 12.6 dB. Fig. 2.9 shows the root locus of the power converter stage. The locations of the PID roots are presented in Fig. 2.10, and the root locus cancellation paths for the control loop are illustrated in Fig. 2.11. It can be seen that the two poles of the dc-dc converter are cancelled with very short paths by the two matched zeros of the PID controller. The PID compensator is set to control the buck converter output voltage at 3.3 V. The transient behaviour of the system is examined by abruptly changing the load of the SMPC. Fig. 2.12 demonstrates the transient response of the designed PID compensator when the load is rapidly switched between 0.66 A-to-1.32 A. It can be noted that there is a small overshoot in the system at step load changes, however the response quickly recovers to the desired value; this verifies the successful design of the digital compensator.

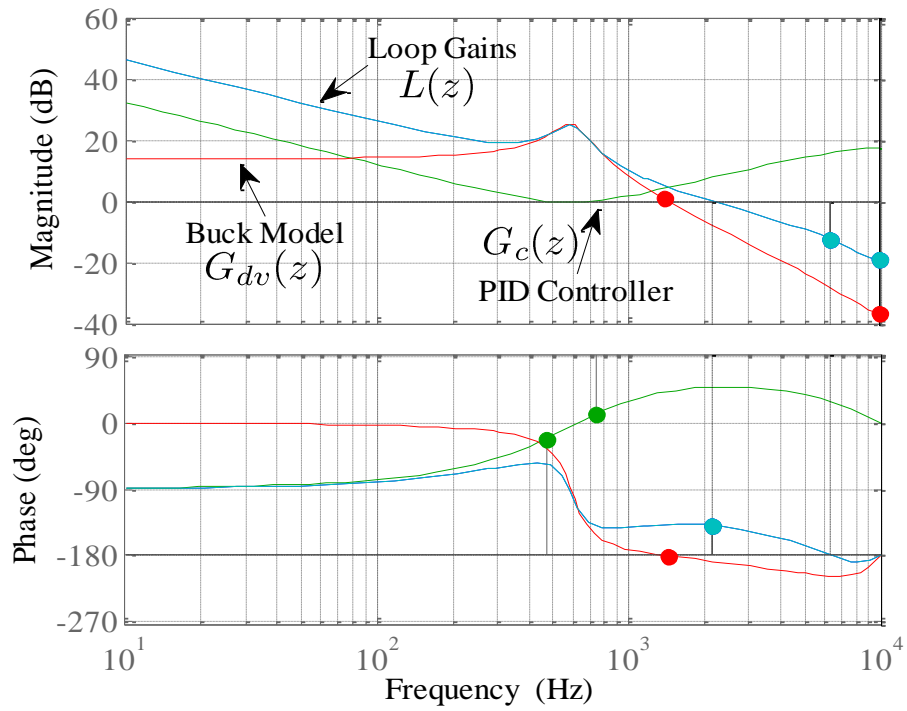


Fig. 2.8 Frequency response of the compensated and uncompensated dc-dc buck SMPC

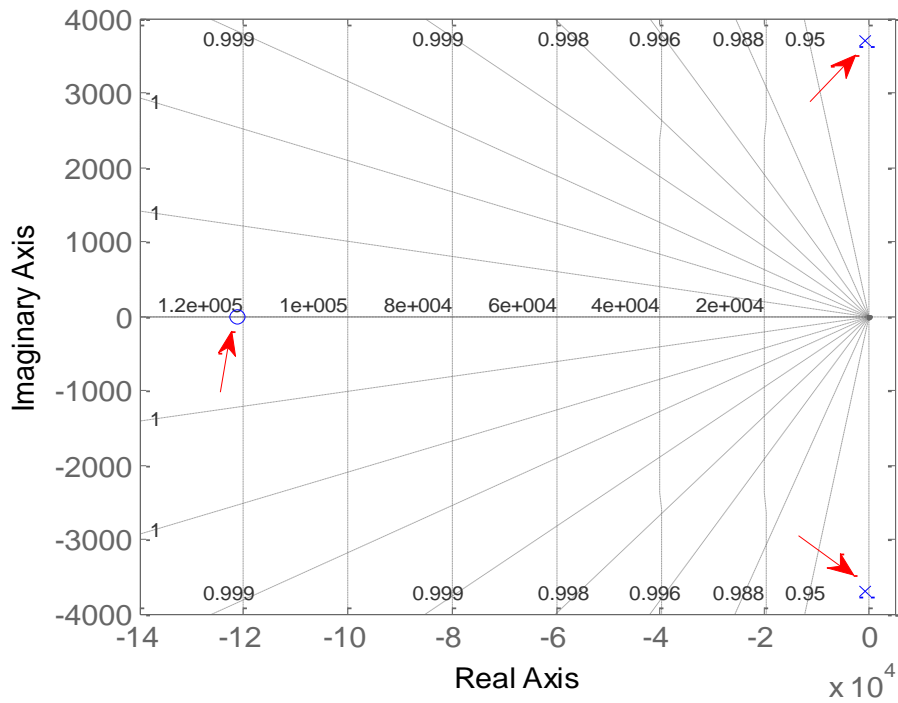


Fig. 2.9 Power stage root locus

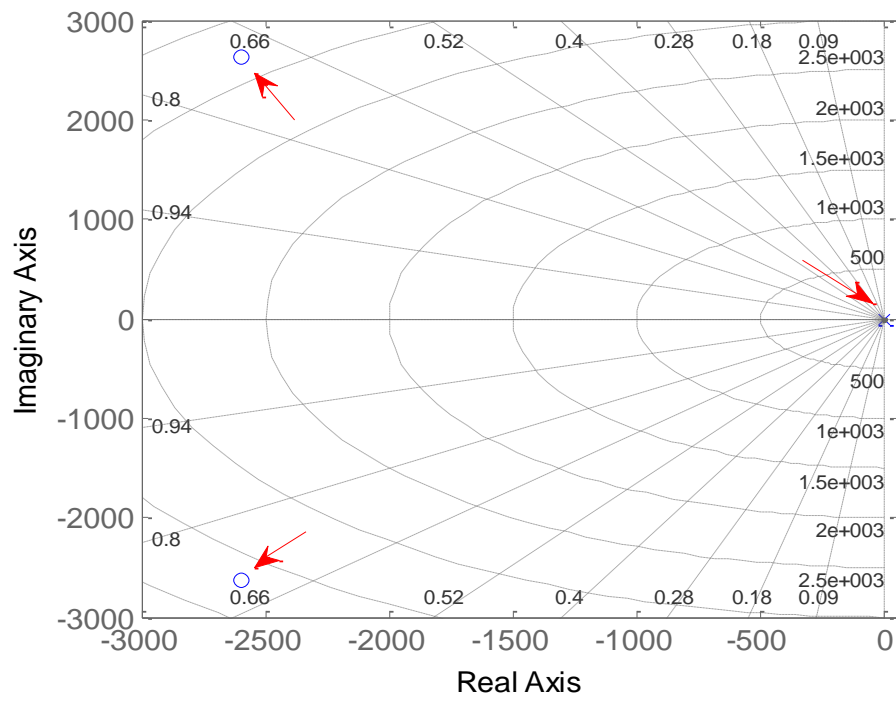


Fig. 2.10 PID compensator root locus

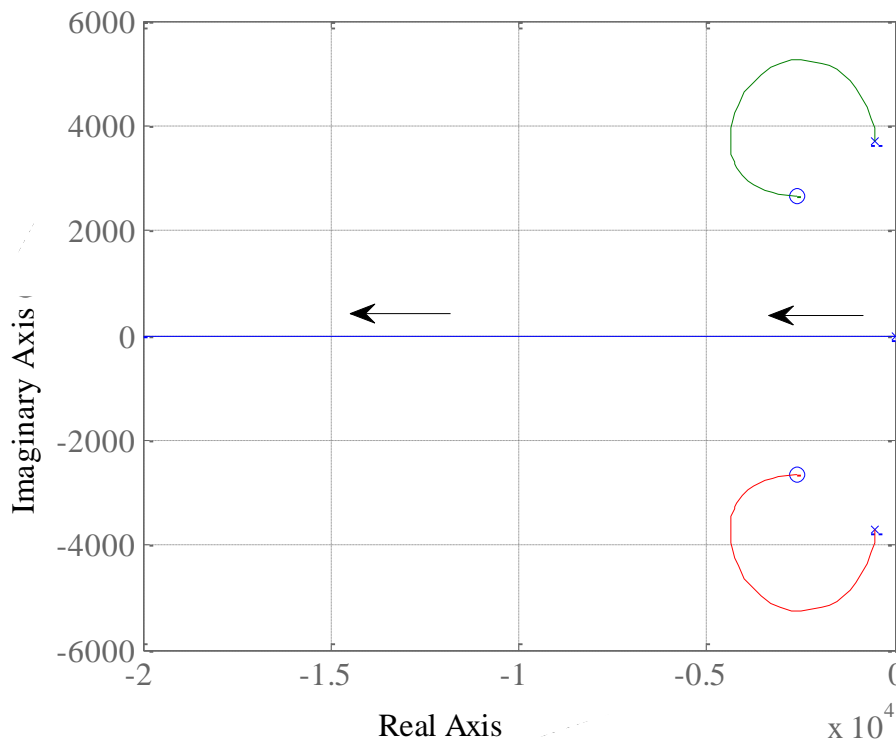
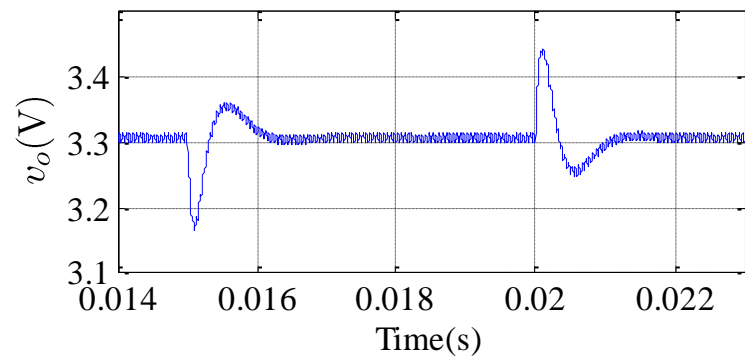
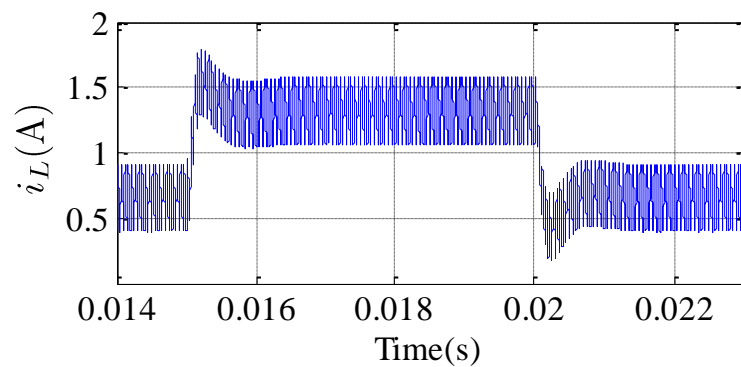


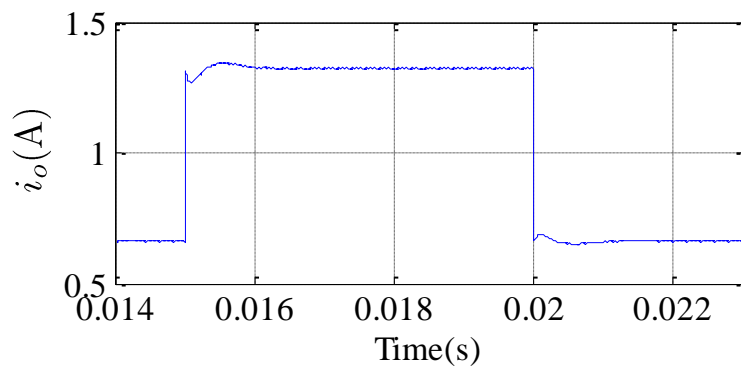
Fig. 2.11 Total loop gains root locus



(a)



(b)



(c)

Fig. 2.12 Transient response of the PID controller, a: output voltage, b: inductor current, c: load current. Load current change between 0.66 A and 1.32 A every 5 ms

2.8.2 Pole Placement PID Controller for DC-DC Buck SMPC

In the pole placement approach, a discrete control-to-output model of the buck converter is utilised (2.29) and the digital PID controller can be described as written in (2.30) [60]. In this case, a two poles/two zeros discrete PID controller (2.30) will be introduced for the digital control of the buck dc-dc converter. Equation (2.31) represents the discrete difference equation form of (2.30).

$$G_{dv}(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (2.29)$$

$$G_c(z) = \frac{\beta_o + \beta_1 z^{-1} + \beta_2 z^{-2}}{(1 - z^{-1})(1 + \alpha z^{-1})} = \frac{\beta(z)}{\alpha(z)} \quad (2.30)$$

$$d(n) = \beta_o e(n) + \beta_1 e(n-1) + \beta_2 e(n-2) + (1 - \alpha)d(n-1) + \alpha d(n-2) \quad (2.31)$$

As shown in Fig. 2.13, the closed loop control transfer function can be written as follow [57, 60]:

$$G_L(z) = \frac{B(z)\beta(z)}{A(z)\alpha(z) + B(z)\beta(z)} \quad (2.32)$$

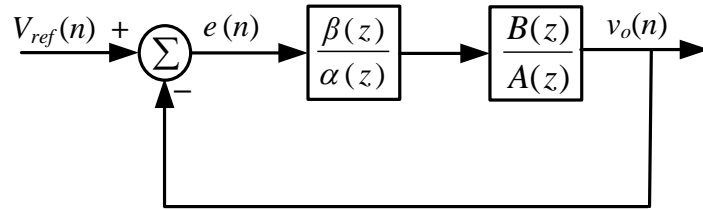


Fig. 2.13 Closed loop control of the buck SMPC

The desired closed loop dynamic of the system can be used to solve the relation in the denominator polynomial expressed in (2.32). In this way, the locations of the closed loop poles are set according to the desired values, unlike other control techniques which required tuning of the control coefficients for acceptable response [28]. The characteristics equation of (2.32) can be formulated as [57, 60]:

$$\begin{aligned}
D(z) = A(z)\alpha(z) + B(z)\beta(z) &= 1 + \sum_{k=1}^{N_d} d_k z^{-k} \\
&= 1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{N_d} z^{-N_d}, N_d \leq 4
\end{aligned} \tag{2.33}$$

In a second order model, such as a dc-dc buck converter, the second order characteristic equation [57, 58] is often utilised to describe the desired closed loop dynamics of the system:

$$G(s) = s^2 + 2\xi w_n s + w_n^2 = 0 \tag{2.34}$$

Therefore, the dynamic characteristic of a closed loop control may be given as in (2.35) [60]. As presented in section 2.5 the dynamics behaviour is defined by the damping factor and the natural frequency. These factors should be selected appropriately for better performance and adequate damping response.

$$\begin{aligned}
d_1 &= -2e^{-\xi w_n T_s} \cos\left(w_n T_s \sqrt{1-\xi^2}\right) \\
d_2 &= e^{-2\xi w_n T_s}
\end{aligned} \tag{2.35}$$

To determine the parameters of the control system, the sets of linear algebra equations are required. This can be obtained by rewriting equation (2.33) in matrix form (2.36). Thus, the parameters of the PID controller can be solved as described in (2.37).

$$\begin{bmatrix} b_1 & 0 & 0 & 1 \\ b_2 & b_1 & 0 & a_1 - 1 \\ 0 & b_2 & b_1 & a_2 - a_1 \\ 0 & 0 & b_2 & -a_2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \alpha \end{bmatrix} = \begin{bmatrix} d_1 + 1 - a_1 \\ d_2 + a_1 - a_2 \\ a_2 \\ 0 \end{bmatrix}, N_d = 2 \tag{2.36}$$

$$\begin{aligned}
\beta_o &= \frac{1}{b_1}(d_1 + 1 - a_1 - \alpha) \\
\beta_1 &= \frac{a_2}{b_2} - \beta_2 \left(\frac{b_1}{b_2} - \frac{a_1}{a_2} + 1 \right) \\
\beta_2 &= \frac{s}{r} \\
\alpha &= \beta_2 \frac{b_2}{a_2} \\
s &= a_2([b_1 + b_2][a_1 b_2 - a_2 b_1] + b_2[b_1 d_2 - b_2 d_1 - b_2]) \\
r &= [b_1 + b_2][a_1 b_1 b_2 + a_2 b_1^2 - b_2^2]
\end{aligned} \tag{2.37}$$

It can be noticed that the solution of (2.36), necessitates a matrix inversion operation to find the control parameters. Consequently, a high computational load is involved with on-line updates of the control loop [57]. Therefore, the pole-placement method is more applicable for off-line control design. This is clearly demonstrated by Shuibao *et al.* [58], where the off-line design based on pole-placement approach is used to control the SMPC. Kelly and Rinne [57] presented a direct method to design a digital control of a dc-dc buck converter based on pole-placement technique. It was discovered that the zeros of the pole placement controller can only be used to fully control the dc-dc buck converter. The resultant control structure may be compared to a PD controller. Whilst this controller is computationally efficient, it actually only applies a PD compensator which can yield a non-zero steady-state error.

2.8.2.1 Simulation design of a buck SMPC based on pole-placement PID controller

Similar parameters to those outlined in section (2.8.1.1) are chosen for the SMPC circuit. The natural frequency is selected to be twice the corner frequency of the power converter $w_n = 2w_o = 7447$ rad/s, with damping factor $\xi = 0.7$ [57]. By using (2.35) and (2.37), the PID parameters are: $\beta_o = 4.672$, $\beta_1 = -7.539$, $\beta_2 = 3.184$, and $\alpha = 0.3747$. Therefore, the discrete PID controller can be given as:

$$\begin{aligned}
d(n) &= 4.672e(n) - 7.539e(n-1) + 3.184e(n-1) \\
&\quad + 0.6253d(n-1) + 0.3747d(n-2)
\end{aligned} \tag{2.38}$$

Fig. 2.14 displays the frequency response of the controlled system; here it is shown that the phase margin of the compensated system is 35.7° and the gain margin is 14.8

dB. It is important to note that a phase margin greater than 40° is essential for a robust SMPC control system [61]. Accordingly, more tuning steps are required to increase the phase margin and improve the bandwidth of the closed loop system.

To investigate the transient characteristic of the system, repetitive step load changes have been applied to the dc-dc converter. Fig. 2.15 presents the transient performance of the feedback system when the current load alternates between 0.66 A-to-1.32 A. As expected from the frequency response results, a poorly dynamic response will be observed by the designed feedback controller. Fig. 2.16 compares the loop gains of the pole-placement controller with the pole-zero cancellation approach. Clearly, a pole-zero cancellation compensator achieves a higher phase margin and loop bandwidth compared with the pole-placement compensator. As a result, a better response is achieved with the pole-zero method, which demonstrates a smaller overshoot and undershoot on the output voltage, as well as a faster recovery time observed during load changes (Fig. 2.17). Therefore, this project has utilised the pole-zero cancellation approach in the control design of dc-dc buck SMPC.

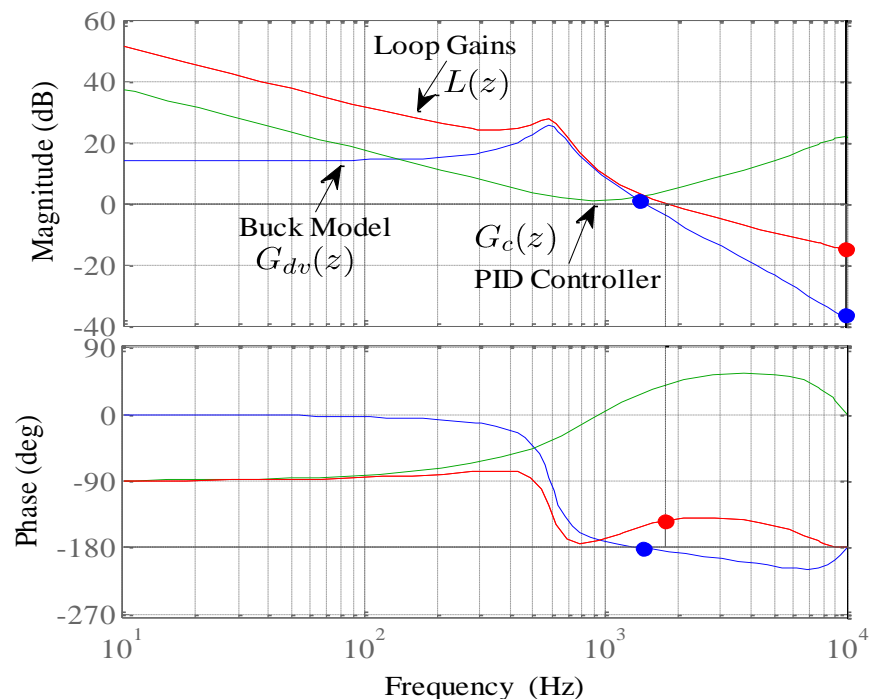
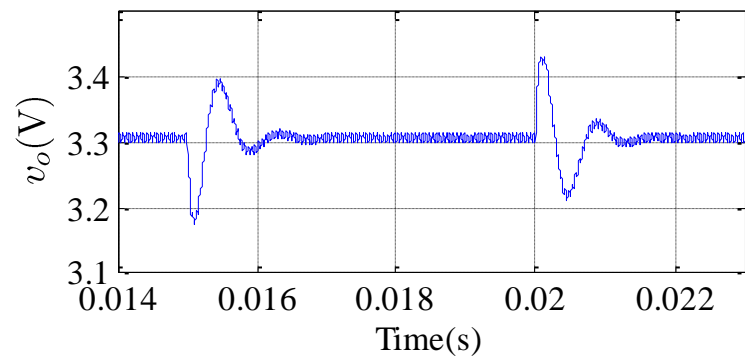
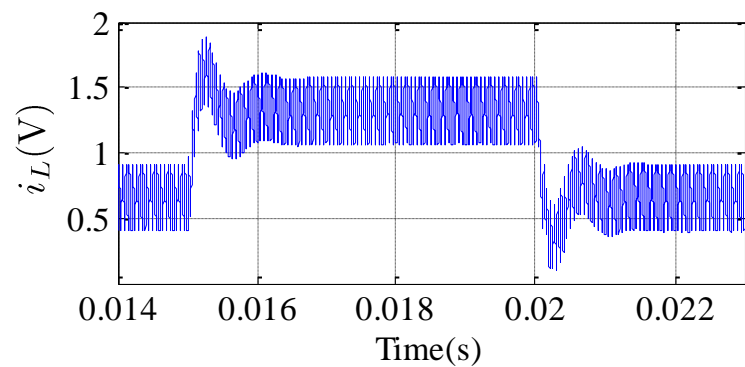


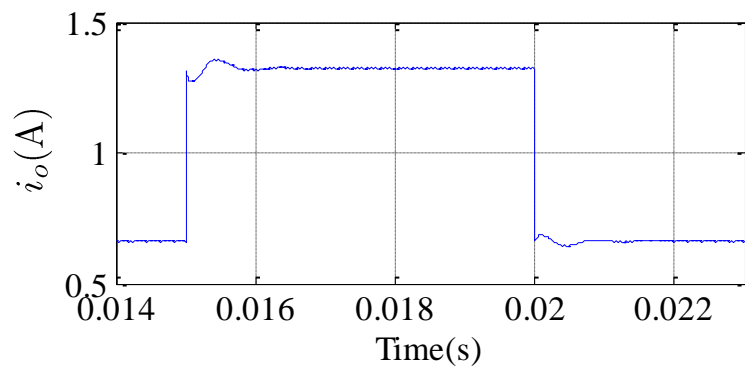
Fig. 2.14 Frequency response of the compensated and uncompensated dc-dc buck SMPC



(a)



(b)



(c)

Fig. 2.15 Transient response of the pole-placement PID controller, a: output voltage, b: inductor current, c: load current. Load current change between 0.66 A and 1.32 A every 5 ms

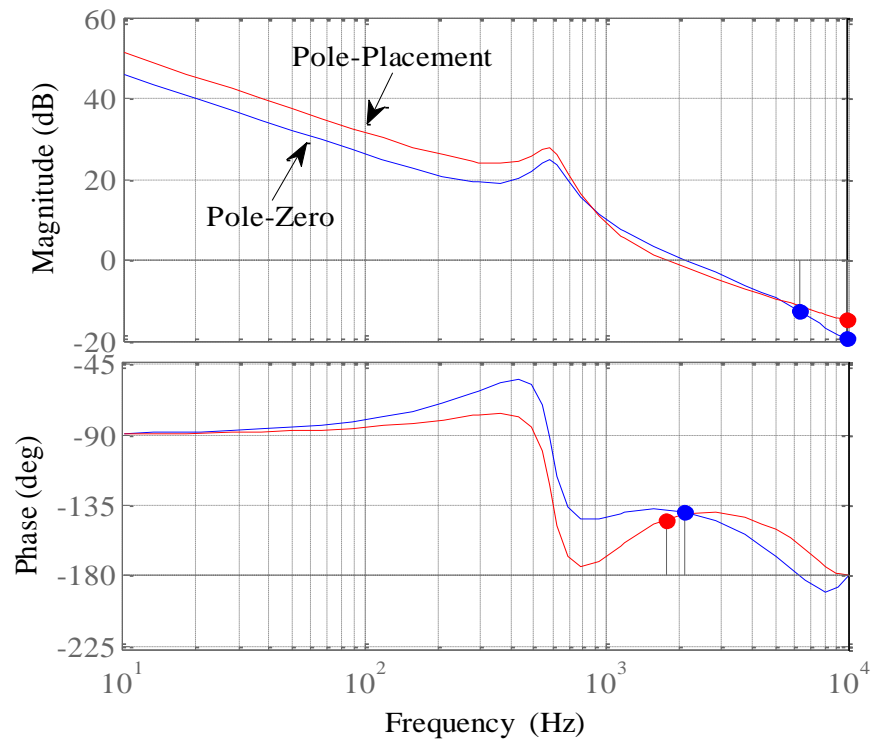


Fig. 2.16 Loop-gain comparison between pole-placement and pole-zero PID controllers

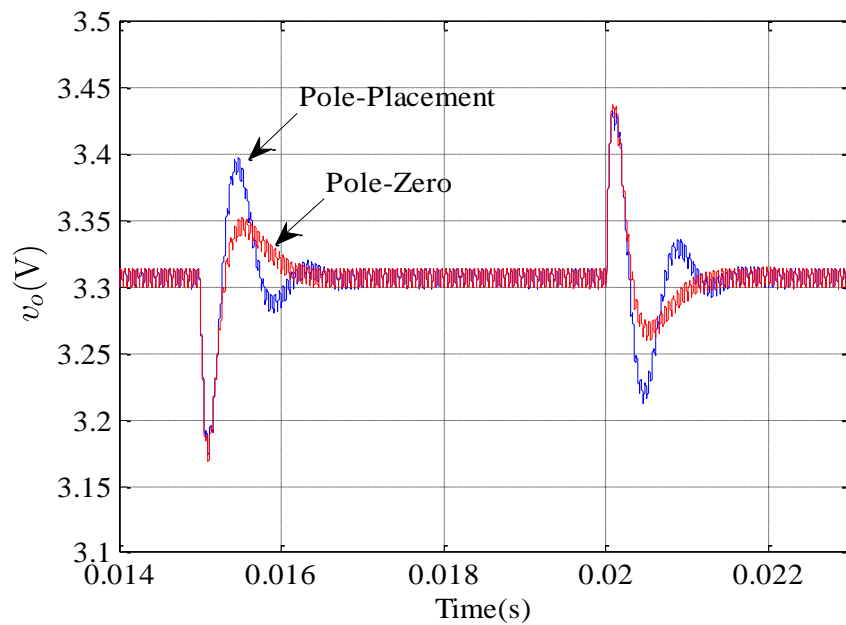


Fig. 2.17 Comparison of transient response results between pole-placement and pole-zero PID controllers. Repetitive load current change between 0.66 A and 1.32 A every 5 ms

2.9 Chapter Summary

Details and analysis of the modelling and control of the dc-dc power converters were introduced in this chapter. Common circuit topologies of dc-dc converters with focus on the buck dc-dc converter configuration and circuit operation were demonstrated. The mathematical modelling in continuous and discrete time domain of the buck SMPC was explained. In addition, chapter 2 provided information on the actual linear state space and linear average model of buck dc-dc converters, with most of the emphasis on the modelling of the control-to-output voltage transfer function of dc-dc buck converter. Therefore, the digital voltage mode control architecture of the buck dc-dc SMPC was demonstrated and an overview of each block in this structure was highlighted. For the digital control of the buck SMPC, two techniques of control loop design were explained: the pole-zero cancellation method and the pole-placement approach. Finally, the proof of concepts for the most important aspects were analysed and simulated.

Chapter 3

SYSTEM IDENTIFICATION, ADAPTIVE CONTROL AND ADAPTIVE FILTER PRINCIPLES -A LITERATURE REVIEW

3.1 Introduction

This chapter presents an overview of recent research in the area of SMPC control. Three topics in this field are specifically considered: system identification, adaptive control and adaptive filtering. The first part of this chapter describes the different methods used in system identification. It explains the difference between parametric and non-parametric estimation techniques and clearly explains the rationale for choosing a parametric approach in this work. The second part of the chapter presents a general introduction to adaptive control and adaptive filtering. The chapter concludes by considering the use of these digital techniques in state of the art solutions for system identification and adaptive control of dc-dc SMPC applications.

3.2 Introduction to System Identification

System identification has been widely used in a plethora of scientific fields and has become essential in the area of signal processing and adaptive/self-tuning control systems (automatic controllers).

The objective of system identification is to capture the dynamic behaviour of the system based on measured data [62]. In a rigorous mathematical sense, system identification entails the construction of the mathematical model that most closely resembles the dynamic characteristic of the system based on observed data [63]. A signal with enriched frequency content is injected into the system, which, along with the measurement of the resultant output, is processed to produce the system model.

This constitutes the underlying principle of a system identification process. Typically, the plant is treated as a black-box model and when the error between the real system and corresponding model output is minimised, an accurate model of the system can be derived (Fig. 3.1) [64]. Many control approaches rely on an accurate model of the system, often represented as a transfer function, to design a robust controller. For example, the pole placement technique is immensely inadequate without the transfer function of the process to successfully modify the location of poles and zeros in order to meet the design requirements [28].

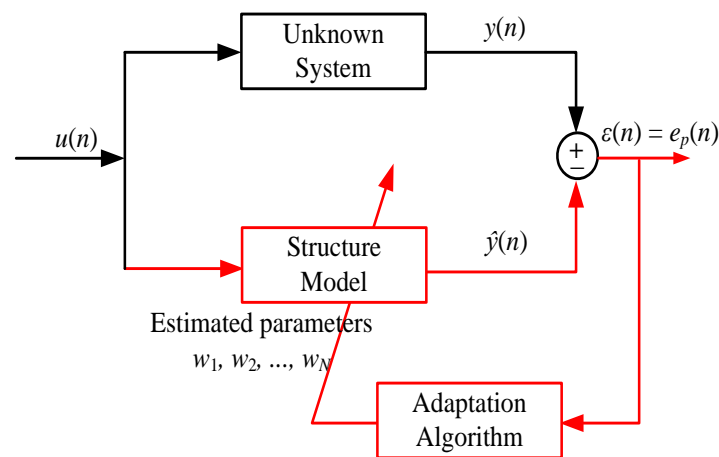


Fig. 3.1 General block diagram of parametric identification

Two broad categories of system identification exist, namely on-line and off-line estimation techniques [64].

- a) In the on-line paradigm, the obtained data in real-time is used to estimate the parameters of the model. RLS is the most recognisable method of on-line system identification [64]. The automatic control scheme incorporates this approach to adapt the controller gains at each sample period. This is accomplished in two phases. In the initial step, the system performance will be monitored and the dynamic characteristics of the closed loop system will actively be identified, providing a real-time estimation of the model parameters. In the second step, the control parameters are fine-tuned according to the uncertainties of the system and this results in a profound improvement of the dynamic performance of the system [54].

- b) In the off-line estimation, the measured data is stored in the memory; a typical approach is to use a block array of memory. Then the batch of observed signals is processed to construct the system model and this process is called batch estimation [64]. Generally, this scheme is preferred when the requirement is to model a highly complicated system. The estimated model is then used to design the desired controller. This can be achieved by firstly constructing the model of the system, relying only on experimental data, and then by determining the controller parameters based on the estimated model. Non-mathematical assumption is required in this approach; therefore, optimal control parameters can be calculated using the off-line estimation method.

It is worth noting that both schemes can be applied to estimate specific parameters in the system; for instance, corner frequency (w_n) and damping factor (ζ)/quality factor (Q) are the valuable parameters to identify in SMPC application.

3.3 Parametric and Non-Parametric Identification

The linear model of a system can be determined using two different techniques: 1) Non-parametric estimation techniques, 2) Parametric estimation techniques [28].

Non-parametric methods often use transient response analysis or correlation analysis to estimate the impulse response of the system, or use frequency analysis and spectral analysis to estimate the frequency response of the system, without using model parameters. Algorithms such as the Fourier Transform (FFT) can be used to construct the non-parametric model of the system. The main advantage of non-parametric estimation techniques is that no prior knowledge of the model is required to estimate the system dynamics. In addition, the level of complexity of non-parametric methods is comparatively manageable for effective implementation [63, 65].

Non-parametric methods are more sensitive to noise and an appropriate excitation signal is required to accomplish accurate estimation. Therefore, long sequences of captured data are essential for noise immunity and data accuracy [30]. Consequently, the identification process can take a significant amount of time to complete. This in turn, restricts a schemes ability to identify rapid system changes, such as abrupt load

changes in SMPCs. Also, it hinders the continuous iterative estimation of the system model, which is an imperative necessity for adaptive control design. Significant hardware resources may also be required in terms of processing power and memory [66]. Furthermore, inaccuracies in the estimated parameters potentially may be increased in the discrete time domain. This is attributable to approximations occurring on transformations from the s -to- z domain, and effects of quantisation error [30]. In addition, it can be difficult to apply transient response analysis or correlation based techniques for closed loop non-parametric estimation. This is because in closed loop systems, the output has an impact on the input signal to the system due to the feedback loop. Therefore, any assumption of non-correlation between the input/output signals is not valid [28]. This is clearly described in (3.2) and (3.3); here the sampled input signal $u(n)$ and the disturbance signal $v(n)$, such as measurement noise should be non-correlated to satisfy condition (a) below for accurate impulse response estimation [65]. As shown in Fig. 3.2, the linear time invariant discrete system can be expressed as [67]:

$$y(n) = \sum_{k=1}^{\infty} g(k)u(n-k) + v(n) \quad (3.1)$$

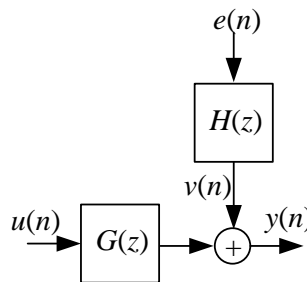


Fig. 3.2 General linear model transfer function

Here, $u(n)$ is the sampled input signal, $y(n)$ is the discrete output signal, $g(n)$ is the discrete impulse response of the system, and $h(n)$ is the discrete impulse response of the noise, $e(n)$, and $v(n)$ is the disturbance signal. Starting from (3.1), the cross-correlation between input $u(n)$ and output $y(n)$ can be described as:

$$\mathbf{R}_{uy}(m) = \sum_{n=1}^{\infty} u(n)y(n+m) = \sum_{n=1}^{\infty} g(n)\mathbf{R}_{uu}(m-n) + \mathbf{R}_{uv}(m) \quad (3.2)$$

where, $\mathbf{R}_{uu}(m)$ is the auto-correlation of $u(n)$ and $\mathbf{R}_{uv}(m)$ is the cross-correlation between the input and the disturbance. Two conditions should be considered for valid non-parametric estimation of the impulse response [68]:

a) The input $u(n)$ and disturbance $v(n)$ are uncorrelated, therefore $\mathbf{R}_{uv}(m) = 0$.

b) $\mathbf{R}_{uu}(n)$ is the auto-correlation of a white noise input signal, thus $\mathbf{R}_{uu}(m) = \delta(m)$.

Consequently, equation (3.2) can be written as:

$$\mathbf{R}_{uy}(m) = g(m) \quad (3.3)$$

In the parametric technique, a model structure is assumed and the parameters of the model are identified using information extracted from the system [65, 68]. Therefore, the parametric identification of the system is required to define the order of system (number of poles, zeros), in advance [69] and the candidate model is application dependent. For example, a dc-dc buck converter may be represented as a second order IIR filter. Different approaches can be incorporated to estimate the system parameters when using parametric techniques. LMS, RLS, and subspace based methods are some of the dominant approaches [63, 65]. Fundamentally, the main target in parametric identification is to determine the optimal parameters that best describe the unknown model in the system. In accordance with this, the definition of a cost function is also required. Parameterised prediction error methods such as RLS are seeking to minimise the error between the real system $y(n)$ and the estimated model $\hat{y}(n)$ for optimal system identification as shown in Fig. 3.1, and given in (3.4). This error is known as the prediction error $\varepsilon(n)$ [63].

$$\varepsilon(n) = y(n) - \hat{y}(n) \quad (3.4)$$

The main advantage of parametric estimation is that advanced control techniques can easily be integrated with the estimation method. Pole placement and model reference control constitute some of the aforementioned paradigms [28]. Furthermore, a direct control design implementable in a discrete time domain can be applied. This

will substantially reduce errors attributable to transformation approximations from the s -domain to z -domain. In addition, the model can be estimated on-line and in closed loop form, immune to concerns associated with weaknesses inherent to non-parametric identification. Another positive attribute of parametric estimation is its insensitivity to noise. A disadvantage of parametric identification methods is the significant dependence on signal processing, which ultimately inflicts a cost penalty for the target application. The case becomes more complicated if the model contains too many coefficients to estimate, where the solution requires significantly large multiplication matrices.

3.4 Model Structures for Parametric Identification

As mentioned in the previous section, the initial step in parametric system identification methods is to select the appropriate model structure that optimally resembles the dynamic behaviour of the system. As depicted in Fig. 3.2, a linear system model can mathematically be represented by equation (3.5) [70]:

$$y(n) = G(z)u(n) + H(z)e(n) \quad (3.5)$$

It is perfectly appropriate to assume that:

$$G(z) = \frac{B(z)}{A(z)F(z)} \quad (3.6)$$

$$H(z) = \frac{C(z)}{A(z)D(z)}$$

Then by substituting (3.6) into (3.5), the linear model can be described as:

$$y(n) = \frac{B(z)}{A(z)F(z)}u(n) + \frac{C(z)}{A(z)D(z)}e(n) \quad (3.7)$$

where, the models polynomials $A(z)$, $B(z)$, $C(z)$, $D(z)$, and $F(z)$ are as [70]:

$$\begin{aligned}
 A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{nA} z^{-nA} \\
 B(z) &= b_1 z^{-1} + a_2 z^{-2} + \dots + b_{nB} z^{-nB} \\
 C(z) &= 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{nC} z^{-nC} \\
 D(z) &= 1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{nD} z^{-nD} \\
 F(z) &= 1 + f_1 z^{-1} + f_2 z^{-2} + \dots + a_{nF} z^{-nF}
 \end{aligned} \tag{3.8}$$

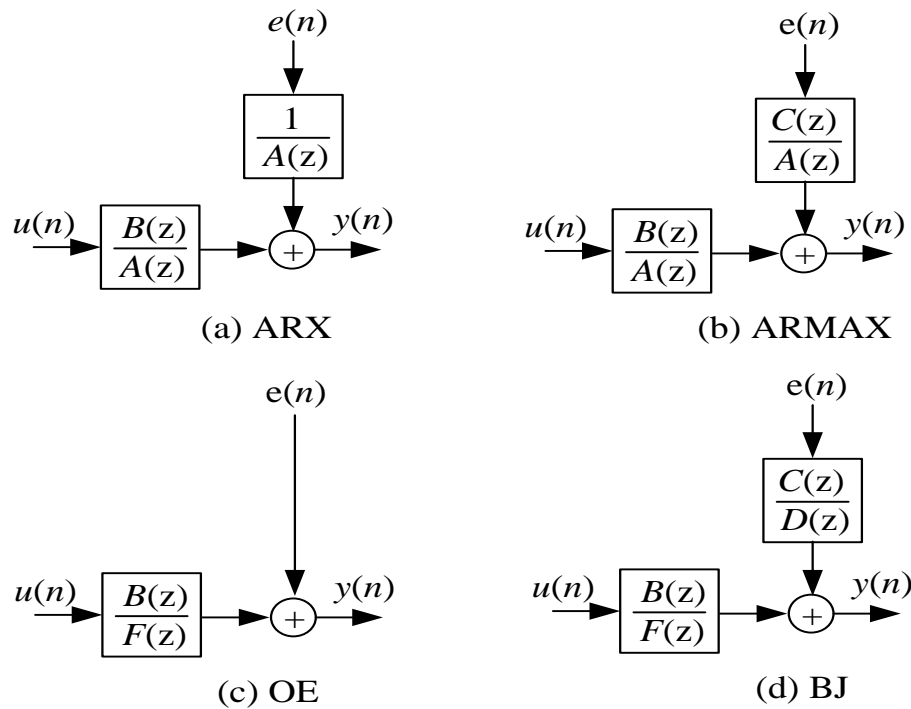


Fig. 3.3 Parametric identification model structures

Depending on the choice of polynomial, there is adequate flexibility to use one of the four popular model structures that are depicted in Fig. 3.3 [63, 70]. The dynamic characteristics of the system and the external disturbance are the most decisive factors in selecting the appropriate model structure. Auto-Regression with Extra input (ARX) is the most popular model, which is often known as the equation error model. The noise term, $e(n)$, is entered directly to the input/output difference equation [63]. Therefore, with minimal effort the minimisation problem can be solved analytically, where the model parameters are estimated directly from the known input and output data vectors. For these reasons, ARX is the preferred choice in many applications

[63]. It is imperative to emphasise that the equation error IIR adaptive filter is incorporated in this research exhibiting similar characteristics with the ARX model. The models of ARX and Auto-Regression Moving Average with Extra input (ARMAX) that include a disturbance term all have a set of common coefficients with the system model, that is $A(z)$ parameters [65]. Thus, the estimation of unknown system parameters using these structures may be biased if the system does not have these common parameters with the noise model. The estimation of the parameters of the noise model using the ARMAX structure provides enhanced flexibility compared with ARX. This is due to the fact that the nominator of the noise model contains the $C(z)$ polynomial. This polynomial can cancel the effects of the denominator polynomial, $A(z)$ [63, 65]. Therefore, to obtain an accurate depiction of the dynamics of the system model independently from the disturbance model, the Output Error (OE) and Box-Jenkins (BJ) structures are immensely more popular. As shown in Fig. 3.3(c, d) the dynamics of the disturbance in BJ and OE models are separated from the system model, rendering a flexibility to handle the disturbance model separately [63, 65]. However, in the OE structure, only the model of the system is described and the noise signal is directly added to the final output, where there is no model that describes the disturbance in this structure [65, 70].

The model structures are further classified into two types: The black box model and the grey box model [63]. In the black box model, there is no prior information about the internal constituents of the system or the physical modelling of the system. Here, the choice of the model structure and the estimation of the parameters of the system are accomplished based on observed data from the system [65, 71]. In the grey box model, the system dynamics and the model structure are partially known in advance. The remaining unknown coefficients are estimated from the measured data. This prior information can be used as a benchmark to analyse the estimation of the model. In addition, this prior information improves the convergence of the applied algorithm. As an illustrative example, some of the power converter parameters in SPMCs such as the capacitance, inductance, or any other measurable physical parameter can be used as known coefficients and can be initially utilised to calibrate the grey box model [31].

3.5 Parametric Identification Process

This section summarise the process of parametric identification of the unknown system. As depicted in Fig. 3.4, the procedure of parametric identification is performed by four main steps [63, 72]. It starts with measuring the experimental input and output data of the unknown system. It is worth noting that an appropriate excitation signal should be injected into the system before collecting the input and output data. This excitation is essential for accurate parameter estimation and to improve the convergence rate of the adaptive algorithm¹.

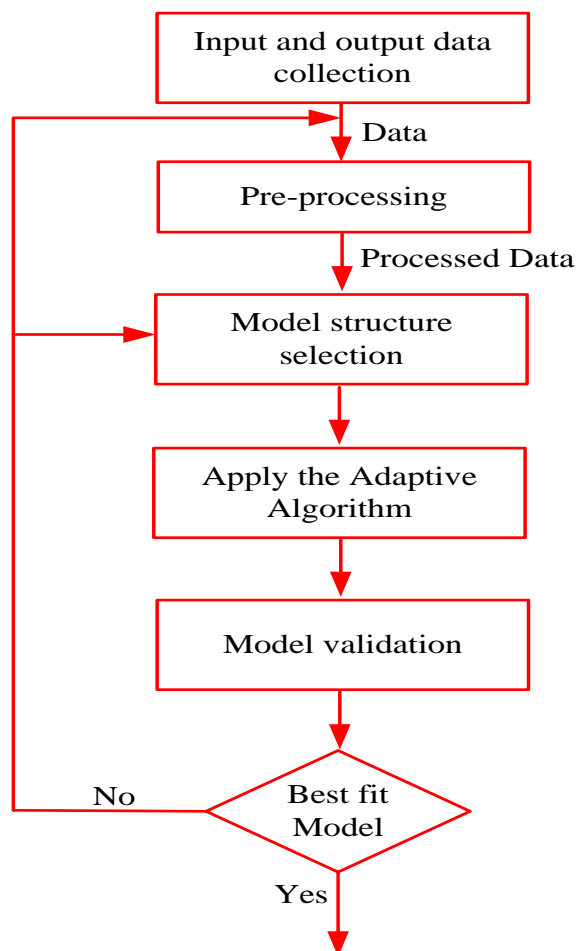


Fig. 3.4 Parametric identification flowchart

¹ More detail on the excitation signal will be presented in the next chapter

Next, the measured data passes to the pre-processing stage. Some examples refer to the pre-processing step, including data filtering to remove the unwanted noise and to determine the mean value from the input and output data for proper estimation. Now, the model structure should be selected and the order of the model is defined. This can be accomplished from the prior knowledge of the system to be estimated. In this case, the selected model considered is a grey box model. The optimisation algorithm is then applied in order to estimate the parameters of the model. The estimated model should provide a best fit with the pre-processed data. This can be achieved by comparing the estimated output data with the measured data. The difference is known as a model error. When the model is acceptable then the estimated parameters are found. Otherwise, the process is repeated by selecting a new model or by pre-processing the input and output data [63, 73].

3.6 Adaptive Control and Adaptive Filter Applications

According to Astrom and Wittenmark, to adapt means “*to adjust a behaviour to conform to new environment*” [74]. Adaptive signal processing and adaptive/self-tuning controllers have a something in common; both scientific disciplines rely on similar mathematical tools and strategies. The design of the adaptive and self-tuning controllers necessitates system identification techniques as a first step, which can be realised by using adaptive signals processing algorithms. Widrow and Plett [75, 76] successfully tuned the parameters of the controller incorporating an adaptive inverse filter scheme. The LMS algorithm has been used to adjust the inverse filter coefficients that pertain to the unknown system. Subsequently, Shafiq in [77, 78] presented a similar paradigm using an inverse adaptive filter. Here, the parameters of the adaptive filter are estimated using the RLS method [78].

One common example of single processing applications is the adaptive filter. Adaptive filters, as well as adaptive controllers, are time varying systems. Their parameters are updated frequently in order to meet the performance requirement. Adaptive controllers offer a robust control solution and can improve the closed loop dynamic response. They are often used in low rate applications, such as process control due to the complexity of the adaptive controllers. This may require a high-specification microprocessor for successful implementation. Advances in

microprocessor efficiency have significantly mitigated the particular drawbacks, making it more feasible to implement adaptive controllers in the applications that operate with a higher sampling rate. Therefore, there is a requirement for further research and development of cost-effective computationally light automatic methods, which continue to offer robust control performance.

3.7 Adaptive Control Structures

There is a plethora of adaptive controller structures that are classified into different categories. The most commonly used controllers are the model-reference adaptive system (MRAS) and the self-tuning controller (STC). In the MRAS paradigm (Fig. 3.5), the control parameters are adjusted based on the error signal between the reference model and the plant. In this way, the parameters converge to their true values. This forces the plant to follow the desired specification as dictated by the model reference. This in turn leads to minimise the error signal to a small value. Here, the error signal is the difference between the reference model and the process model output. To minimise the error signal and ensure system stability, an appropriate adjustment mechanism is required. This is the biggest issue in MRAS [60, 74].

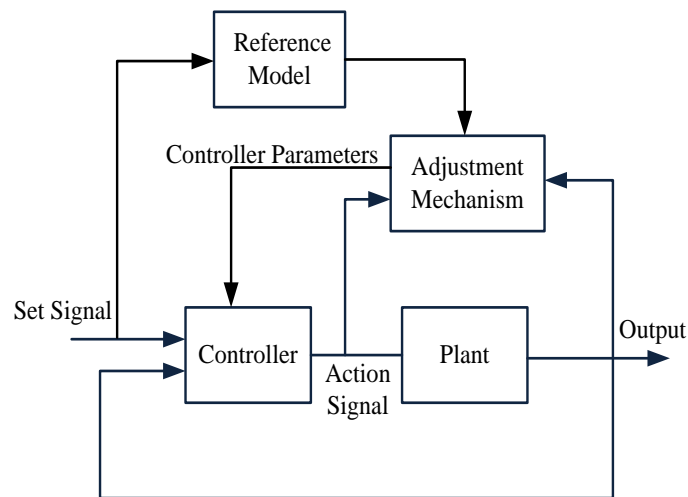


Fig. 3.5 Adaptive model reference structure

In the STC design paradigm, the tuning of the control parameters is accomplished with on-line system identification techniques and the adjustment is performed on-the-fly via the appropriate control design block (Fig. 3.6). Normally the unknown

parameters are estimated based on RLS algorithms. An injection of a perturbation signal in the feedback loop may be essential to improve the convergence of the estimated parameters [74]. The main issue in an STC scheme is the reliability and complexity that characterises the identification part of the process. The auto-tuning controller can also be considered as a special case of STC. In such a system, the adaptation process is only enabled to satisfy tuning demand. Some examples refer to adaptation performed upon the start-up phase, adaptation accomplished by monitoring changes in the system, such as load changes in SMPCs, or adaptation inferred by the user. Clearly, this architecture imposes a reduction to the computational complexity of the adaptation process. Increasingly enhanced artificial intelligence techniques are also used in the design of adaptive controllers. Some of the candidates are fuzzy-logic and neural-networks [60, 79].

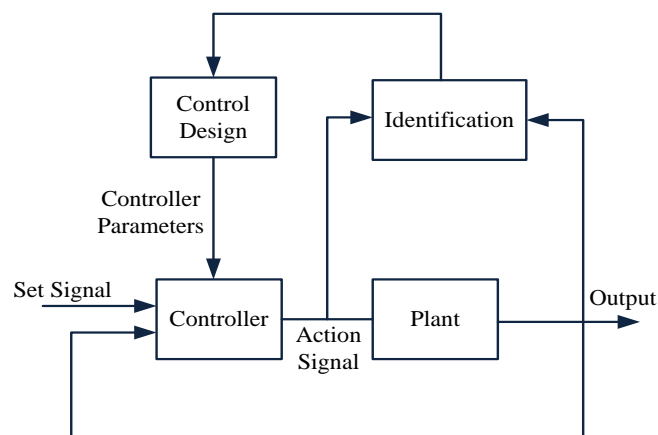


Fig. 3.6 Self-tuning controller block-diagram

3.8 Adaptive Filter Techniques

An adaptive filter may be defined as a “*self-designing*” filter [80], where the filter coefficients are varying continuously until the desired signal is achieved. Often, the desired signal is chosen to be the filter input or the desired estimated output. As shown in Fig. 3.7, the adaptive filter consists of two key components: a digital filter and an adaptation algorithm which is used to vary the tap weight coefficients in real-time. Least square algorithms (LS), such as RLS and LMS, are the most common

adaptive algorithms. The essence of these algorithms is to minimise the estimation error. They accomplish the task by iteratively updating the filter parameters.

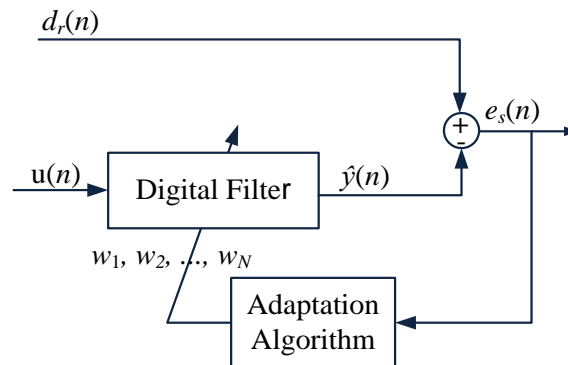
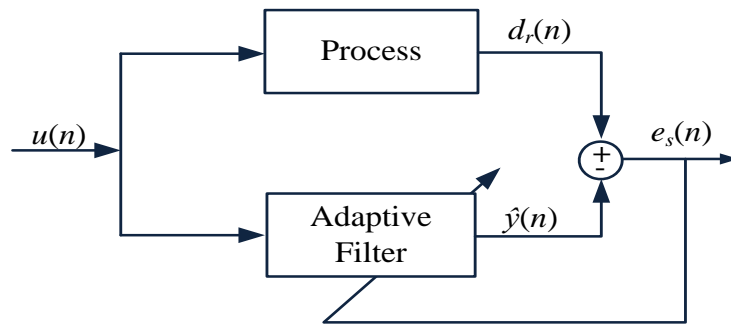


Fig. 3.7 An adaptive filter structure

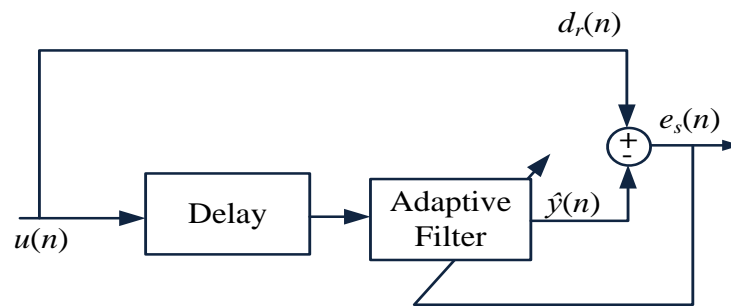
The digital filter can be realised as either: FIR filter (all zeros filter), or IIR filter (poles/zeros filter). The selection of the filter structure depends on the application and the characteristics of the input signal [80, 81]. The FIR filter is simpler to design and robust, as the feedback path does not impose on the general structure of this filter. In contrast, the IIR filter structure, which contains both poles and zeros, entertains a higher level of complexity in the design process. However, the modelling of the unknown system using the IIR filter is computationally more efficient than an FIR filter, since it requires fewer tap-weights in the system model [81].

It is important to emphasise that minimising the estimation error signal $e_s(n)$ is the main objective in adaptive filter structure design. The updated values of the filter coefficients are accomplished by performing error minimisation at each time instance. This minimisation serves two purposes: the finding of optimal filter coefficients, and ensuring the output signal of the adaptive digital filter $\hat{y}(n)$ (estimated signal) is approximately equal to the desired signal $d_r(n)$. An adaptive filter can have different structures depending upon its intended application. Candidates for this may be system identification, signal prediction, noise cancellation, or inverse modelling. The theoretical development for these applications is usually based on a general block diagram of an adaptive filter as illustrated in Fig. 3.8. Four different basic schemes of adaptive filter are depicted each tailored for optimality for individual applications[80, 81]. In the system identification scheme, Fig. 3.8(a), the main design objective is to

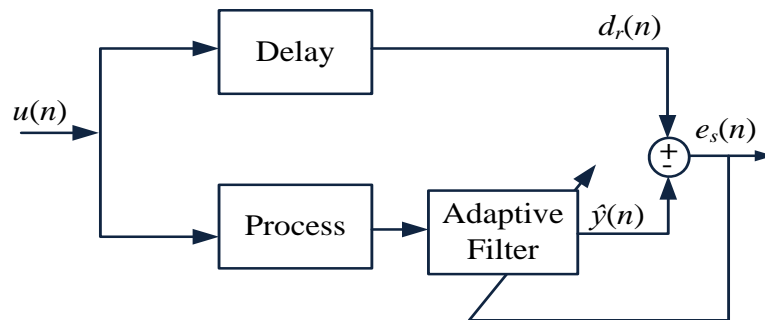
implement a filter that is ideally identical to the unknown process. In this case, the estimation error signal is approximately equal to zero and the adaptive filter algorithm no longer updates the filter coefficients, as long as the system characteristics remain unchanged. In the case of the adaptive prediction error paradigm, Fig. 3.8(b), the previous derived signal is applied as input to the filter and the adaptive filter output is the present estimated or predicted value of the desired signal. The requirement for the error signal to be approximately equal to zero is essential to best design a prediction model. In the scheme depicted in Fig. 3.8(c), the inverse model of the adaptive filter must be matched with the transfer function of the unknown plant. In this way, the error signal between the previous desired signal and the output of the adaptive inverse filter is used in the identification process. In a real time solution, a delay function for the input signal is required to ensure that the system causality is preserved. Finally, an adaptive filter structure can also be used to cancel the effects that the unknown interference in the input signal $v(n)$ may impart. Here, Fig. 3.8(d), an auxiliary signal $v_1(n)$ is supplied to the adaptive filter as a reference input. When the filter coefficients are convergent to their optimal values, the information related to the desired signal is extracted without ambiguity [81]. In this research, adaptive system identification and adaptive filter prediction schemes have been employed to estimate the system parameters as well as to design a real time adaptive controller for SMPC. More details will be presented in Chapter 4 and Chapter 5 relating to these two schemes.



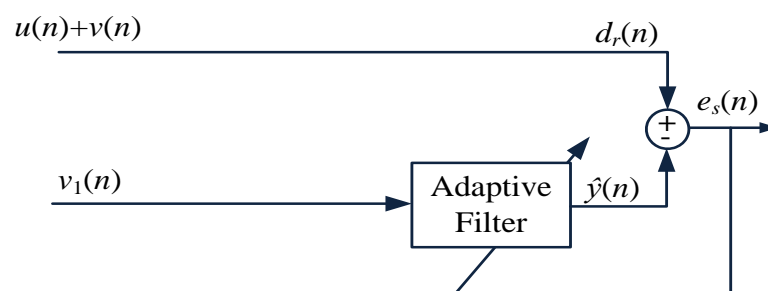
(a)



(b)



(c)



(d)

Fig. 3.8 Adaptive Filter structures, a: system identification, b: signal prediction, c: inverse modelling, d: noise cancellation

3.9 Literature Review on System Identification and Adaptive Control for DC-DC Converters

Recently an enormity of research effort was devoted to system identification and adaptive control techniques for power electronic converter applications. However, these solutions are not always aimed towards low complexity systems. Often, the algorithms require advanced digital signal processing resources which may introduce cost penalties to the target application. This section provides details on recent publications and the motivations in system identification and adaptive/self-tuning controllers for dc-dc power converters.

3.9.1 Non-Parametric System Identification Techniques and Adaptive Control for SMPC

A successful non-parametric method which considers perturbing the duty cycle with a frequency rich input signal (PRBS), is presented in [68, 69, 82]. It starts with estimating the impulse response of the system by performing a cross-correlation between the injected PRBS and output voltage of dc-dc converters. Following that, Fourier Transform method (FFT) is applied to the resulting impulse response data, in order to identify the frequency response of the system. The proposed approach is simple and can handle a wide range of uncertainty in the power converter. However, the identification process may require significant amounts of time to complete and may need to process long data sequences [8]. According to Miao *et al.* [68], the capture of data using 100 kHz as a sampling frequency takes approximately 123 ms to complete. In addition, during the identification process, the system operates in an open loop paradigm without adequate regulation. Furthermore, the ADC quantisation has a significant impact on the identification accuracy. Therefore, Shirazi *et al.* [69] proposed the introduction of a pre-emphasis and de-emphasis filtering techniques to improve the accuracy and to smooth the estimated frequency response. Barkley and Santi [67] developed a technique to improve the accuracy of control-to-output identification by windowing the measured cross-correlation between the input and output of the dc-dc converter. Roinila *et al.* [83, 84] proposed the injecting of the other types of PRBS known as inverse repeat binary sequence (IRBS) to improve the identification sensitivity to disturbances in the system.

Subsequent to [69], Yan Liu *et al.* [85] presents a similar technique to tune the controller coefficients based on the identified control-to-output model of the dc-dc converter using a correlation approach.

An alternative system identification methodology based on frequency domain techniques is employed in [6]. The authors here proposed to inject a sinusoidal signal in order to directly estimate the frequency response of the control-to-output transfer function using FFT.

Whilst these methods are straightforward to implement, designing a controller using non-parametric system identification methods is usually limited to frequency response methods only. In addition, a complete real-time solution of system identification and adaptive control for SMPCs based on frequency measurement is rarely presented in the literature. The authors in [67, 68, 82], used an FPGA board to implement the control loop, the PRBS generation and to collect the experimental data. This data is subsequently post-processed in MATLAB for off-line testing of the proposed algorithms. In [83, 84] an advanced, high cost, data acquisition card (NI PCI-6115) is used. Again, off-line evaluation based on the system identification algorithm is carried out in MATLAB/Simulink. Kong *et al.* [6] used a Texas Instrument UCD9240 device based DSP for system verification. The literature confirms that there is only one complete embedded auto-tuning controller that relies upon the on-line frequency response identification, presented by the authors in [2, 86]. The implementation in this study was achieved through the Virtex-4 FPGA.

Recently, Costabeber *et al.* [87], incorporated the cross-correlation approach presented earlier with a model reference adaptive controller for a digitally controlled SMPC. The difference between the estimated impulse response and the model reference impulse response has been utilised to tune the gains of the PID controller. For simplicity the integral gain was assumed to be fixed. As a result, the auto-tuning process is only performed on the proportional-derivative gains. An optimised search method is used to tune the PD coefficients; this results in minimising the estimated error. According to [87] the control parameters take a long time to converge to the final value. Consequently, a deterministic approach that does not depend on the impulse response estimation is also investigated by the authors in [87]. Here, the

difference between the loop impulse response and the model reference impulse response is considered in the tuning algorithm. The convergence rate using this approach is superior to the cross-correlation scheme. The proof of concept was experimentally verified using a low cost TMS320F2808-DSP.

In summary, in many of the methods presented, it was found that these approaches restrict the ability of continuous parameters estimation that is required in continuous parameters tuning [3] for adaptive controller applications. These self-tuning and adaptive control techniques are most effective during the steady-state and the parameters are tuned using pre-determined rules, such as phase margin and gain margin requirements. Therefore, these categories of controller are generally unsuitable for time varying systems where on-line compensation is desirable. One solution for on-line parameter estimation is introduced by using RLS algorithm. For this reason, RLS is used in many system identification and adaptive control strategies.

3.9.2 Parametric Estimation Techniques and Adaptive Control for SMPC

Straightforward relay-feedback based methods have been successfully used in the parameter identification and auto-tuning of dc-dc converters [7, 88, 89]. The identification and tuning processes are performed during the period of system start-up. The method starts to introduce oscillations at a specific frequency into the regulated output for a short period. Then, the system parameters are estimated based on the measured frequency of the oscillated signal. Following this, the parameters of the PID controller are auto-tuned iteratively, until the predefined feedback-loop specifications are met. However, this type of approach requires relatively complex algorithmic steps to tune the controller parameters. Typically, it requires three iterative tuning phases to adapt the PID parameters. In addition, a relatively large oscillated signal at the output voltage of dc-dc converter is introduced during the auto-tuning phases [52]. The auto-tuning process is completed after 27 ms at 200 kHz sampling frequency [86]. The algorithm is implemented on a Virtex IV-FPGA using the MATLAB System Generator toolbox [86].

Similar technique in [54, 90] has been proposed, such as inserting LCO into the system during steady-state period. Here, the LCO is generated by reducing the

resolution of DPWM instead of using a relay in the feedback loop. Also, the feedback loop is temporarily compensated by integral control only. In consequence, the effect of LCO is amplified, thus it can be easily observed. The amplitude and frequency information are then extracted from the LCO signal to find the dc-dc converter parameters (corner frequency and quality factor) [52]. In the second phase, the PID compensator is re-tuned using the pole-zero cancellation approach. Whilst hardware efficient, this method results in a lower system identification accuracy [48]. Another negative aspect is that the identifier and the auto-tuner does not consider the influence of the R_C resistance in the design [54]. The authors here implemented the DPWM by an Altera-FPGA and the proposed algorithm has been validated by Analog Device ADMC-401-DSP.

As previously indicated, for simplicity of the identification and adaptive control design, recursive techniques are also developed for dc-dc converters. Recursive identification methods are a very familiar approach in on-line applications. However, these methods are not fully exploited in low cost, low power SMPCs due to the computational complexity of the identification algorithm, which may necessitate a high specification microprocessor for effective implementation.

Peretz and Bin-Yaakov [1, 30, 91] demonstrated an open loop system identification approach, to determine the control-to-output voltage model of a dc-dc converters. The authors proposed to perturb the system by means of a step change in the duty cycle signal. The same injection sequence has been repeated for a number of times, five sequences in total. The DSP is then utilised to collect the averaged input and output sampled data. The recorded data is used for estimation of the system parameters. It uses the iterative least square method incorporating Steiglitz and McBride IIR filter. According to the authors [1], a 5 % step change in the duty cycle causes a change of 1 V at the output of the dc-dc converter. The time elapsed for the identification procedure to complete is about 120 ms. Therefore, the presented approach is not applicable for the design of on-line adaptive controller and for tracking the variation in parameters within the system. The identification scheme was implemented on a TMS320F2808-DSP involving MATLAB Real-Time Workshop toolbox. The resultant open loop discrete dc-dc model was incorporated for the direct

digital control design method by Ragazzini's [27]. The proposed controller has been implemented experimentally by DSP platform. However, the design steps necessitated an off-line optimisation or curve fitting method, to convert the resultant high order Ragazzini controller to match the desired second order digital PID controller. The authors here concluded that the digital control model relying upon discrete estimation provides better performance than the mathematically calculated model.

A black box Non-linear modelling based on least square algorithm of dc-dc converter is proposed by Alonge *et al.* [92, 93]. The technique presented here is based on the Hammerstein model; this model consists of a non-linear static model in conjunction with a LTI ARX model. The ARX model captures the dynamic characteristics of the system. Two steps are required to define the system model. In the first step, and during the steady-state period, the converter is supplied by a constant input voltage with a variable duty cycle signal and the corresponding output voltage is measured; the non-linear static model will then be identified. In the second phase, a PRBS is injected to excite the system dynamics, and the measured values of the control-to-output voltage data are observed to estimate the second order ARX model candidate. This technique accurately describes the dc-dc converter model; therefore, a robust controller is derived. However, the approach is quite complex and time-consuming for real-time operation [92]. The experimental data is captured using a DSP platform (dSpace DS1103).

Another approach of parametric black box modelling of the dc-dc converter is presented by Valdivia *et al.* [94]. Here, the dynamic response of the dc-dc converter is excited by a step load change and the output response is captured. When the resultant dynamic is analysed as a LTI responses, the model can be identified using the LTI identification approach (LS algorithm); otherwise the non-linear method should be used (Hammerstein scheme). The OE model is employed in this technique to identify the LTI parameters of the dc-dc converter using the MATLAB System Identification toolbox [95]. The proposed method is suitable for a simulation estimation of the dc-dc converter, where the estimation procedure requires many steps and advance analysis prior to estimation.

Kelly and Rinne [96, 97] proposed an adaptive, self-learning, digital regulator, based on a one-tap LMS prediction error filter (PEF) for on-line system identification. The presented solution is simpler than many other methods and a prior knowledge of system parameters is not required in the adaptation process. However, there appears to be two limitations to this system. Firstly, the scheme involves subjecting the system to a repetitive disturbance to excite the FIR filter and improve the convergence of filter tap-weights [98], which after many iterations the controller begins to learn. Furthermore, in this scheme only a PD controller is considered and this can yield a non-zero steady-state error [54], thus a feed-forward loop should be introduced to ensure system stability and achieve regulation. Initially, this adaptive controller was implemented using a DSP from Analog device. This subsequently lead to the design of a microprocessor architecture adopting dual multiply-accumulator (MAC) [99]. The feed-forward gain for the digitally controlled buck converter as described in [57], has been adaptively determined based upon the same concept as using a first order PEF.

A real time parametric system identification method using a classical RLS technique is presented by Pitel and Krein [31]. It identifies the parameters of an open loop buck converter during abrupt load changes from the control signal to the inductor current transfer function. This work accurately estimates the parameters during the initial start-up of the system, and during periods of relatively slow load changes. It concludes that a major challenge is to estimate the load value after abrupt changes. An effective implementation of the RLS algorithm based on fixed-point DSP (TMS320F2812) using the MATLAB Embedded Target Support Package toolbox has been demonstrated in this research. However, the estimation process using the RLS algorithm operates only with a very low sampling rate of approximately 4 kHz.

B. Miao *et al.* [5] presented a dual identification scheme. In this approach both a parametric and a non-parametric method are combined to estimate the parameters of an SMPC and then to directly design a digital controller. The identification occurs in two phases. Initially the open-loop frequency response of the system is identified based on FFT techniques, then the converter parameters are estimated using a

parametric recursive method, based on the obtained frequency response data. Implementing two different methods is clearly more complex and computationally heavy for on-line system identification purposes. Therefore, it is more suitable to address off-line scenarios.

A similar approach has been proposed in [100, 101], for auto-tuning the controller of an SMPC. During the period that the system has reached the steady-state a perturbed signal is injected into the control loop and the system frequency response is estimated. In this approach, it was proposed to incorporate a model fitting technique with a recursive parameterisation algorithm. The objective is to determine the candidate model which resembles the estimated frequency response data. Subsequently, the controller parameters are re-tuned based on the estimated model. This approach is not immune to high computations burdens which restrict its applicability for on-line estimation of SMPCs.

Tae-Jin *et al.* [102] proposes an aging diagnosis approach for the dc-dc converter using a least square identification algorithm. A white noise signal is injected into the feedback loop and the input and output data (control/output signals) are stored into the DSP memory. The parameters of the dc-dc converter are then estimated using MATLAB System Identification toolbox based upon the output-error model (OE) structure [103]. The diagnostic decision relies upon estimating the parasitic resistance (R_L/R_C) of the dc-dc converter. These values are then compared by using a manufactured of the dc-dc converter sample and cross-referencing the manufacture disclosed characteristics with those obtained to confirm validity. The proposed approach can be used as an off-line indicator of converter aging.

An application of an adaptive controller for a dc-dc converter based on the conventional RLS scheme has been proposed by Beid *et al.* [104]. A pole placement approach is utilised in this scheme for the on-line tuning of control parameters. The performance of the proposed adaptive controller has been verified by simulation only. Therefore, system complexity is not investigated for this highly hardware demanding combination of RLS and pole-placement controller for the target application.

3.9.3 Independent Adaptive Control Technique for SMPC

Several techniques that involve the design of adaptive controllers immune to the need for system identification process are incorporated in the case of dc-dc converters. The most popular paradigm in the literature is that of the non-linear control. Non-linear adaptive controllers are widely used in the control design of dc-dc SMPCs where their placement in the control loop results in improvement of the transient response of the dc-dc converter. The non-linear compensators can be employed as a standalone controller in the feedback loop or as an augmented controller. It is worth mentioning that the non-linear PID controller is the most frequently structure that is applied to the SMPC. This is due to balance of the simplicity of design and effectiveness. The authors in [42, 51, 105] have developed this type of controller for the case dc-dc SMPCs. In these schemes, the gains of the PID controller are adaptively tuned based on non-linear methodology. However, other non-linear structures have also been proposed in the publications such as fuzzy logic (FL) control.

Fuzzy logic (FL) adaptive schemes are effectively implemented for digitally control of SMPCs. Farahani *et al.* [106] utilised a look-up table technique to implement a fuzzy logic controller on an 8-bit microcontroller chip (PIC18F452). The performance of the controller was compared with the conventional PI controller, shown that the FL controller provides better dynamic performance over the PI control. However, the author has validated the system performance during initial start-up only, where no abrupt parameter changes are applied to the SMPC to verify the robustness of the proposed controller subjected to fast changes. A real time adaptive controller based on a FL system has also been presented by Ofoli and Rubaai in [107]. Here, the FL system is implemented using a PC and the inputs signals are sampled via a data acquisition card (DAP 840) using a 14-bits ADCs. MATLAB and LABVIEW are incorporated to acquire the sampled data and then to implement the FL on using the PC. The output from the fuzzy controller is then exported to the microcontroller for PWM generation. The results from the fuzzy control are preferable in comparison to the conventional digital PID compensator. However, the experimental setup requires the availability of significant hardware resources, in

excess of what would be anticipated in a typical dc-dc converter application. It is worth mentioning that L. Guo *et al.* [24] presented a thorough comparison between fuzzy controller and classical digital PID controllers in terms of demands on experimental implementation for the two schemes. The evaluation was applied to both buck and boost dc-dc converters using a TMS320F218-DSP. Again, it was demonstrated that FL control was more robust and provided faster transient response compare to conventional PID controller.

Alternative adaptive schemes that do not rely upon a system identification approach have been presented in the literature. One such paradigm is known as the dual mode adaptive approach. In this approach a linear controller such as the PID controller operates at the steady-state mode and an advanced control algorithms, is used in transient mode; for example, non-linear controllers. This scheme was employed in [47, 108, 109]. Two loops, linear and non-linear with a transient monitoring circuit, are utilised to obtain an efficient transient response of the SMPC.

Another techniques, using a charge balance controller, is presented by [110, 111]. This methodology requires monitoring the peak and the valley points of the output voltage and inductor current to achieve optimal dynamic response during load changes. The main challenge in these schemes is formulating the transient curve and the method of detecting/measuring the required points on this curve. This process involves complex mathematical analysis and precise knowledge of the power converter parameters [52].

Finally, a model reference auto-tuning scheme was also proposed for digital control of dc-dc converters [48, 98]. The authors consider injecting the control loop with a perturbation signal at a desired cross-over frequency and then tune the model reference controller until the pre-defined targets (loop bandwidth and phase margin) are achieved. Here, only the PD parameters are tuned and a fixed integral gain is placed in parallel with the adaptive PD controller into the feedback loop. The proposed solution has been experimentally tested using the TMS320F2808-DSP platform.

3.10 Chapter Summary

This chapter has presented an overview of the principles and techniques used in system identification. It has provided details of the methods that are used in system identification, with the focusing more on parametric estimation techniques. Model structures used in parametric estimation techniques have been demonstrated. In the chapter adequate information on adaptive controllers and adaptive filter was provided. Adaptive control structures were outlined, with emphasis on model reference and self-tuning adaptive schemes. Adaptive filter applications were also demonstrated. Recent research on system identification and adaptive control techniques for dc-dc SMPCs were reviewed. The main focal point is on adaptive controllers based upon parametric/non-parametric system identification processes. Adaptive control strategies that do not necessitate the incorporation of system identification for the case of dc-dc converters were appropriately examined.

Chapter 4

SYSTEM IDENTIFICATION OF DC-DC CONVERTER USING A RECURSIVE DCD-IIR ADAPTIVE FILTER

4.1 Introduction

For a high performance controller with high dynamic performance, accurate estimation of the system parameters is essential [5]. Normally, in digitally controlled systems, a discrete time transfer function model of the plant is used for the control design [5, 6]. The actual form of the transfer function, and the numerical values of its coefficients, are dependent upon the individual parameters of the plant to be controlled [54]. It is the fundamental role of the system identification process to evaluate each coefficient of the transfer function. In many applications, it is very important that the coefficients are calculated as accurately as possible, since this will ultimately determine the closed loop controller response. However, in SMPC applications, it is also necessary to acquire the system parameters rapidly. The time constants in PWM switched power converters are often very short, and it is not uncommon for abrupt load changes to be observed. Any system identification scheme must be able to respond appropriately to these characteristics. However, to achieve improved accuracy and/or speed also implies the need for a faster, more powerful microprocessor platform. This is not always viable in SMPC applications, where it is essential to keep system costs low and competitive. Therefore, there is a need for computationally light system identification schemes which enable these advanced techniques to be performed on lower cost hardware.

Unfortunately, in many of the methods discussed in the literature review, significant signal processing is required to implement these schemes and this

eventually has a cost penalty for the target application. Furthermore, the computational complexity impacts upon time of execution in the microprocessor, and this in turn makes it difficult to adopt in continuous parameter estimation for adaptive control applications [4]. In addition, identification/adaptation process required many steps to achieve.

For this reason, this chapter introduces a novel technique for on-line system identification. Specific attention is given to the parameter estimation of dc-dc SMPC. However, the proposed method can be implemented for many alternative applications where efficient and accurate parameter estimation is required. The proposed technique is computationally efficient, based around a DCD algorithm, and uses an IIR adaptive filter as the plant model. The system identification technique reduces the computational complexity of classical RLS algorithms. Importantly, the proposed method is also able to identify the parameters quickly and accurately; thus offering an efficient hardware solution which is well suited to real time applications. This algorithm has previously been developed for use in the field of telecommunications [112, 113]. Here, we adapt the algorithm and apply it for the first time in the system identification of power electronic circuits. Results clearly demonstrate that the proposed scheme estimates the dc-dc converter parameters quickly and accurately. Importantly, the approach can be directly embedded into adaptive and self-tuning digital controllers to improve the control performance of a wide range of industrial and commercial applications.

4.2 System Identification of DC-DC Converter Using Adaptive IIR DCD-RLS

Algorithm

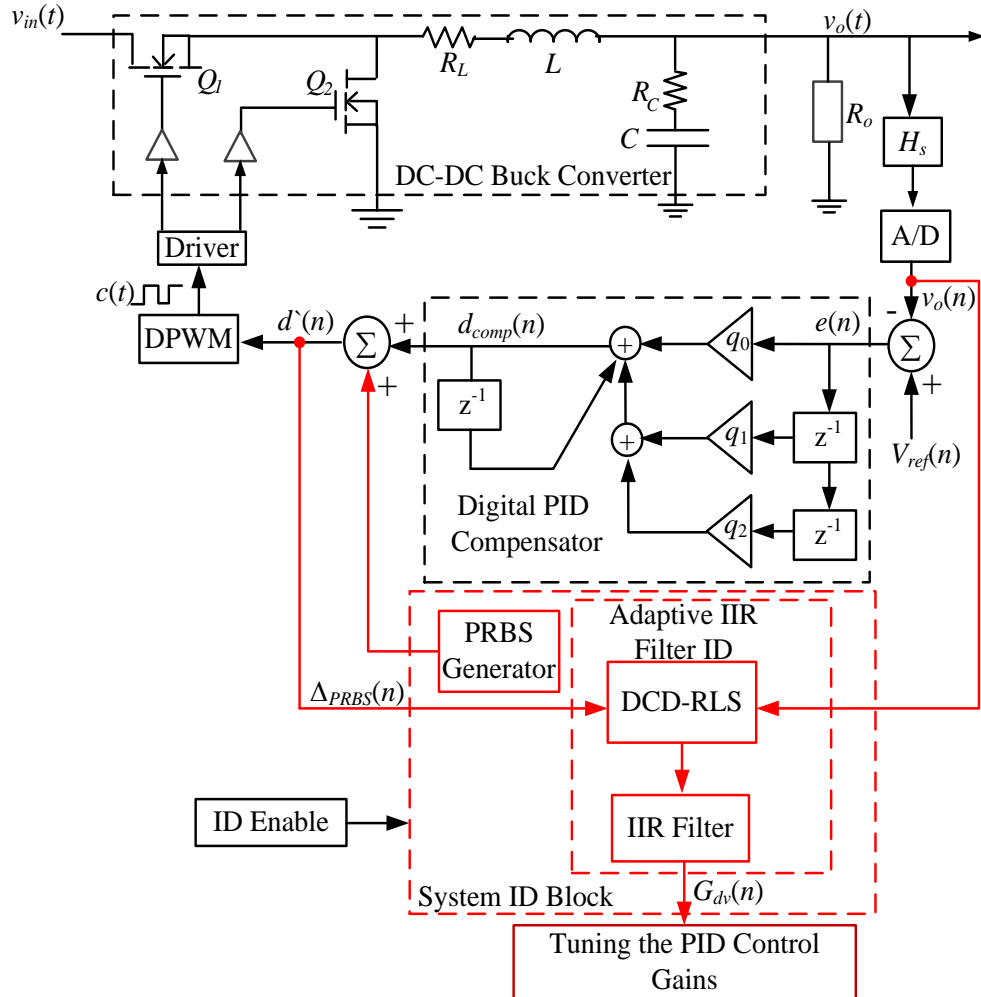


Fig. 4.1 The proposed closed loop adaptive IIR identification method using DCD-RLS algorithm

Fig. 4.1 illustrates a block diagram of the proposed identification scheme. Here, a closed loop synchronous dc-dc buck converter is controlled via a digital PID compensator. In addition, a real-time system identification algorithm is inserted alongside the controller, continually updating the parameters of a discrete model of the buck converter system on a sample by sample basis. The identification system can be enabled and disabled on demand during operation. For example, it may be applied at start-up, at regular set intervals, or enabled on detection of a system change such as

a variation in the system load. Monitoring the voltage loop error is one simple way to detect a system change and enable the system identification process. When enabled, a small excitation signal is injected into the control loop. This is required to improve the convergence time of the adaptive filter; this is the time to obtain optimal filter tap weights for accurate parameter estimation. For all on-line identification methods, some form of system perturbation is essential for the estimation process. In this scheme, the Pseudo-Random-Binary-Sequence (PRBS) is selected. As shown in Fig. 4.1, the PRBS signal is added to the PID controller output signal, $d_{comp}(n)$. This creates a control signal, $d'(n)$, with a superimposed persistent excitation component. Once applied to the DPWM, a small disturbance in the output duty cycle, $c(t)$ is generated. In this way, the duty cycle command signal at steady-state will vary between $d_{comp}(n) \pm \Delta_{PRBS}(n)$. Here, the average steady-state duty cycle is 0.33 and the magnitude of PRBS signal, $\Delta_{PRBS} = \pm 0.025$, therefore a change of approximately equal to $33 \% \pm 2.5 \%$ in duty cycle signal will be observed. This will then cause an excitation signal in the buck converter output voltage, $v_o(t)$. During this process, the excited output control signal and the sampled output voltage are ($d'(n)$ and $v_o(n)$ in Fig. 4.1). Once the samples have been pre-processed to eliminate any unwanted high frequency noise, they are passed to the identification algorithm (DCD-RLS block in Fig. 4.1) to estimate the system parameters and update the discrete IIR filter model of the SMPC. The following sections describe each block in Fig. 4.1 more details, including a complete description of the algorithms proposed to implement the system identification.

4.3 Adaptive System Identification

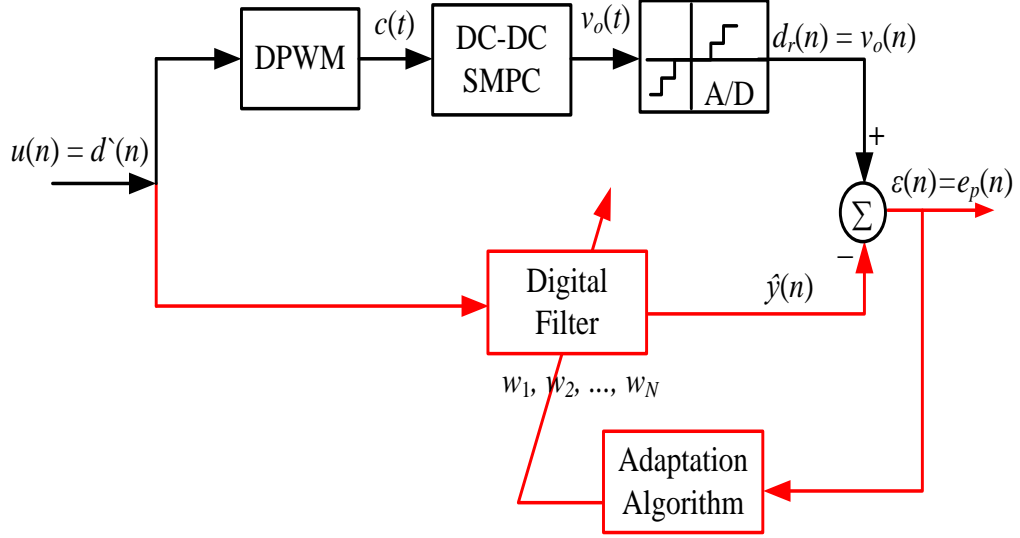


Fig. 4.2 Adaptive system identification block diagram

As initially presented in Chapter 3, an adaptive filter can have different structures depending upon its application. In Fig. 4.1, an adaptive IIR filter is employed for system identification. The major concern is minimising the prediction error signal, $e_p(n)$. Ideally, we want this signal to equal zero, indicating excellent parameter estimation. However, practical issues such as measurement errors, unwanted noise, quantisation, and delay time make this difficult to achieve. By minimising the prediction error signal, the optimal parameters estimation is found. As shown in Fig. 4.2, the desired signal is the sampled output voltage of the dc-dc converter. Based on this, we can write [80]:

$$\hat{y}(n) = \sum_{k=0}^N w_k u(n-k) = \mathbf{w}^T \mathbf{u} \quad (4.1)$$

$$\mathbf{w}(n) = [w_0 \quad w_1 \quad \cdots \quad w_N]^T \quad (4.2)$$

$$\mathbf{u}(n) = [u(n) \quad u(n-1) \quad \cdots \quad u(n-N)]^T$$

where, the pre-filtered input signal $u(n)$, is continuously adapted in response to the filter weight update. The model of the unknown plant system (in this case, the dc-dc converter system) is defined by the transfer function of the adaptive filter. Therefore,

as long as the parameters of the plant do not change, the digital filter coefficients, \mathbf{w} , will remain the same [80]. However, defining the digital filter coefficients requires analytical calculation of the linear system equations. This can be achieved using Wiener equations, but requires considerable computational effort [96]. Alternative methods, such as adaptive approaches can also be used to optimally calculate the tap weights and can help to reduce the mathematical burden and trim the computational load [80, 96]. Here, we employ an adaptive DCD-RLS algorithm to continuously adjust the filter coefficients and minimise $e_p(n)$. The error prediction is defined as [80]:

$$e_p(n) = d_r(n) - \hat{y}(n) = d_r(n) - \sum_{k=0}^N w_k u(n-k) = d_r(n) - \mathbf{w}^T \mathbf{u}(n) \quad (4.3)$$

According to (4.3), the error prediction signal is determined by applying the input signal to the digital filter to produce an estimation output signal, $\hat{y}(n)$. The prediction error is then the difference between the desired signal, $d_r(n)$, and this generated estimation output signal. When the prediction error is minimised, the adaptive filter tap-weights reach steady-state and no longer require updating. However, if any parameters of the plant change, the prediction error will deviate from the minimum point and the adaptive algorithm will start to determine the new filter tap-weights in response to this change. To minimise the error signal, the adaptive algorithm must solve a series of linear equations to estimate the vector coefficients, \mathbf{w} . Generally, this is can be accomplished using the well known least square (LS) algorithms [80].

4.4 Least Square Parameters Estimation

LS estimation techniques are fundamental in adaptive signal processing applications. In real-time applications, the solution is typically based on matrix inversion which, due to the computational complexity, is particularly difficult to implement [114]. The LS algorithm evaluates and calculates the finite vector of estimated parameters, to obtain a small estimation error. This is achieved by minimising the prediction error signal based on the criterion of the sum of the prediction error squares [80]:

$$J(n) = \sum_{k=1}^n e_p^2(k) = \sum_{k=1}^n [d_r(k) - \mathbf{w}^T \mathbf{u}(k)]^2 \quad (4.4)$$

By differentiating equation (4.4) with respect to \mathbf{w} and setting this equal to zero; the estimated parameters that obtained the minimisation criterion of sum squares of prediction error can be found [28, 72]:

$$\frac{\partial J}{\partial \mathbf{w}} = -2 \sum_{k=1}^n [d_r(k) - \mathbf{w}^T \mathbf{u}(k)] \mathbf{u}(k) \quad (4.5)$$

$$\frac{\partial J}{\partial \mathbf{w}} = 0$$

If we assume that $\mathbf{w} = \hat{\mathbf{w}}_{\text{LS}}$, and by solving equation (4.5) for $\hat{\mathbf{w}}_{\text{LS}}$; the estimated parameters values are calculated [63]:

$$\hat{\mathbf{w}}_{\text{LS}} = \left[\sum_{k=1}^n \mathbf{u}(k) \mathbf{u}^T(k) \right]^{-1} \sum_{k=1}^n d_r(k) \mathbf{u}(k) \quad (4.6)$$

From (4.6), the estimated least square parameters vectors can be written as:

$$\hat{\mathbf{w}}_{\text{LS}} = \mathbf{R}^{-1} \boldsymbol{\beta} \quad (4.7)$$

where:

$$\mathbf{R}(n) = \sum_{k=1}^n \mathbf{u}(k) \mathbf{u}^T(k) \quad (4.8)$$

$$\boldsymbol{\beta}(n) = \sum_{k=1}^n d_r(k) \mathbf{u}(k)$$

\mathbf{R} is an auto-correlation matrix of size $N \times N$, and $\boldsymbol{\beta}$ is the cross-correlation vector of length N . These series of equations can be used to find the estimated parameters values of $\hat{\mathbf{w}}_{\text{LS}}$. They are called normal equations [28, 63].

4.5 Conventional RLS Estimation

Many adaptive control systems are based upon real time parameter estimation [60, 74]. Among them, RLS based algorithms provide a simple adaptive scheme which is capable of a fast convergence rate, good estimation accuracy, and fast tracking ability to system parameter changes. However, only limited literature describes the application of these methods in low complexity systems, such as dc-dc converters. This is because the solution is normally based on matrix inversion operation, which is computationally heavy and presents implementation difficulties. The best way to reduce computational complexity is to avoid or find an approximation method to the matrix inversion operation [113]. Typically, a matrix inversion lemma algorithm is required to eliminate such operation [80].

The RLS process can be performed by arranging the computations in such way that the results obtained at time instance $(n - 1)$ can be used in order to find the estimates at time instance (n) [74]. Therefore, the auto-correlation matrix and cross-correlation vector are sequentially computing as given in equation (4.9). The filter coefficients are updated recursively with complexity of $O(N^2)$ for matrix vector multiplication and around $O(N^3)$ for auto-correlation matrix inversion (4.10) [112, 113]. As a result, the final solution of normal equations in (4.10) is directly proportional to $[O(N^2) + O(N^3)]$.

$$\mathbf{R}(n) = \mathbf{R}(n-1) + \mathbf{u}^T(n)\mathbf{u}(n) \quad (4.9)$$

$$\boldsymbol{\beta}(n) = \boldsymbol{\beta}(n-1) + d_r(n)\mathbf{u}(n)$$

$$\mathbf{w} = \mathbf{R}^{-1}\boldsymbol{\beta} \quad (4.10)$$

Many adaptive filter methodologies are based on matrix inversion operation which results in numerical inaccuracies due to finite precision implementation. Another technique can be used to solve the inverse operation in (4.10), often results in more accurate adaptive algorithm [112]. However, the conventional RLS algorithm based matrix inversion lemma is summarised in Table 4.1 (Appendix A shows the derivation details of the RLS algorithm using matrix inversion lemma) and the closed loop signal operation is depicted in Fig. 4.3 [80]. In Table 4.1, $\mathbf{u}(n)$ is the data vector,

$\hat{\mathbf{w}}(n)$ is the estimated tap-weights, $e(n)$ priori estimation error, $\mathbf{P}(n)$ is a $N \times N$ inverse correlation matrix, $\mathbf{k}(n)$ is a $N \times 1$ adaptation gain vector, and for ordinary RLS the forgetting factor ($\lambda = 1$).

Table 4.1 Conventional RLS algorithm based matrix inversion lemma

Step	Equation
	Initialization: $\hat{\mathbf{w}} = 0$, $\mathbf{P}(0) = \frac{1}{\delta} \mathbf{I}_N$
	for $n = 1, 2, \dots$
1	$\mathbf{S}(n) = \mathbf{P}(n-1)\mathbf{u}(n)$
2	$\mathbf{k}(n) = \frac{\mathbf{S}(n)}{\lambda + \mathbf{u}^T(n)\mathbf{S}(n)}$
3	$e(n) = d_r(n) - \hat{\mathbf{w}}^T(n-1)\mathbf{u}(n)$
4	$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)e(n)$
5	$\mathbf{P}(n) = \frac{1}{\lambda} \left[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{u}^T(n)\mathbf{P}(n-1) \right]$

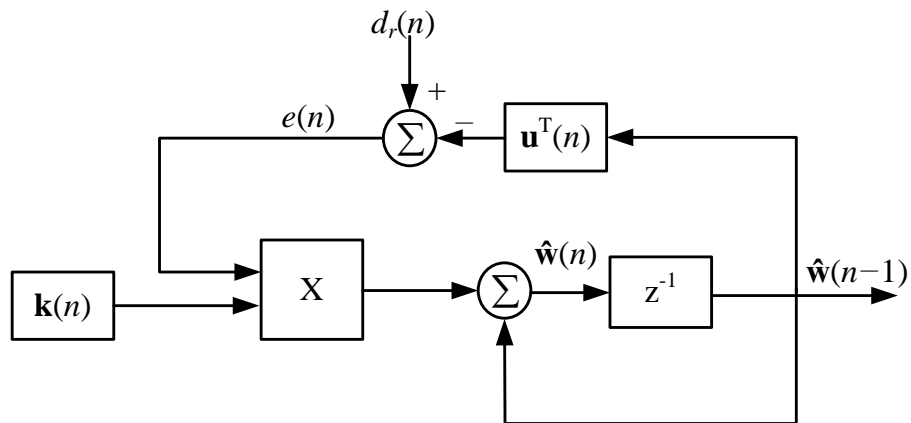


Fig. 4.3 Closed loop operation of conventional RLS algorithm based matrix inversion lemma

4.6 Normal Equations Solution Based On Iterative RLS Approach

As described in Table 4.1, the solution of normal equations at every time instance is computationally heavy and presents implementation difficulties. However, there are alternative algorithms for solving the linear equations expressed in (4.10). Amongst them, the DCD algorithm appears to be a particularly effective method [112, 113, 115]. Attractively, the computation is based on an efficient, iterative approach with no explicit division operations. This makes it very appropriate for real time hardware implementation. As mentioned earlier, direct methods require a complex matrix inversion operation to solve the linear equations in (4.10). However, in this method (first proposed by Zakharov *et al.* [112], in the field of communications) an alternative solution is presented by converting (4.10) into a sequence of auxiliary normal equations that can be solved using iterative techniques. Firstly, at time instance $(n - 1)$, the solution to the system equation $\mathbf{R}(n - 1)\mathbf{w}(n - 1) = \boldsymbol{\beta}(n - 1)$ can be approximated; the approximate solution is $\hat{\mathbf{w}}(n-1)$. The residual vector of this solution can be written as [112]:

$$\mathbf{r}(n - 1) = \boldsymbol{\beta}(n - 1) - \mathbf{R}(n - 1)\hat{\mathbf{w}}(n - 1) \quad (4.11)$$

The system in (4.10) is then solved at each time instance, n . From which:

$$\begin{aligned} \Delta\mathbf{R}(n) &= \mathbf{R}(n) - \mathbf{R}(n - 1) \\ \Delta\boldsymbol{\beta}(n) &= \boldsymbol{\beta}(n) - \boldsymbol{\beta}(n - 1) \\ \Delta\mathbf{w}(n) &= \mathbf{w}(n) - \hat{\mathbf{w}}(n - 1) \end{aligned} \quad (4.12)$$

The objectives is to find a solution $\hat{\mathbf{w}}(n)$ of linear equation in (4.10) by using the previous solution $\hat{\mathbf{w}}(n - 1)$ and the residual vector $\mathbf{r}(n - 1)$. From this, a solution for $\hat{\mathbf{w}}(n)$ in (4.10) can be described as:

$$\mathbf{R}(n)[\hat{\mathbf{w}}(n - 1) + \Delta\mathbf{w}(n)] = \boldsymbol{\beta}(n) \quad (4.13)$$

Using (4.11)-(4.13), and solve with respect to the unknown vector $\Delta\mathbf{w}$, the normal equations in (4.10) can then be represented as a system of equations [112]:

$$\begin{aligned}
\mathbf{R}(n)\Delta\mathbf{w}(n) &= \boldsymbol{\beta}(n) - \mathbf{R}(n)\hat{\mathbf{w}}(n-1) \\
&= \boldsymbol{\beta}(n) - \mathbf{R}(n-1)\hat{\mathbf{w}}(n-1) - \Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1) \\
&= \mathbf{r}(n-1) + \Delta\boldsymbol{\beta}(n) - \Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1)
\end{aligned} \tag{4.14}$$

Therefore, a solution $\Delta\hat{\mathbf{w}}$ can be determine by solving the auxiliary system of equations:

$$\mathbf{R}(n)\Delta\mathbf{w}(n) = \boldsymbol{\beta}_o(n) \tag{4.15}$$

Here:

$$\boldsymbol{\beta}_o(n) = \mathbf{r}(n-1) + \Delta\boldsymbol{\beta}(n) - \Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1) \tag{4.16}$$

The approximate solution of the original system (4.10) can then be determined as:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \Delta\hat{\mathbf{w}}(n) \tag{4.17}$$

Considering (4.16), this approach requires $\mathbf{r}(n)$ of the original system to be known at each time instance n . However, it can be shown that the residual vector for the solution $\Delta\hat{\mathbf{w}}(n)$ to the auxiliary system (4.15) is actually equal to $\mathbf{r}(n)$ of the original system (4.10) [112]:

$$\begin{aligned}
\mathbf{r}(n) &= \boldsymbol{\beta}(n) - \mathbf{R}(n)\hat{\mathbf{w}}(n) \\
&= \boldsymbol{\beta}_o(n) - \mathbf{R}(n)\Delta\hat{\mathbf{w}}(n)
\end{aligned} \tag{4.18}$$

The iterative approach can be formulated to solve the aforementioned sequence of system equations as illustrated in Table 4.2 [112]. At each time instance n , this approach requires a solution to an auxiliary problem (4.15) which deals with the increment of the filter weights, $\Delta\mathbf{w}(n)$, rather than the actual filter weights $\mathbf{w}(n)$, as described in the original problem, (4.10). This approach is preferable since it takes into account the accuracy of the previous solution through the residual vector $\mathbf{r}(n-1)$, as well as the variation of the problem to currently be solved through the increments $\Delta\mathbf{R}(n)$ and $\Delta\boldsymbol{\beta}(n)$ [112]. The proposed approach can also be applied to the exponentially weight RLS algorithm. This will be described in the next section.

Table 4.2 Iteratively solving for auxiliary equations

Step	Equation
	Initialisation: $\hat{\mathbf{w}}(-1) = 0, \mathbf{r}(-1) = 0,$ $\boldsymbol{\beta}(-1) = 0$
	for $n = 0, 1, \dots$
1	Find $\Delta\mathbf{R}(n)$ and $\Delta\boldsymbol{\beta}(n)$
2	$\boldsymbol{\beta}_o(n) = \mathbf{r}(n-1) + \Delta\boldsymbol{\beta}(n) - \Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1)$
3	Solve $\mathbf{R}(n)\Delta\mathbf{w}(n) = \boldsymbol{\beta}_o(n) \Rightarrow \Delta\hat{\mathbf{w}}(n), \mathbf{r}(n)$
4	$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \Delta\hat{\mathbf{w}}(n)$

4.6.1 Exponentially Weighted RLS Algorithm (ERLS)

Exponentially Weighted Recursive Least Squares (ERLS) is commonly used in dynamic systems to track time varying parameters. Generally, a weighting function is used to ensure past samples are gradually “forgotten” if the operating point of the system is constantly changing. Exponential forgetting factor or exponentially weighting algorithm is a familiar method that used in data weighting of the system, where the weighting function is given as [80]:

$$\eta(n,k) = \lambda^{n-k}, k = 1, 2, \dots, n \quad (4.19)$$

Here, λ is a positive constant factor known as the forgetting factor, ($0 < \lambda \leq 1$). According to equation (4.19), more weight is assigned to the recently recorded data. Approximately the value of $(1/1-\lambda)$ determines the memory size of the estimation algorithm. When the value of λ is near to one, this corresponding to long memory and if $\lambda = 1$ (ordinary RLS algorithm) the memory becomes infinite, whilst a small value of λ make the algorithm memory short [28, 80]. Therefore, the identification will improve and the estimation for time varying parameters will enhance, but the estimation is more affected by the noise. However, the minimisation of the sum of the squared error based on ERLS algorithms can be define as [112]:

$$J_{\min}(n) = \lambda^{n+1} \mathbf{w}^T(n) \mathbf{\Pi} \mathbf{w}(n) + \sum_{k=0}^n \lambda^{n-k} [d_r(k) - \mathbf{w}^T(n) \mathbf{u}(k)]^2 \quad (4.20)$$

where:

$\mathbf{\Pi}$ is a regulation matrix, usually selected as: $\mathbf{\Pi} = \delta \times \mathbf{I}_N$. \mathbf{I}_N is an N -by- N identity matrix, and δ is a small positive parameter (often referred to as the regulation parameter). Now, at each sample, the ERLS can be used to solve the linear equation described in (4.10). In weighting RLS, the auto-correlation matrix and cross-correlation vector are computing as [112]:

$$\begin{aligned} \mathbf{R}(n) &= \lambda \mathbf{R}(n-1) + \mathbf{u}(n) \mathbf{u}^T(n) \\ \boldsymbol{\beta}(n) &= \lambda \boldsymbol{\beta}(n-1) + d_r(n) \mathbf{u}(n) \end{aligned} \quad (4.21)$$

In order to iteratively compute the ERLS based on Table 4.2, the cross-correlation vector $\boldsymbol{\beta}_o(n)$ should be presented in terms of the filter inputs $\mathbf{u}(n)$ and the desired signal $d_r(n)$. By substitute (4.21) into (4.12), this results in [112]:

$$\begin{aligned} \Delta \mathbf{R}(n) &= (\lambda - 1) \mathbf{R}(n-1) + \mathbf{u}(n) \mathbf{u}^T(n) \\ \Delta \boldsymbol{\beta}(n) &= (\lambda - 1) \boldsymbol{\beta}(n-1) + d_r(n) \mathbf{u}(n) \end{aligned} \quad (4.22)$$

From (4.11) and (4.22) we achieve:

$$\Delta \mathbf{R}(n) \hat{\mathbf{w}}(n-1) = (\lambda - 1) [\boldsymbol{\beta}(n-1) - \mathbf{r}(n-1)] + \mathbf{u}(n) \hat{y}(n) \quad (4.23)$$

where, at each time instant (n), the estimated output signal is computed as:

$$\hat{y}(n) = \mathbf{u}^T(n) \hat{\mathbf{w}}(n-1) \quad (4.24)$$

Then, based on (4.23) and (4.3), the vector $\boldsymbol{\beta}_o(n)$ can be described as:

$$\boldsymbol{\beta}_o(n) = \lambda \mathbf{r}(n-1) + e_p(n) \mathbf{u}(n) \quad (4.25)$$

Finally, Table 4.3 summarises the steps to find the parameter vector $\hat{\mathbf{w}}$, and the computational effort of each step [112]. The overall complexity of the algorithm can be shown to be; $2N^2 + 3N + M_n$ multiplications and $2N^2 + 3N + A_n$ additions,

where N is the filter order, and M_n, A_n are the number of multiplications and additions required to solve the linear equation in step 5. Again, these numbers depend significantly on the specific algorithms chosen to solve this particular step [112]. For example, the matrix inversion lemma is one familiar technique to complete the division process in step 5. In this work, we consider the use of the DCD algorithm to achieve a computationally light solution to solving this problem.

Table 4.3 ERLS algorithm using auxiliary equations

Step	Equation	×	+
	Initialisation: $\hat{\mathbf{w}}(-1) = 0, \mathbf{r}(-1) = 0,$ $\mathbf{R}(-1) = \mathbf{\Pi}$		
	for $n = 0, 1, \dots$		
1	$\mathbf{R}(n) = \lambda \mathbf{R}(n-1) + \mathbf{u}(n)\mathbf{u}^T(n)$	$2N^2$	N^2
2	$\hat{\mathbf{y}}(n) = \mathbf{u}^T(n)\hat{\mathbf{w}}(n-1)$	N	$N-1$
3	$e(n) = d_r(n) - \hat{\mathbf{y}}(n)$	–	1
4	$\boldsymbol{\beta}_o(n) = \lambda \mathbf{r}(n-1) + e(n)\mathbf{u}(n)$	$2N$	N
5	$\mathbf{R}(n)\Delta\mathbf{w}(n) = \boldsymbol{\beta}_o(n) \Rightarrow \Delta\hat{\mathbf{w}}(n), \mathbf{r}(n)$	M_n	A_n
6	$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \Delta\hat{\mathbf{w}}(n)$	–	N

4.7 Coordinate Descent and Dichotomous Coordinate Descent Algorithms

There are many iterative methods to solve the normal linear equations in step 5 of Table 4.3. Solving the linear equations is equivalent to minimising the following function [112, 116]:

$$f(\Delta\mathbf{w}) = \frac{1}{2}\Delta\mathbf{w}^T \mathbf{R}\Delta\mathbf{w} - \Delta\mathbf{w}^T \boldsymbol{\beta}_o \quad (4.26)$$

Minimising this function determines the exact solution of the normal linear equations. Iterative methods considered to minimise $f(\Delta\mathbf{w})$ [117]. Typically, the iterative algorithms takes an initial estimation of the value denoted by $\Delta\mathbf{w}^0$ and at each cycle a new sequence will be constructed $\Delta\mathbf{w}^1, \Delta\mathbf{w}^2, \dots, \Delta\mathbf{w}^k$ [80]. At each iteration cycle the update of the next sequence $\Delta\mathbf{w}^{k+1}$ is selected to be in a descending

direction as $[f(\Delta\mathbf{w}^{k+1}) \leq f(\Delta\mathbf{w}^k)]$, and is preferred to be as $[f(\Delta\mathbf{w}^{k+1}) < f(\Delta\mathbf{w}^k)]$ [117]. In this way, at each step, the algorithm continues to move towards the minimum of the value of $f(\Delta\mathbf{w})$. Once the solution of the linear equations ($\mathbf{R} \Delta\mathbf{w}^k = \boldsymbol{\beta}_o$), approaches the desired result, the iteration process is halted and the estimated value, $\Delta\mathbf{w}^k$, is accepted [117].

Calculating the step from $\Delta\mathbf{w}^k$ to $\Delta\mathbf{w}^{k+1}$ depends on the choices of both the vector direction \mathbf{p}^k , and the step size (μ). Here, \mathbf{p}^k indicates the direction of movement from $\Delta\mathbf{w}^k$ to $\Delta\mathbf{w}^{k+1}$, and μ represents the step length along the line $\Delta\mathbf{w}^{k+1} = \Delta\mathbf{w}^k + \mu^k \mathbf{p}^k$ [117]. The step size (μ) is appropriately chosen to ensure that: $f(\Delta\mathbf{w}^{k+1}) = \min f(\Delta\mathbf{w}^k + \mu^k \mathbf{p}^k)$ [118]. The procedure of selecting the step size is known as a line search method. The main difference between the individual methods is the choice of update directions and the step size. However, it can be shown that setting the step size $\mu = \frac{\mathbf{p}^T \mathbf{r}}{\mathbf{p}^T \mathbf{R} \mathbf{p}}$ minimises the function $f(\Delta\mathbf{w} + \mu \mathbf{p})$ [112]. Therefore, to ensure a reduction in the step size, the direction (\mathbf{p}^k) should be chosen to be non-orthogonal to the residual vector (\mathbf{r}) ($\mathbf{p}^T \mathbf{r} \neq 0$) [116]. Details of the line search approach taken in this research are described in Table 4.4 [112, 116]. Here, N_u is the number of the iteration.

Table 4.4 Exact line search algorithm description

Step	Equation
	Initialisation: $\Delta\hat{\mathbf{w}} = 0, \mathbf{r} = \boldsymbol{\beta}_o$
	for $k = 1, \dots, N_u$
1	Choose a direction \mathbf{p} such that $\mathbf{p}^T \mathbf{r} \neq 0$
2	$\mathbf{v} = \mathbf{R} \mathbf{p}$
3	$\mu = \mathbf{p}^T \mathbf{r} / \mathbf{p}^T \mathbf{v}$
4	$\Delta\hat{\mathbf{w}} = \Delta\hat{\mathbf{w}} + \mu \mathbf{p}$
5	$\mathbf{r} = \mathbf{r} - \mu \mathbf{v}$

As shown in Table 4.4, the step size update is not a trivial task. It requires a matrix/vector division and multiplication. The coordinate decent algorithm (CD) is one approach which may be used to simplify the process. In the CD algorithm, the directions are selected based on the Euclidean coordinate ($\mathbf{p} = \mathbf{e}_i$). Here, only the i -th element of vector \mathbf{e}_i is one and the other elements are zeros [112]. As a result, step 2 in Table 4.4 which requires $O(N^2)$ matrix-vector multiplication is significantly simplified. This results in further simplification of the other steps in Table 4.4 (especially in step 3 and 4) as follows [112]:

$$\left. \begin{array}{l} \mathbf{v} = \mathbf{R}\mathbf{p} = \mathbf{R}^{(i)} \\ \mathbf{p}^T \mathbf{r} = r_i \\ \mathbf{p}^T \mathbf{v} = R_{i,i} \end{array} \right\} \mu = \frac{r_i}{R_{i,i}} \rightarrow \Delta \hat{w}_i = \Delta \hat{w}_i + \mu \quad (4.27)$$

Here, $\mathbf{R}^{(i)}$ is the i -th column of the matrix \mathbf{R} .

When the order ($i = 1, \dots, N$) of the direction is chosen cyclically, as shown in Table 4.5, the algorithm is known as a cyclic CD algorithm [113]. However, in adaptive filter applications the cyclic approach is not efficient, where at each time instant, N iterations are required [112]. According to [112], the order of coordinate direction can be chosen by selecting the leading index (i) element as given in (4.28).

$$i = \arg \max_{p=1, \dots, N} \left\{ |r_p| \right\} \quad (4.28)$$

where, $\arg \max$ is the “maximum argument”.

This leading index corresponds to the maximum absolute value of the residual element [$\max(|\text{residual element}|)$] [113]. In this way, instead of defining the cyclic order direction, the leading element is chosen to speed up the convergence rate of the adaptation process [112]. This procedure is known as the leading CD algorithm (Table 4.6). The leading CD algorithm requires one division, N multiplications and $2N$ additions [112, 113].

It worth noting that the DCD algorithm is derived from the CD techniques. The main difference between CD and DCD is the selection of the step size. Here, it is

chosen in a different way that can further simplify the computation load and preserve a faster convergence rate.

Table 4.5 Cyclic CD algorithm description

Step	Equation	×	+	÷
	Initialization: $\Delta\hat{\mathbf{w}} = 0, \mathbf{r} = \boldsymbol{\beta}_o, k = 0$			
	for $i = 1, \dots, N$			
1	$\mu = \mathbf{r}_i / R_{i,i}$			1
2	$\Delta\hat{\mathbf{w}}_i = \Delta\hat{\mathbf{w}}_i + \mu$		1	
3	$\mathbf{r} = \mathbf{r} - \mu\mathbf{R}^{(i)}$	N	N	
4	$k = k + 1$			
5	If $k > N_w$, algorithm stop			

Table 4.6 Leading CD algorithm description

Step	Equation	×	+	÷
	Initialization: $\Delta\hat{\mathbf{w}} = 0, \mathbf{r} = \boldsymbol{\beta}_o$			
	for $k = 1, \dots, N_u$			
1	$i = \arg \max_{p=1, \dots, N} \{ r_p \}$,		$N-1$	
2	$\mu = \mathbf{r}_i / R_{i,i}$			1
3	$\Delta\hat{\mathbf{w}}_i = \Delta\hat{\mathbf{w}}_i + \mu$		1	
4	$\mathbf{r} = \mathbf{r} - \mu\mathbf{R}^{(i)}$	N	N	

4.7.1 Dichotomous Coordinate Descent Algorithm

The DCD algorithm is similar to the CD algorithm which is based on an iterative approach to estimating N parameters within an estimation parameters vector, $\Delta\hat{\mathbf{w}}$. The DCD algorithm begins to evaluate the residual vector and, based on its amplitude, will update the parameters vector. Initially, the step size, μ is chosen such that it equals H . Then during each pass of the algorithm, the step size is halved ($\mu = \mu/2$, step 1). This divide by two process is very important from a hardware point of view. It allows a division operation to be replaced with a more computationally efficient shift register [113]. Here, the reduction of the step size is configured with M iterations. The exact number of M depends on the accuracy required by the application.

Table 4.7 Cyclic DCD algorithm description

Step	Equation	+
	Initialisation: $\Delta\hat{\mathbf{w}} = 0, \mathbf{r} = \boldsymbol{\beta}_o, \mu = H, k = 0$	
	for $m = 1, \dots, M$	
1	$\mu = \mu / 2$	
2	Flag = 0	
	for $i = 1, \dots, N$	
3	If $ r_i > (\mu / 2)R_{i,i}$	1
4	$\Delta\hat{\mathbf{w}}_i = \Delta\hat{\mathbf{w}}_i + \text{sign}(r_i)\mu$	1
5	$\mathbf{r} = \mathbf{r} - \text{sign}(r_i)\mu\mathbf{R}^{(i)}$	N
6	$k = k + 1, \text{Flag} = 1$	–
7	If $k > N_w$ algorithm stop	–
8	If Flag = 1, repeat for step 2	–

Table 4.7 shows the operational steps of the cyclic DCD algorithm [112, 113]. Step 1: On each pass of the algorithm, the step size is reduced until the update is complete and the required level of accuracy is reached [113]. Steps 2 - 3: The magnitude of the residual vector, \mathbf{r} , is analysed during each pass ($i = 1, \dots, N$). Two

outcomes are possible: 1) an unsuccessful iteration, where the condition set out in step 3 is not met. In this case, the solution and the residual vector are not updated, 2) A successful iteration, where the condition in step 3 is met. Here, the solution in steps 4 and 5 is updated [112]. Step 4 - 5: If the residual is sufficiently large (Step 3: successful iteration), one element of the parameter vector is updated by adding or subtracting the value of μ ; depending upon the polarity of r_i . Following this, the residual vector (\mathbf{r}) is updated (Step 5). For every change of the step size, the algorithm repeats this process until all elements in the residual vector \mathbf{r} become small enough that the set condition in step 3 results in an unsuccessful iteration [113], or the number of iterations reaches a predefined limit number (N_u) [112]. The iteration limit may be used to control the execution time of the algorithm.

As shown in Table 4.7, a major advantage of the DCD algorithm is that both multiplication and division operations can be avoided. This is advantageous from a digital hardware implementation point of view. According to Zakharov *et al.* [112], the upper bound of the number of additions using cyclic DCD is $A_n = N(2N_u + M - 1) + N_u$. Therefore, if $N_u \gg M$, the complexity of the DCD can be approximated by $2NN_u$. However, if (N_u) is small and $N_u \ll M$, the term NM will dominate the DCD computational effort [112]. The actual dominate term will be application specific. Here, in the system identification of a dc-dc converter, it is found that the second case is generally true; $N_u \ll M$. For this reason, a refined form of the DCD algorithm (Leading-DCD) that presented in [112] is considered. In this particular version of the algorithm, it is possible to eliminate the NM dominant term.

In the leading-DCD, at each iteration the algorithm begins to analyse the residual vector and determine the maximum absolute value of \mathbf{r} (Step 1, Table 4.8) [113]. This maximum absolute value of \mathbf{r} represents the identity of the i -th element (leading element) in $\Delta\hat{\mathbf{w}}$ to be updated [113]. Here, the update of the element is similar to the leading-CD algorithm. Table 4.8 summarises the operational steps of the leading-DCD algorithm [112, 113]. The number of additions here is limited to $A_n = (2N + 1)N_u + M$, however, this is based on the worst case scenario and only results when the update process completes N_u iterations and the condition in process 3 (Table 4.8) is not satisfied [115].

The DCD algorithms described have been successfully implemented in hardware using FPGA technology [112, 113], a 16 tap-weight FIR filter is implemented using a Xilinx-Virtex II FPGA running at 100 MHz clock frequency and the update rate up to 200 kHz. The performance of this filter is close to the conventional RLS method [112].

Table 4.8 Leading DCD algorithm description

Step	Equation	+
	Initialisation: $\Delta \hat{\mathbf{w}} = 0, \mathbf{r} = \boldsymbol{\beta}_0, \mu = H, m = 1$	
	for $k = 1, \dots, N_u$	
1	$i = \arg \max_{p=1, \dots, N} \{ r_p \}$, go to step 4	$N - 1$
2	$\mu = \mu / 2, m = m + 1$	
3	if $m > M$, algorithm stops	
4	if $ r_i \leq (\mu / 2) R_{i,i}$, then go to step 2	1
5	$\Delta \hat{\mathbf{w}}_i = \Delta \hat{\mathbf{w}}_i + \text{sign}(r_i) \mu$	1
6	$\mathbf{r} = \mathbf{r} - \text{sign}(r_i) \mu \mathbf{R}^{(i)}$	N

4.8 Pseudo-Random Binary Sequence and Persistence Excitation

To accurately identify the dynamic behaviour of the system and to improve the performance of the identification, the input signals are required to be rich in frequency content. This ensures that, the input signals are changed or are fluctuated sufficiently to provide adequate excitation to estimate the unknown system [72]. System identification algorithms typically use the input signals to update their parameters; a persistently excited input signal is crucial to update the estimated parameters properly. The key element in signal processing applications such as adaptive filters is to understand the characteristics of the correlation matrix which in turn leads to identify the discrete time linear system and discover if the input is persistently excited [80, 81]. The input is persistently excited if the correlation matrix is non-singular (determinant of $\mathbf{R} \neq 0$), this in turn means that the input power

spectral density $\Phi_u(e^{j\omega t})$ is non-zero [63, 81]. Accordingly, to ensure that the estimated parameters of the unknown system will convergence to their correct values, a higher order of persistently excitation signal should be applied to the system; which also means that a higher input power spectral density provides a better system estimation [63].

There are different types of input excitation signals that can be injected into the system during the identification process. These perturbed signals can take different forms such as sine wave, white noise, or impulse signal. A Pseudo Random Binary Sequence (PRBS) is another type of excitation signal that is commonly used in system identification, since it is frequency rich and contains a wide range of frequencies of interest that provides sufficient information for the identification of the system. The PRBS has very similar spectral properties to white noise [83, 119]. Therefore, it is possible to apply the PRBS to obtain a high order persistently excited signal to the system [63].

A PRBS is a periodic, deterministic, rectangular pulse sequence modulated in width (Fig. 4.4) [120]. This sequence is easily to generate without need of any random number in the generation using a set of shift registers and an exclusive-or gate (XOR) in the feedback; as depicted in Fig. 4.5, here a nine bits PRBS is utilised. This kind of the PRBS is known as a maximum length pseudo binary sequence (MLBS). The length or the period of MLBS sequence is $L = 2^m - 1$, where m is integer and represent the number of bits [84]. A MLBS is generated by iteratively performing the XOR operation between the k -th cell register and a specific r -th cell register (Table 4.9) [120]. For instance, the 9-bits MLBS can be achieved by performing the XOR between bit 5 and bit 9 (Fig. 4.5), resulting in $L = 511$ [82]. At least one value in the PRBS register should initially be set to logic one in order to generate the pseudo random sequence, $s(n)$ [63, 119].

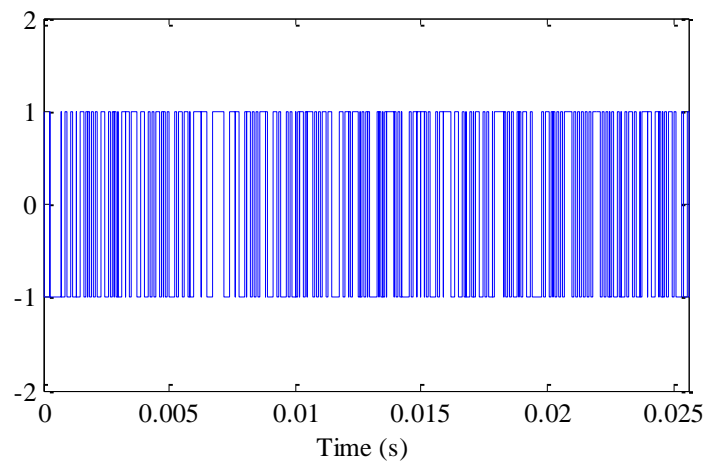


Fig. 4.4 Nine-bits single period PRBS

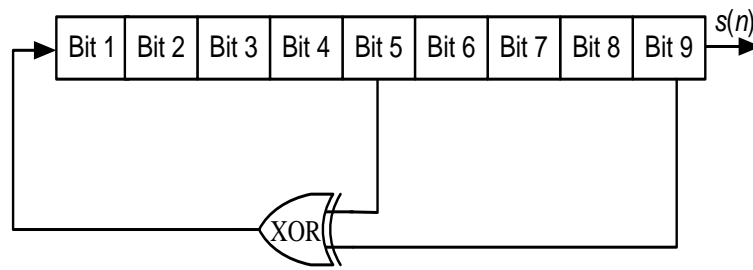


Fig. 4.5 Nine-bits shift register with XOR feedback for 511 maximum length PRBS generation

Table 4.9 Bit cell setup for different MLBS generation

Number of bits (m)	$L = 2^m - 1$	Bits in XOR operation k -th, r -th bits
2	3	1 and 2
3	7	1 and 3
4	15	3 and 4
5	31	3 and 5
6	63	5 and 6
7	127	4 and 7
8	255	2,3,4, and 8
9	511	5 and 9

The binary perturbation amplitude generated by shift registers is either one or zero. These logic levels are usually mapped into two possible amplitudes as presented in (4.29). An arbitrary symmetrical impulse sequence is resulted. For long sequence period this approximately has a zero mean value as described in equation (4.30) [63, 84].

$$u(n) = \begin{cases} +u_p & \text{if } s(n) = 1 \\ -u_p & \text{if } s(n) = 0 \end{cases} \quad (4.29)$$

$$M_P = \left| \frac{1}{L} \sum_{n=0}^{L-1} u(n) \right| = \frac{u_p}{L} \quad (4.30)$$

Equation (4.31) describes the auto-correlation properties of MLBS [63, 68], which illustrates that for very large value of L the auto-correlation can be approximated to a periodic sequences of impulses as it is shown in Fig. 4.6 [68]. The amplitude of these impulse is equal to u_p^2 at $n = 0, L, 2L, \dots$, otherwise it equal to $(-u_p^2/L)$ for all other n . As a result, the auto-correlation of MLBS is approximate to that of white noise [63]. Fig. 4.7 demonstrate the auto-correlation of a single period 9-bit PRBS.

$$\mathbf{R}_{uu}(n) = \frac{1}{L} \sum_{k=0}^{L-1} u(k)u(n+k) = \begin{cases} u_p^2 & n = 0, \pm L, \pm 2L, \dots \\ -\frac{u_p^2}{L} & \text{else} \end{cases} \quad (4.31)$$

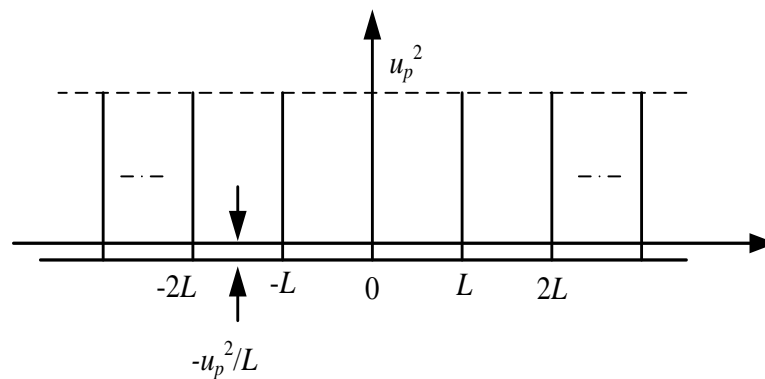


Fig. 4.6 Ideal auto-correlation of an infinite period of PRBS

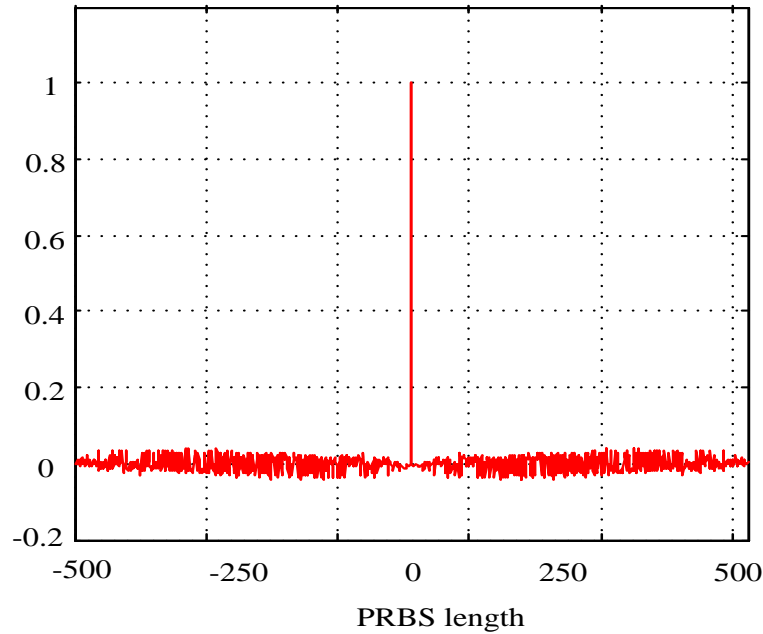


Fig. 4.7 Single period 9-bit auto-correlation of PRBS

4.9 Discrete Time Modelling of DC-DC Converter and Adaptive IIR Filter

Discrete time modelling of an SMPC is essential for a parametric identification process. The primary candidate model for system identification in this work is the voltage transfer function (control-to-output transfer function). However, the important factor in system identification is to select a low complexity model that has few parameters to estimate. In contrast, the selected model should be equivalent to the actual behaviour of the real system. ARMA model is the simplest model structure that is widely used in digital signal processing applications. The ARMA model structure is a combination between Auto-Regressive (AR) model and Moving-Average (MA) model. The AR process is defined as a linear mixture of predicts or past output values $y(n)$, in this way an all-pole-filter is created, with M order model. The MA model has an opposite representation of AR model, in this model (MA) the process output is equal to the combination of past input values; in this case an all-zero-filter with an N order model is constructed. Therefore, an MA model is inherently a stable filter; hence it has a similar form of FIR filter [80]. Finally, the ARMA model with order (M, N) can be constructed [70]:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\sum_{k=1}^N b_k z^{-k}}{1 + \sum_{k=1}^M a_k z^{-k}} = \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}} \quad (4.32)$$

From the general form of the direct realisation of the IIR filter (4.33), if $M = N$ and $b_0 = 0$. It can be deduced that, IIR filter has a counterpart form of ARMA model [80]:

$$y(n) = \sum_{k=0}^N b_k u(n-k) - \sum_{k=1}^M a_k y(n-k) \quad (4.33)$$

As expressed in Chapter 2, starting with the state space equivalent model of the buck converter circuit in continuous time domain, it can be shown that the control signal $d^*(s)$, to output voltage, $v_o(s)$, transfer function is described as follow (Fig. 4.1):

$$G_{dv}(s) = \frac{v_o(s)}{d^*(s)} = \frac{V_{in}(C R_C s + 1)}{s^2 LC \left(\frac{R_o + R_C}{R_o + R_L} \right) + s \left(R_C C + C \left(\frac{R_o R_L}{R_o + R_L} \right) + \frac{L}{R_o + R_L} \right) + 1} \quad (4.34)$$

The average continuous-time transfer function described in (4.34) can be converted to a discrete equivalent model using conventional continuous to discrete transformation methods, resulting in general a second order discrete transfer function:

$$G_{dv}(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (4.35)$$

Here, b_1 , b_2 , a_1 and a_2 are the parameters to be identified. They all depend on circuit component values and the sampling frequency. The input-output relation given in (4.35) may also be described as a linear difference equation:

$$v_o(n) + a_1 v_o(n-1) + a_2 v_o(n-2) = b_1 d^*(n-1) + b_2 d^*(n-2) \quad (4.36)$$

In this research, an IIR adaptive filter is employed to model the buck dc-dc SMPC. However, the DCD-RLS algorithm described in section 4.7 is normally applied with FIR adaptive filters. For this reason, an equation error approach is developed here

whereby an IIR filter is effectively derived from an equation error structure of two FIR filters, as it will be shown in the following section.

4.9.1 Equation Error IIR Adaptive Filter

There are two common paradigms to realise the adaptive IIR filter: 1) Output error scheme, and 2) Equation error scheme [80, 81]. Fig. 4.8 shows the block diagram of the adaptive output error IIR filter. Here, the input signal is applied to the both unknown system and to the numerator, $\mathbf{B}(z)$ of the IIR filter. The estimated signal, $\hat{y}(n)$ is then used as an input signal to the denominator, $\mathbf{A}(z)$ of the IIR filter. The error signal $e_p(n)$ is computed based on the differences between the desired and the estimated signals [80], hence the name of the output error. However, it is difficult to solve the cost function in equation (4.4) for output error IIR adaptive filter which is required a complicated mathematical analysis [81]. This can be solved by the second scheme of the adaptive IIR filter (equation error IIR filter) [80] which is effectively realised using two FIR filters. In this paradigm, the error signal is defined by an error equation rather than obtained directly from the output of the IIR filter as the case of the output error model [80].

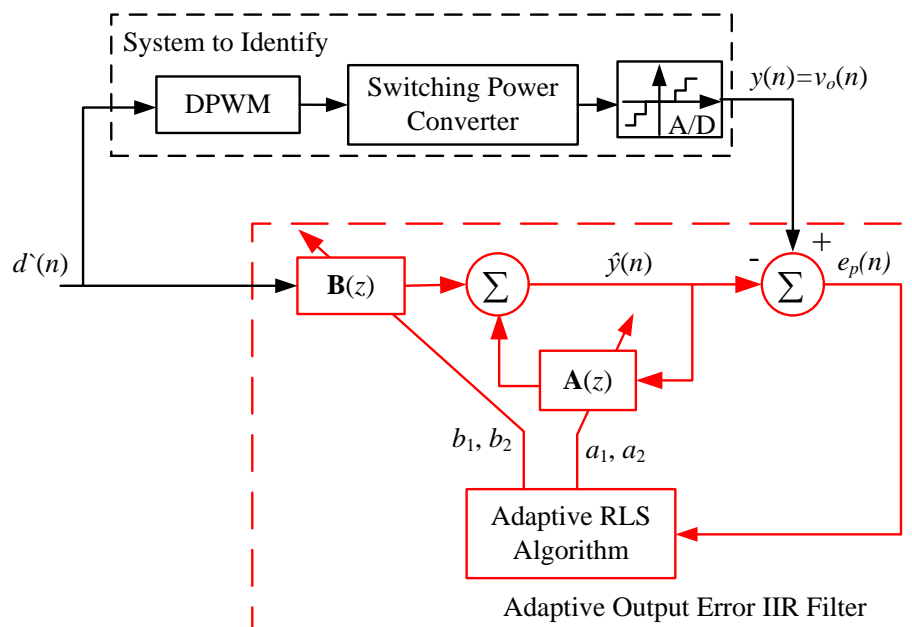


Fig. 4.8 System identification based on adaptive IIR filter using output error block diagram

In an equation error IIR filter, the input signal is applied to the unknown system and to the first FIR filter (Feed-Forward filter), thus the input data vector can be observed as in (4.37). The second FIR filter (Feed-back filter) utilises the desired signal, (illustrated in Fig. 4.9) and the output data vector can be given as in (4.38).

$$\mathbf{u}_{FIR1} = [d(n-1) \quad d(n-2) \quad \cdots \quad d(n-M)]^T \quad (4.37)$$

$$\mathbf{u}_{FIR2} = [-v_o(n-1) \quad -v_o(n-2) \quad \cdots \quad -v_o(n-N)]^T \quad (4.38)$$

$$\mathbf{w}_{FIR1} = [b_1 \quad b_2 \quad \cdots \quad b_M]^T \quad (4.39)$$

$$\mathbf{w}_{FIR2} = [a_1 \quad a_2 \quad \cdots \quad a_N]^T$$

Here, the second FIR filter does not use past adaptive filter output samples as in the output error structure. Instead, it uses the delayed samples of the desired signal. Thus, the minimisation criterion is analytically simple to derive using this structure of IIR filter, where the input and output signals are not function of the adaptive filter parameters [81], compare with the output error IIR structure. Consequently, the same data vector that is used in the basic identification model of ARX systems [63] can be observed in the equation error scheme. Therefore, the input/output difference equation can be written as: [80, 81]:

$$\hat{y}(n) = \sum_{k=1}^M b_k d(n-k) - \sum_{k=1}^N a_k v_o(n-k) \quad (4.40)$$

The prediction error is defined as:

$$\hat{e}_e(n) = y(n) - \hat{y}(n) \quad (4.41)$$

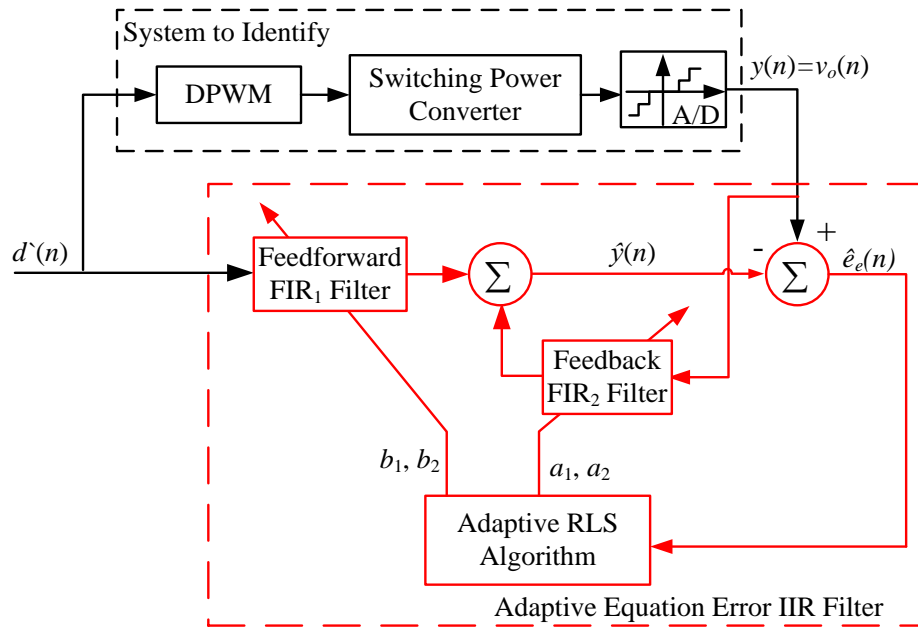


Fig. 4.9 System identification based on adaptive IIR filter using equation error block diagram

However, the update sequence for each FIR filters in Fig. 4.9 is not optimal using the DCD algorithm. Each filter requires an independent input data vector and adaptive algorithm to update a separate auto-correlation and cross-correlation matrix; as defined previously in (4.9). Accordingly, the overall complexity of the adaptive filter is increased. For this reason, this can be simplified by combining the input and output data from the unknown system and the parameter vectors into a single data and parameters vector [81]:

$$\boldsymbol{\varphi} = [-v_o(n-1) \quad \cdots \quad -v_o(n-k), \quad d^{\wedge}(n-1) \quad \cdots \quad d^{\wedge}(n-k)]^T \quad (4.42)$$

$$\boldsymbol{\theta} = [a_1 \quad \cdots \quad a_N, \quad b_1 \quad \cdots \quad b_M]^T$$

Therefore, the estimation output can be written as:

$$\hat{y} = \boldsymbol{\varphi}^T \boldsymbol{\theta} \quad (4.43)$$

4.10 Parameter Estimation Metrics and Validation

In parametric estimation, several metrics may be used to evaluate the results of the identification process. Prediction error, convergence rate and parameters estimation accuracy (parameters error) are the important metrics. These factors measure the performance of the estimation and determine how closely the identified model matches the actual system [31]. Accordingly, appropriate optimisation algorithms are required to minimise the approved metrics, where these algorithms are adaptively adjusted to the candidate model parameters until the objective function is satisfied. In adaptive signal processing algorithms, the quadratic error (LS method) is the popular factor to evaluate the performance of identification as expressed in equation (4.4), where the adaptive algorithms seek to minimise the summation of the square error by finding the optimal model parameters [28, 63]. In parametric estimation methods, the prediction error signal is the key element to minimise.

Convergence rate is another metric that measure the number of iterations or the time that the adaptive algorithms need to estimate the optimal parameters. A fast convergence is essential to track the time varying system and to identify the abrupt changes in the system [80]. For instance, automatic controllers of SMPCs require a fast convergence rate to tune the controller gains and quickly account for any changes in the system, such as the step load current change [31]. With respect to convergence time, the parameter accuracy, or the parameter error measurement, can be used to assess the true convergence of the parameters. The smaller the parameter error, \mathbf{e}_w , (4.44) the more accurate estimation of $\hat{\mathbf{w}}$, which in turn means that the parameters converge to the actual values of (\mathbf{w}).

$$\mathbf{e}_w = |\mathbf{w} - \hat{\mathbf{w}}| \quad (4.44)$$

To further validate the performance of adaptive algorithms in digital implementations, the finite word length (rounding-off-error and truncation) and quantisation of the A/D converter has to be taken into consideration as it has an effect on the parameter accuracy and can impact the overall performance of identification [31]. In addition, finite numeric precision of the input signals and internal microprocessor computations can introduce further errors in the system identification

process. In particular, parameters error and prediction error will be distorted due to these effects [81]. A method to help alleviate these side effects would be to increase the number of bits used within the internal computation, which will reduce the numerical error variance and thus will improve the estimation accuracy; in this case the adaptive algorithms will be numerically stable [80].

4.11 Model Example and Simulation Results

In order to test the concept of the proposed DCD-RLS identification scheme (Fig. 4.1), a voltage controlled synchronous dc-dc buck SMPC circuit has been simulated using MATLAB/Simulink (see appendix C). The circuit parameters of the buck converter are: $R_o = 5 \Omega$, $R_L = 63 \text{ m}\Omega$, $R_C = 25 \text{ m}\Omega$, $L = 220 \mu\text{H}$, $C = 330 \mu\text{F}$, $V_o = 3.3 \text{ V}$, $V_{in} = 10 \text{ V}$, $H_s = 0.5$. The series resistance ($R_S = 5 \text{ m}\Omega$) is added to measure the inductor current; thus the equivalent series resistance $R_q = R_L + R_S = 68 \text{ m}\Omega$. The R_{Dson} of the power MOSFET can also be added to the equivalent series resistance. The buck converter is switched at 20 kHz and the output voltage is also sampled at the same switching frequency rate. Consequently, the control-to-output voltage discrete transfer function of the SMPC can be calculated as follow:

$$G_{dv}(z) = \frac{0.226z^{-1} + 0.1118z^{-2}}{1 - 1.914z^{-1} + 0.949z^{-2}} \quad (4.45)$$

For the exponentially weighted leading element DCD-RLS algorithm (Table 4.8), the parameters are as follow: $N_u = 1$, $H = 1$, $M = 8$. The forgetting factor is chosen as $\lambda = 0.95$ and the typical value of regulation factor chosen as $\delta = 0.001$ [72]. For completeness, the simulation model includes all digital effects; such as ADC quantisation and sample and hold delays. To present the viability of the proposed DCD-RLS algorithm, an equivalent system based on a conventional exponentially weighted RLS (using matrix inversion lemma) is also simulated (Table 4.1). The same settings and initial conditions are used for both DCD-RLS and conventional RLS algorithms. For a regulated SMPC, the digital PID gains are tuned using a pole-zero matching technique that presented in Chapter 2. The PID controller is expressed as follows:

$$G_c(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \quad (4.46)$$

where, $q_0 = 4.127$, $q_1 = -7.184$, and $q_2 = 3.182$. It is important to mention that, the system model and the loop control design are simulated and evaluated in Chapter 2. Fig. 2.8 and Fig. 2.12 presented the tested results of the closed loop system.

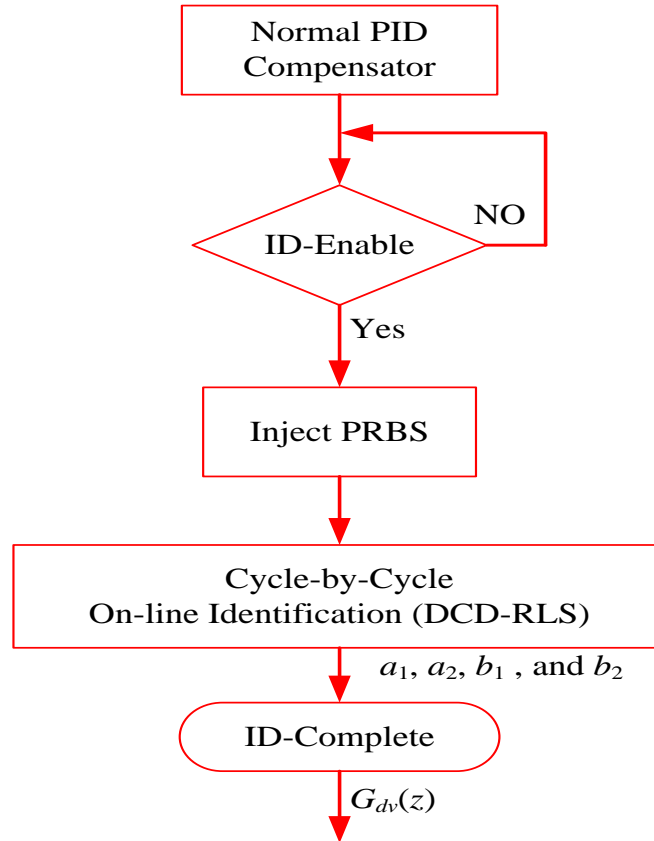


Fig. 4.10 The procedure of system identification

Based on the system in Fig. 4.1, the system identification sequence is described by the flowchart in Fig. 4.10, whilst the corresponding step-by-step results are illustrated in Fig. 4.11. Initially, the system is operating normally and is regulated by the PID compensator. When the identification process is enabled as shown in Fig. 4.11(e), a 9-bit PRBS is injected into the feedback loop as a frequency rich excitation signal as shown in Fig. 4.4. Here, as an example, the PRBS signal is injected during the steady-state period for 20 ms, superimposed with the control signal as depicted in Fig. 4.11(a, b). This is sufficient to determine the parameter convergence time. The PRBS

sampling frequency, f_p , is selected as 20 kHz. From this, the maximum PRBS pulse length is 511 ($L = 2^m - 1$), and the magnitude of PRBS signal, $\Delta_{PRBS} = \pm 0.025$. This is sufficiently small to cause excitation in the PWM output, but not enough to significantly compromise the normal operation of the SMPC; the output voltage ripple caused by this perturbation signal is approximately $\pm 2\%$ of the dc output voltage, as shown in Fig. 4.11(a). As each PRBS sample is injected, the DC components are removed from the input and the output, thus a zero mean value is determined in the input/output signal. The DCD-RLS is then measures the control output signal, $d(n)$, and the sampled power converter output voltage, $v_o(n)$. The algorithm is implemented and the IIR filter tap-weight estimation is updated. The effectiveness of the algorithm is verified in Fig. 4.11(c, d). The algorithm rapidly estimates the SMPC parameters [a_1 , a_2 , b_1 , and b_2] and then minimises the error prediction signal. It is worth noting that the initial value for each parameter is assumed to be zero. This demonstrates that prior knowledge of the SMPC parameters is not essential for convergence of the algorithm.

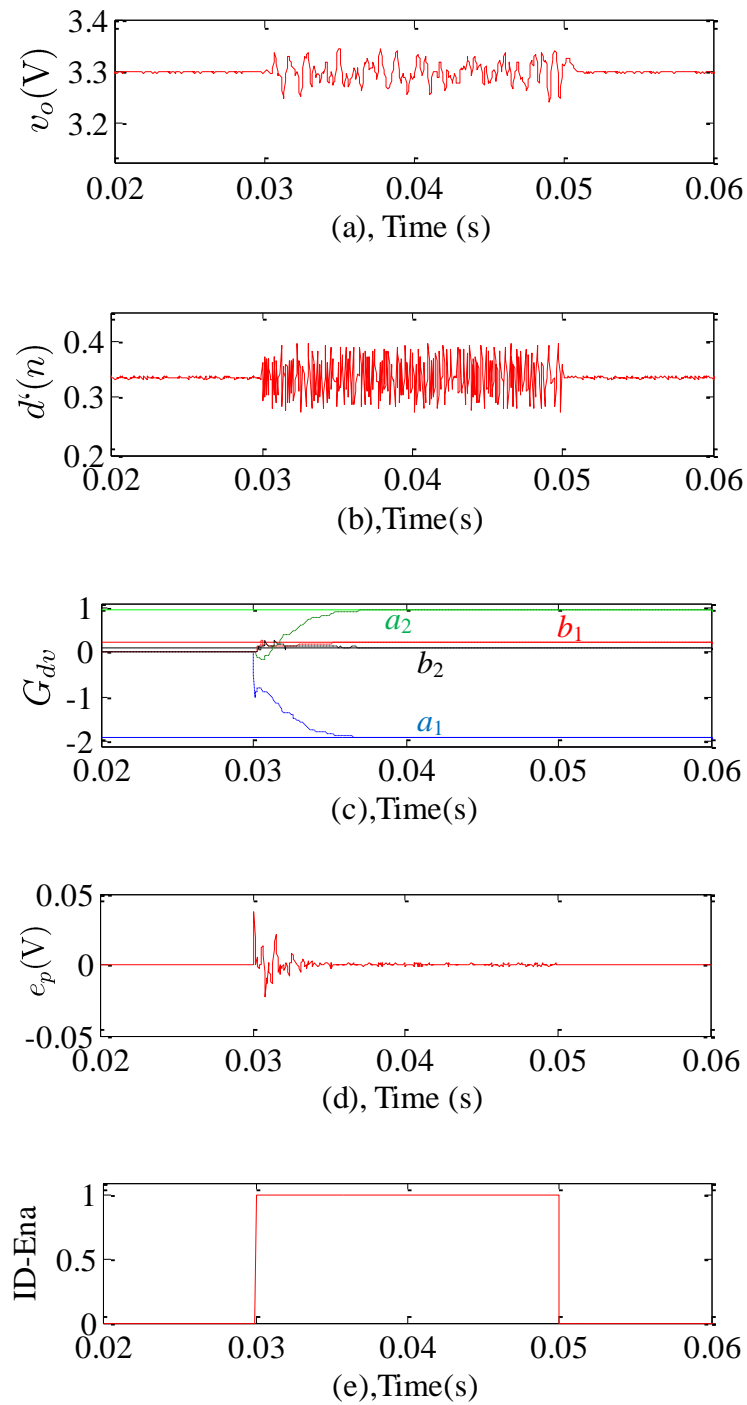


Fig. 4.11 Identification sequence, a: output voltage during ID, b: voltage model parameters ID, c: voltage error prediction, d. ID enable signal

Fig. 4.12 shows a comparison between the DCD-RLS identification algorithm and the classical RLS identification method. As depicted in Fig. 4.12, the DCD-RLS algorithm converges quickly (less than 10 ms) and identifies the unknown IIR filter coefficients. This in turn minimises the prediction error signal as shown in Fig. 4.13. Both techniques appear to converge to the same estimation values. The actual estimation error is illustrated in Fig. 4.14, where it can be seen that the performance of the DCD-RLS is comparable with the conventional RLS scheme. Fig. 4.14(a, b) demonstrates the parameters estimation error for the classical RLS scheme and DCD-RLS algorithm respectively. It is worth noting that the DCD-RLS estimation accuracy can further be improved by increasing the number of iterations (N_u), or the number of step size update (M). In the algorithm results are also presented where the effective resolution is reduced; $M = 4$. Fig. 4.15 compares the estimation performance of DCD-RLS with the conventional RLS method; the number of iterations, $N_u = 4$. It is observed that the DCD-RLS performance is enhanced and approaches the characteristics of the conventional RLS method. Making this adjustment will increase the execution time of the algorithm but, with many systems, a compromise between complexity and accuracy must be established. The estimation performance of the DCD-RLS is also compared to the leading CD algorithm. Fig. 4.16, clearly shows that the convergences of the parameters in the DCD-RLS algorithm is faster than those obtained with the CD algorithm; and as mentioned previously requires less computation. Further validation of the proposed algorithm is observed when comparing the frequency response characteristics of the estimated and calculated discrete time model as shown in Fig. 4.17. It can be seen that the DCD-RLS algorithm is closely matched to the control-to-output model of the of the dc-dc converter.

The versatility of the proposed DCD-RLS scheme has been verified with a range of the dc-dc discrete time models (duty-to-output voltage transfer function). In each case, the proposed method shown very promising results and can handle a wide range of uncertainty in the SMPC parameters. Table 4.10 presents three example systems, clearly showing how the algorithm closely matches the actual parameters for each buck converter model. Here, the parameters estimation accuracy has been measured at the final convergence values.

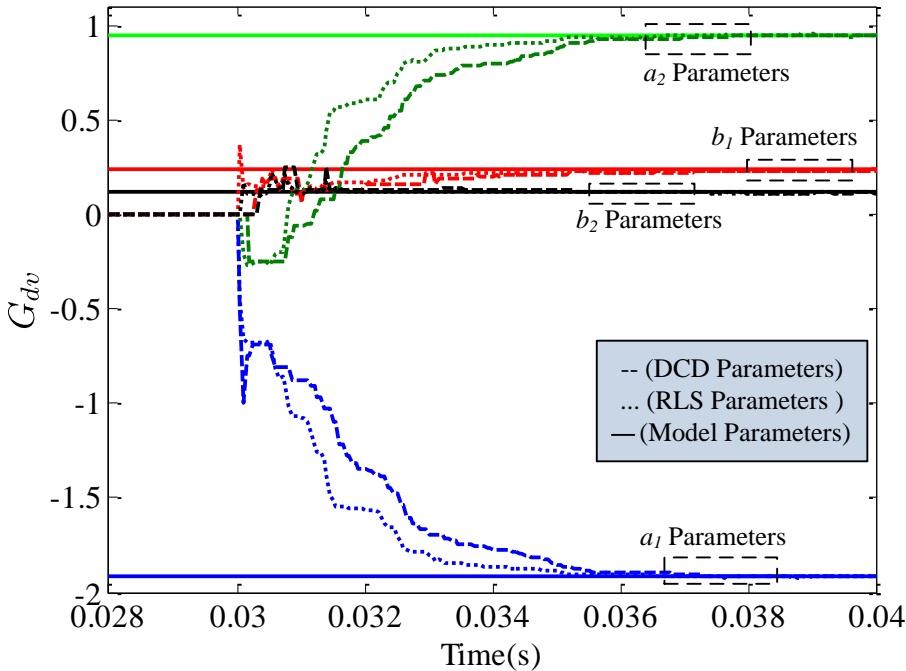


Fig. 4.12 Tap-weights estimation for IIR filter using DCD-RLS and classical RLS methods; compared with calculated model

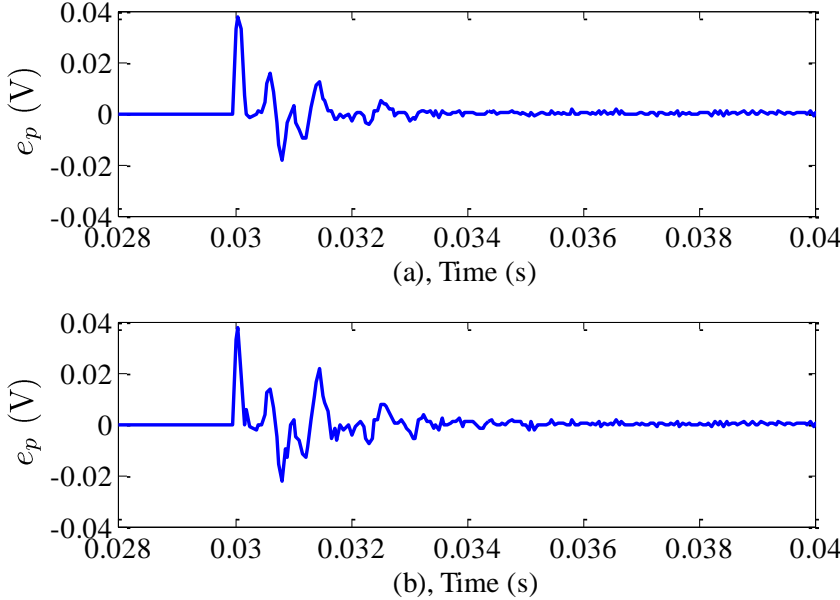
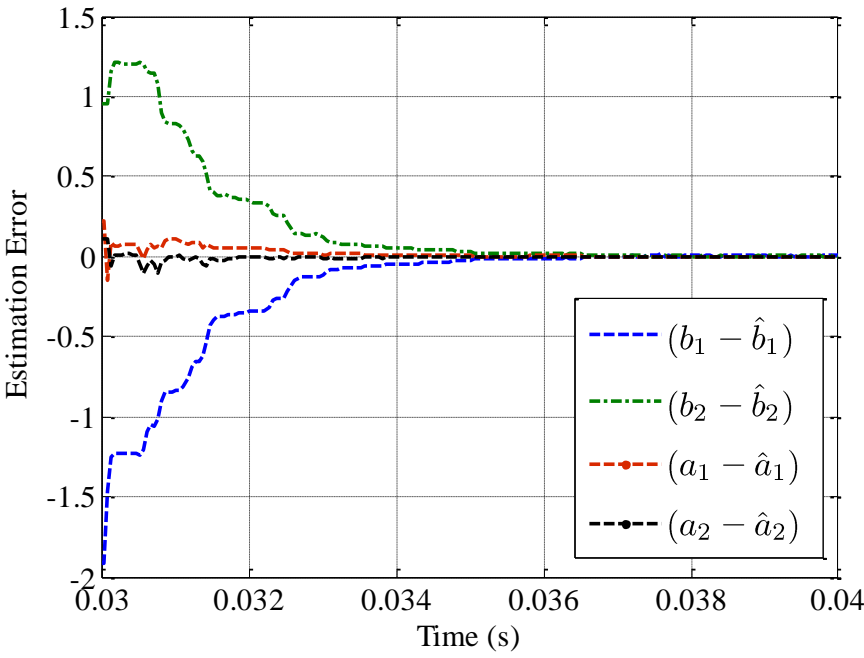
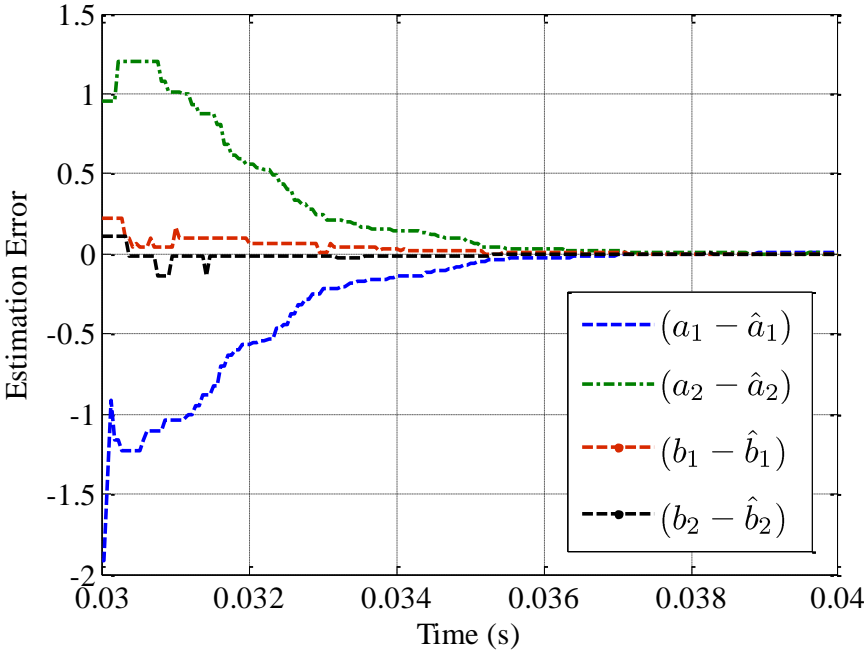


Fig. 4.13 Prediction error signals, a: classical RLS, b: DCD-RLS



(a)



(b)

Fig. 4.14 Parameters estimation error, a: classical RLS, b: DCD-RLS

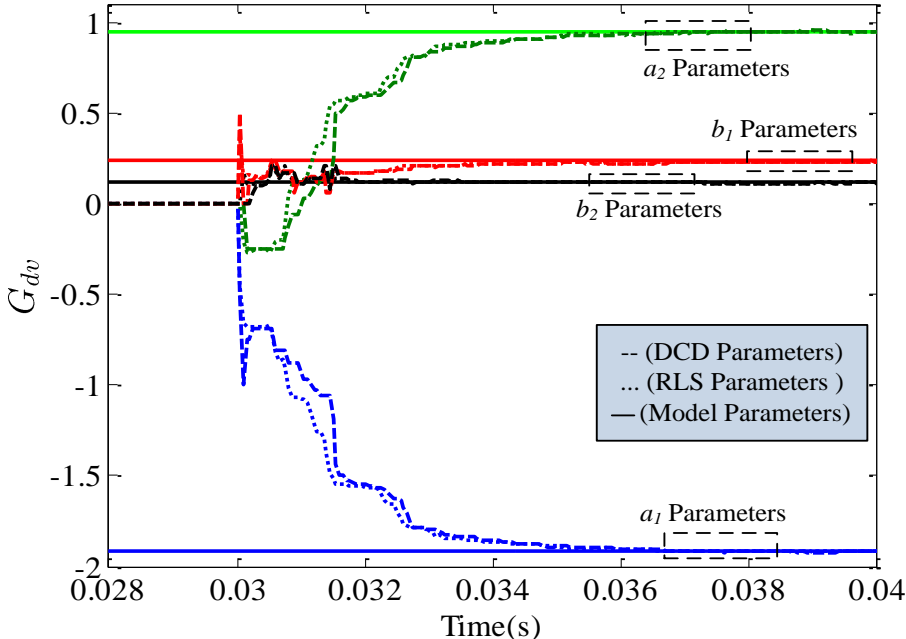


Fig. 4.15 Tap-weights estimation DCD-RLS at $N_u = 4$ and classical RLS

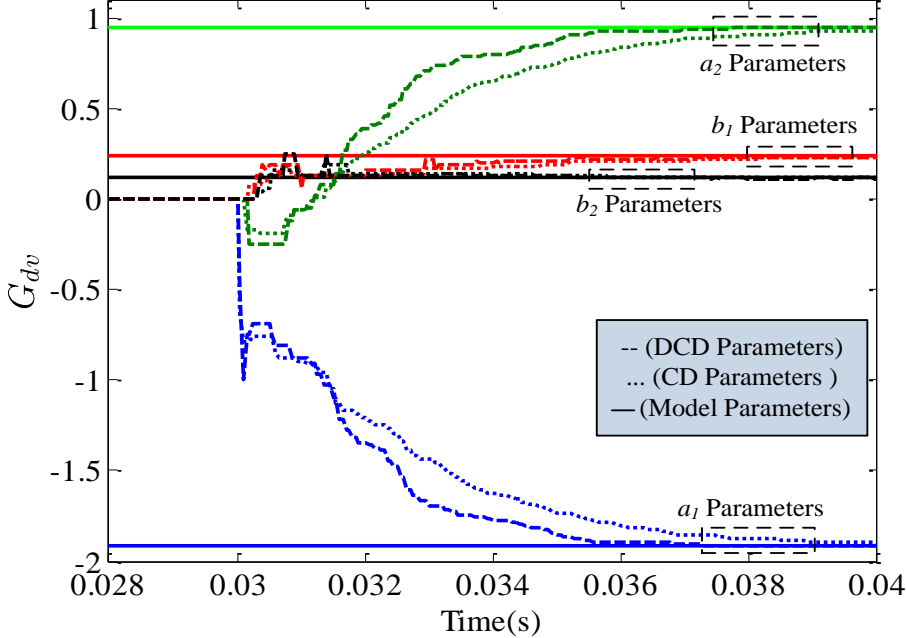


Fig. 4.16 Tap-weights estimation DCD-RLS and CD algorithms

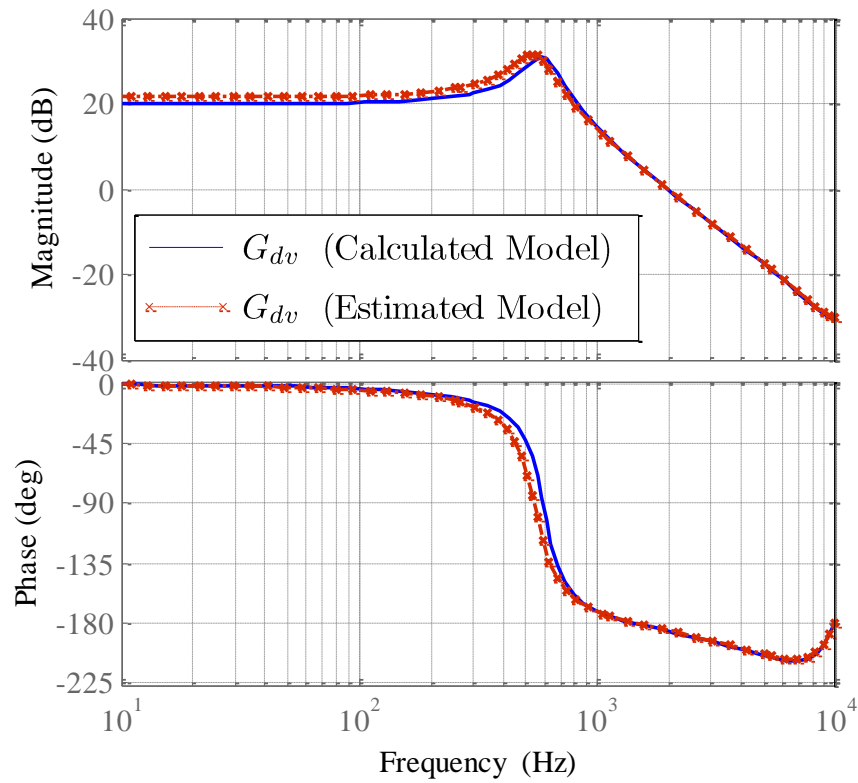


Fig. 4.17 Frequency responses for control-to-output transfer of function; estimated and calculated model

Table 4.10 Discrete time control-to-output transfer function identification

SMPC Model	Duty-to-Output Transfer Function	Estimation Parameters $\{b_1, b_2, a_1 \text{ and } a_2\}$
Buck 1	$G_{dv} = \frac{0.286 z - 0.1502}{z^2 - 0.925 z + 1.9401}$	$\{0.304, -0.179, -0.9414, 1.9258\}$
Buck 2	$G_{dv} = \frac{0.3422 z - 0.03409}{z^2 - 1.82 z + 0.8585}$	$\{0.3398, -0.062, -1.8203, 0.8594\}$
Buck 3	$G_{dv} = \frac{0.3862 z - 0.2321}{z^2 - 1.895 z + 0.9075}$	$\{0.414, -0.253, -1.875, 0.8867\}$

4.12 Adaptive Forgetting Strategy

Using recursive estimation and adaptive techniques for time varying systems is an important issue in a dynamic system where the behaviour, and hence parameters, of the system may change over time. It is important to monitor behavioural changes to optimise the controller design [121]. The RLS remains an effective identification method in tracking time-varying systems. However, rapid changes of parameters lead to numerical problems due to small data sets. For this reason, an appropriate choice of forgetting factor (λ) is vital, where the sensitivity of an estimate can be improved by adjusting the forgetting factor effectively. Generally, the forgetting factor is varying between $0 < \lambda \leq 1$ [74]. Small values of forgetting factor will lead to improvements in tracking ability. However, the RLS algorithm becomes more sensitive to noise. In contrast, large values of the forgetting factor will result in a poor tracking ability at slow parameter variations. However, the RLS algorithm is less sensitive to noise [122]. As a result, application of an adaptive forgetting factor method to a dc-dc converter system is proposed in order to make the identification algorithm more sensitive to change during system parameter changes, by assigning more weight to recent samples. Different techniques are proposed in the literature using the adaptive forgetting factor [121-124]. The accuracy, complexity, robustness, and the tracking ability are the main factors to consider when selecting the appropriate adaptive forgetting factor. In this thesis, a method from the telecommunication field is adopted (originally presented by Chia *et al.* [124]) to track the load changes in a closed loop dc-dc converter. This method uses a fuzzy variable forgetting factor RLS (FRLS).

4.12.1 Fuzzy RLS Adaptive Method for Variable Forgetting Factor

The FL system has been extensively used in various applications, and is popular in feedback control design, automatic control system, and system identification processes [125]. The FL system deals with linguistic variables rather than numerical numbers to achieve the design goal, without a mathematical model of the process. This is accomplished by converting the expert linguistics description into a desired strategy. Linguistic variables are forms of words that give the best description to input variables [126]. Fig. 4.18 illustrates the proposed adaptive forgetting factor (AFF) for a dc-dc converter using the FL system. Here, a fuzzy adaptation block is

designed to continually update the forgetting factor, based on two inputs: the squared prediction error and the squared change of prediction error $[e_p^2(n), \Delta e_p^2(n)]$. One of the best signals utilised in RLS in respect to monitoring and supervision the performance of the RLS, is the value of $e_p^2(n)$ [121]. The rate of the square prediction error is defined as:

$$\Delta e_p^2(n) = e_p^2(n) - e_p^2(n-1) \quad (4.47)$$

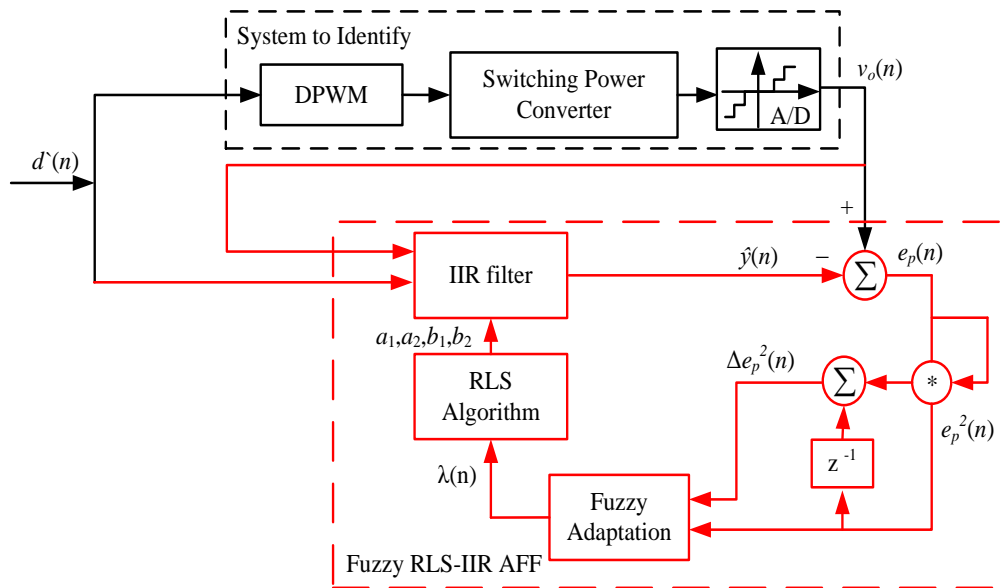


Fig. 4.18 The proposed system identification structure for a dc-dc converter based on RLS fuzzy AFF

The distinct advantage of this method is in respect to the nonlinear changes within the error signal. This is a result of the change in the model parameters. The FL rules based can be mapped this changes in the error signals and therefore, defining a better forgetting factor. A more precise dynamic and adaptation capability can be defined by using the two inputs, $[e_p^2(n), \Delta e_p^2(n)]$. The instantaneous change of the prediction error signal can be exploited within the FL system by utilising the $\Delta e_p^2(n)$ signal. This will provide invaluable assistance to the FL system for it to select the desired forgetting factor to be incorporated within the RLS algorithm [124].

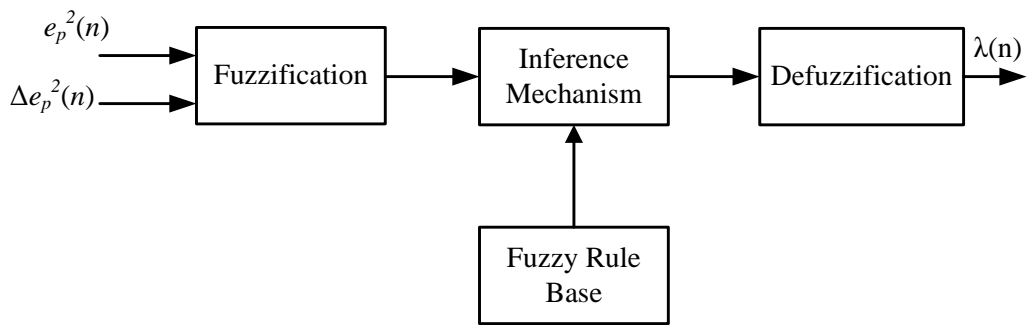


Fig. 4.19 General block diagram of the fuzzy logic system

Generally, the FL system or adaptation block in Fig. 4.18 is composed into three main sections (Fig. 4.19) [125]:

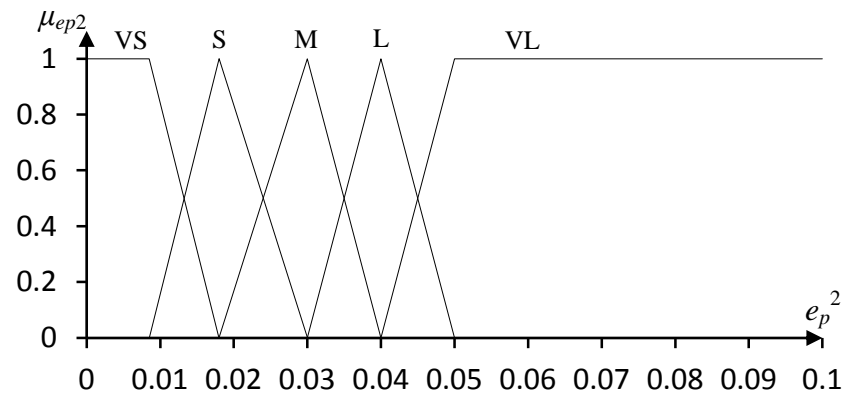
- 1- Fuzzification: in this phase the FL inputs $[e_p^2(n), \Delta e_p^2(n)]$ are converted into information that the inference mechanism can easily use to find the successful rules which map to one of the defined fuzzy sets. This is achieved by assigning each point in the input signal a membership degree. For simplicity of design a triangular membership functions are typically used in the fuzzification step [24]. Here, the number of membership functions are trimmed compared with [124], thus the computation load of the proposed solution of AFF will be reduced. However, the number of membership functions is mainly dependent on the accuracy of the change in prediction error. As shown in Fig. 4.20(a, b, c), there are five membership functions for $e_p^2(n)$, four membership functions for the second input $\Delta e_p^2(n)$, and five output membership functions. The linguistic labels are $\{Very\ Small, Small, Medium, Large, Very\ large\}$, but for brevity are referred to as $\{VS, S, M, L, VL\}$. The universe of discourse for the inputs is chosen between 0 and 0.1 as shown in Fig. 4.20(a, b), whilst the universe of discourse for the output is varied between 0 and 1 as shown in Fig. 4.20(c). The choice of these values will significantly affect the performance of AFF.
- 2- Inference Mechanism: the connection between the fuzzified input and the output fuzzy sets are achieved using the inference mechanism. Fuzzy rule base are used to obtain the combination between the fuzzified inputs to fuzzy

output. A set of (If-Then) expressions are used to describe these relations [125]. Table 4.11 shows the rule base that was developed in this AFF. A set of 20 rules are used in AFF [121, 124] for system identification of the dc-dc converter. When the prediction error abruptly increases, perhaps as a result of a step change in load, λ will quickly decrease to compensate for the change. This occurs when the prediction error signal is high, thus e_p^2 is VL and the Δe_p^2 is VL, a VS value is assigned to λ to increase the rate of convergence [124]. When the prediction error approaches zero, representing the steady-state, λ will settle to a constant value, typically approaching a high value. Here, e_p^2 is VS and Δe_p^2 is S then VL is assigned to the FL output. However, to prevent the λ becoming too small, and to obtain an acceptable convergence rate at start up, a stationary rule should be added [121]. This rule is activated when e_p^2 is VL and Δe_p^2 is S, thus λ is M.

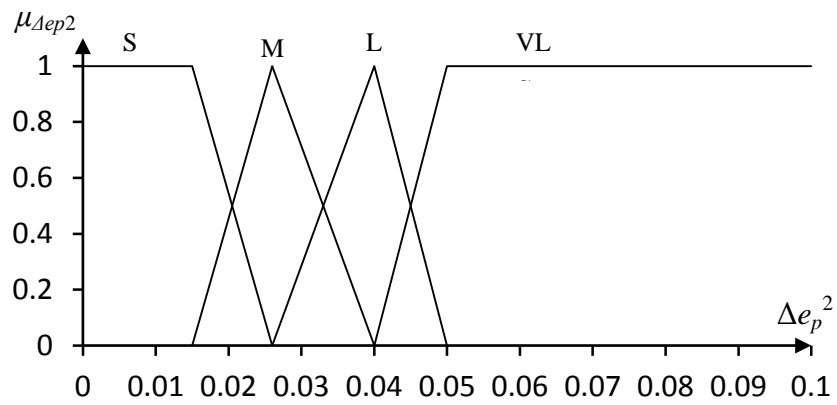
- 3- Defuzzification: as shown in Fig. 4.19, the input of this phase is the fuzzy set and the output is a real number. Centre of area or gravity is used to calculate the forgetting factor, as presented the following equation [24, 125]:

$$\lambda(n) = \frac{\sum_{j=1}^q \mu(\lambda_j) \lambda_j}{\sum_{j=1}^q \mu(\lambda_j)} \quad (4.48)$$

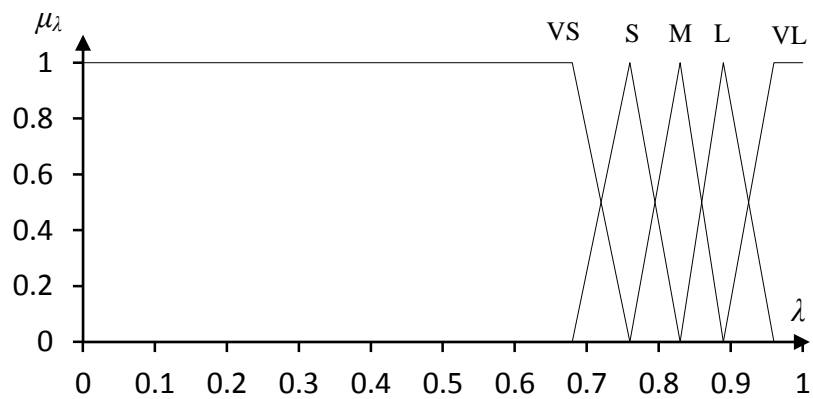
where, $\mu(\lambda_j)$ is the membership grade of the element λ_j and q is the number of the activated rules.



(a)



(b)



(c)

Fig. 4.20 Fuzzy logic input and output membership functions, a: e_p^2 , b: Δe_p^2 , c: λ

Table 4.11 The rule base for the forgetting factor (λ)

$e_p^2(n)$	VS	S	M	L	VL
$\Delta e_p^2(n)$					
S	VL	L	M	VS	M
M	L	L	M	S	VS
L	L	M	M	S	VS
VL	VL	M	S	VS	VS

4.13 Simulation Test

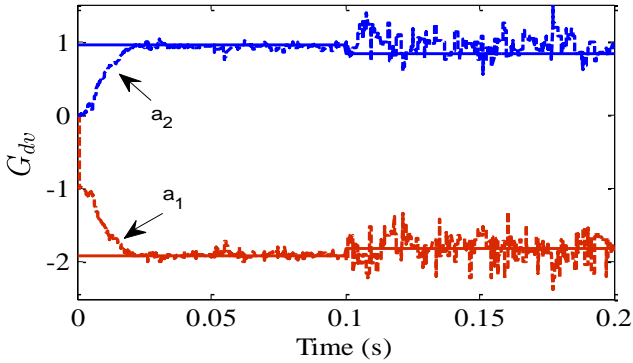
Similar circuit parameters to those outlined in section 4.11 are chosen. To demonstrate the effect of the forgetting factor for tracking the time varying parameters in a dc-dc converter, we assume that the load is changing abruptly from 5 Ω -to-1 Ω at each 0.1 s. This yield:

$$G_{dv}(z) \text{ at } (R_o = 5 \Omega) = \frac{0.2259z^{-1} + 0.1118z^{-2}}{1 - 1.915z^{-1} + 0.949z^{-2}} \quad (4.49)$$

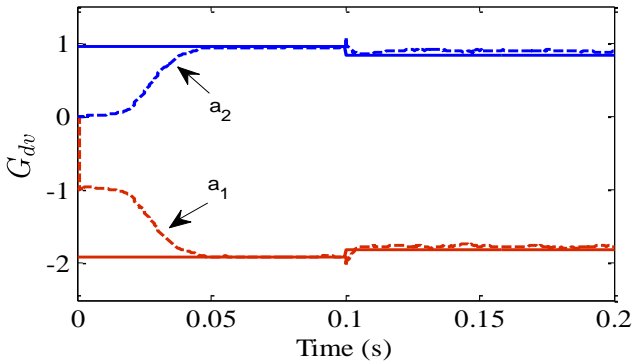
$$G_{dv}(z) \text{ at } (R_o = 1 \Omega) = \frac{0.2243z^{-1} + 0.1062z^{-2}}{1 - 1.811z^{-1} + 0.8437z^{-2}} \quad (4.50)$$

This considerable change in the load of the dc-dc converter is chosen to clearly observe the tracking ability of RLS algorithm. A 9-bit PRBS with $\Delta_{PRBS} = \pm 0.025$ amplitude is superimposed with the control signal as a rich excitation signal. Similar settings for the PID compensator, PRBS generator, and DCD-RLS are chosen as outlined in section 4.11. Initially the parameter values are set to zeros. The denominator parameters $[a_1, a_2]$ are the only parameters in the control-to-output transfer function presented in the estimation results. This is because the pole parameters vary significantly during the load change as described in equation (4.49) and (4.50), thus making the system disturbance easy to detect. The design of the FL system is carried out using MATLAB Fuzzy Logic toolbox.

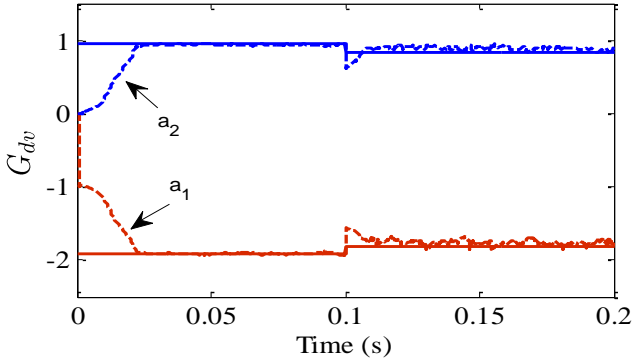
The tracking ability of the algorithm at a small fixed value of forgetting factor ($\lambda = 0.7$) is presented in Fig. 4.21(a). It can be seen that the convergence rate during initial start-up and at sudden load changes is rapid, but the estimated parameters chatter around the steady-state value making the estimation more sensitive to noise, and thus the final convergence values are difficult to determine. A similar setting is used with a higher forgetting factor ($\lambda = 0.99$) as shown in Fig. 4.21(b). As expected, the convergence rate is relatively slow during the initial start-up of the system, where it takes approximately 50 ms; but the estimated parameters are less sensitive to noise. However, as illustrated in Fig. 4.22, the prediction error signal provides an opportunity to both monitor the parameters change with the system, and to be included within the identification algorithm where, at initial system start-up and during load change, there is a greater disturbance in the prediction error signal. Therefore, a variable forgetting factor relying on prediction error signal can be applied to track this change in the system parameters, as well as to increase the convergence rate. The proposed AFF structure in Fig. 4.18 has been employed to track the abrupt load changes in the dc-dc converter. The result in Fig. 4.21(c) shows the effectiveness of the proposed AFF using the fuzzy logic system, where the algorithm successfully estimates the system parameters quickly during the initial start-up and at abrupt load changes with accurate estimation metrics. Fig. 4.23 show the change of variable forgetting factor. This forgetting factor is directly linked to the parameter variation during the load change. This clearly shows that at a high change of prediction error, the AFF produces a small λ and at a steady-state the forgetting factor then recovers to a high value (around $\lambda \approx 0.95$). The rapid change, and recovery, of the forgetting factor demonstrates the ability of the method to track parameter changes.



(a)



(b)



(c)

Fig. 4.21 Parameters estimation of control-to-output voltage transfer of a dc-dc converter at load changes from 5-to-1 Ω using DCD-RLS algorithm at a: $\lambda = 0.7$, b: $\lambda = 0.99$, c: fuzzy AFF

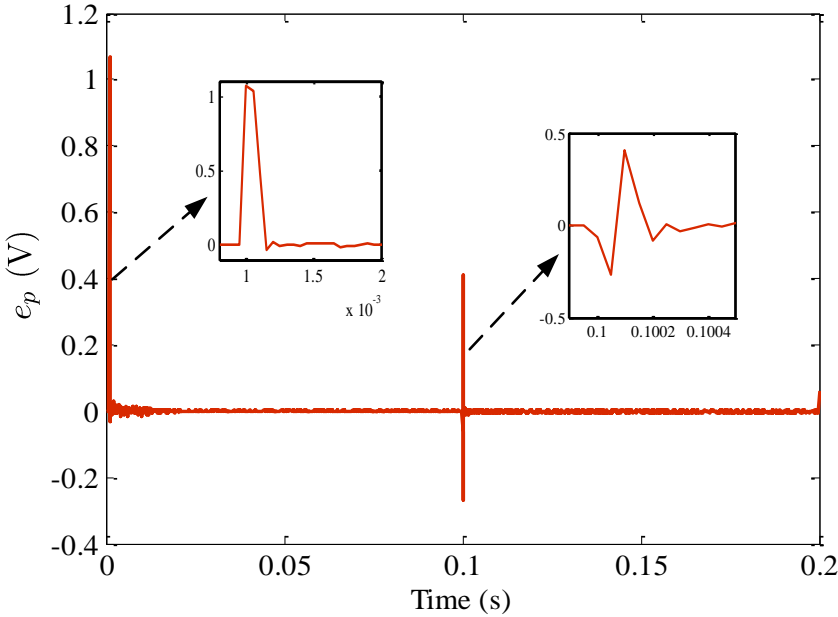


Fig. 4.22 Prediction error signal during initial start-up and at load change

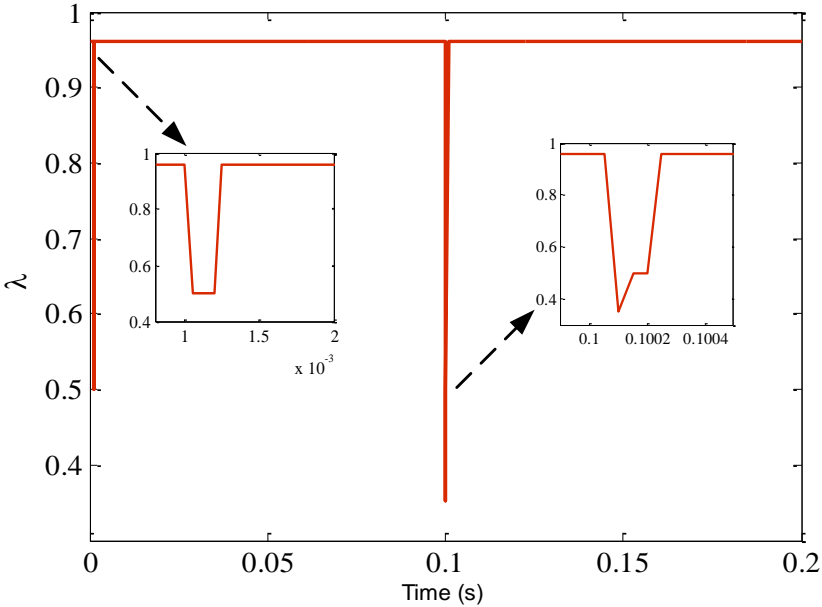


Fig. 4.23 Forgetting factor at initial start-up and at load change

However, continuous monitoring and estimation of time varying parameters required continuous injection of excitation signal in the feedback loop. Therefore, a small oscillation is continuously observed in the output response of SMPC. Here, the

perturbation signal is approximately equals to $\pm 2\%$ of the regulated dc output voltage which is chosen to be 3.3 V. Another concern is the resultant computational burden from applying the AFF/FL system. To reduce the system complexity, a FL system can be implemented using a two dimensional look-up table which ultimately reduces the amount of computation required. In real time implementation, a trade-off between the size of the look-up table and the estimation performance should be considered.

4.14 Chapter Summary

In the area of system identification, least square methods, like the basic RLS algorithm, provide promising results in terms of fast convergence rate, small prediction error, and accurate parametric identification. However, they often have limited application in SMPC and other low power, low cost applications due to computationally heavy calculations demanding significant hardware resources. Therefore, this chapter has introduced a novel computationally efficient DCD-RLS method to overcome some of the limitations of many classic RLS algorithms. The process is based on a proposed equation error IIR adaptive filter scheme, which is well suited for SMPC parameter estimation. The system identifies the IIR filter tap-weights on a cycle-by-cycle basis by injecting a perturbed input signal and monitoring the corresponding output response. The proposed solution demonstrated that the identification algorithm is able to work continuously in the control loop and quickly minimise the prediction error power; thus estimate the model parameters. Simulation results demonstrated that this approach exhibits very good identification metrics (convergence rate, parameters estimation, and prediction error) and the performance is comparable to more complex solutions such as recursive least squares techniques. The proposed scheme can be easily accompanied with many adaptive control solutions.

The second new scheme in this chapter is the adaptive forgetting factor based on fuzzy logic system. A two input, single output, fuzzy adaptive forgetting factor technique was applied to improve the estimation process during time varying system, such as abrupt load changes. This method has a simple structure, detecting the fast change in the system via sudden change in voltage prediction error. The AFF

structure has been validated by simulations and the results showed that the convergence rate and the estimation of the model parameters are very good in this method, where the abrupt changes of load are adapted to very quickly and smoothly via the variable forgetting factor which simply responds to parameters change. The adaptive forgetting factor method was successfully employed for the first time to the DCD-RLS algorithm. In summary, the proposed DCD-RLS algorithm can be implemented for many alternative applications where efficient and accurate parameter estimation is required.

Chapter 5

ADAPTIVE CONTROL OF A DC-DC SWITCH MODE POWER CONVERTER USING A RECURSIVE FIR PREDICTOR

5.1 Introduction

Many classical digital control systems for SMPCs suffer from inaccuracies in the design of the controller. Therefore, auto-tuning and adaptive digital controllers are playing an increasingly important role in SMPC systems. Adaptive digital controllers offer a robust control solution and can rapidly adjust to system parameter variations. This chapter presents a new technique for the adaptive control of power electronic converter circuits. The proposed technique is based on a simple adaptive filter method and uses a one-tap FIR-PEF. This is a computationally light technique based around the previously described DCD-RLS algorithm. In this case, the DCD-RLS algorithm is applied as the adaptive PEF. As a result, compared to the existing RLS algorithm, the computational complexity is reduced. Results show the DCD-RLS is able to improve the dynamic performance and convergence rate of the adaptive gains within the controller. In turn, this yields a significant improvement in the overall dynamic performance of the closed loop control system, particularly in the event of abrupt parameter changes. The results clearly demonstrate the superior dynamic performance and voltage regulation compared to conventional PID and adaptive LMS control scheme, with only a modest increase in the computational burden to the microprocessor. The proposed controller uses an adaptive Proportional-Derivative + Integral (PD+I) structure which, alongside the DCD algorithm, offers an effective substitute to a conventional PID controller. The non-adaptive integral controller (+I), introduced in the feedback loop, increases the excitation of the filter tap-weight and

ensures good regulation. The approach results in a fast adaptive controller with self-loop compensation. This is required to minimise the prediction error signal, and in turn minimise the voltage error signal in the loop by automatically calculating the optimal pole locations. The prediction error signal is further minimised through a second stage FIR filter (adaptation gain stage). This ensures the adaptive gains converge to their optimal value.

5.2 Self-Compensation of a DC-DC Converter Based on Predictive FIR

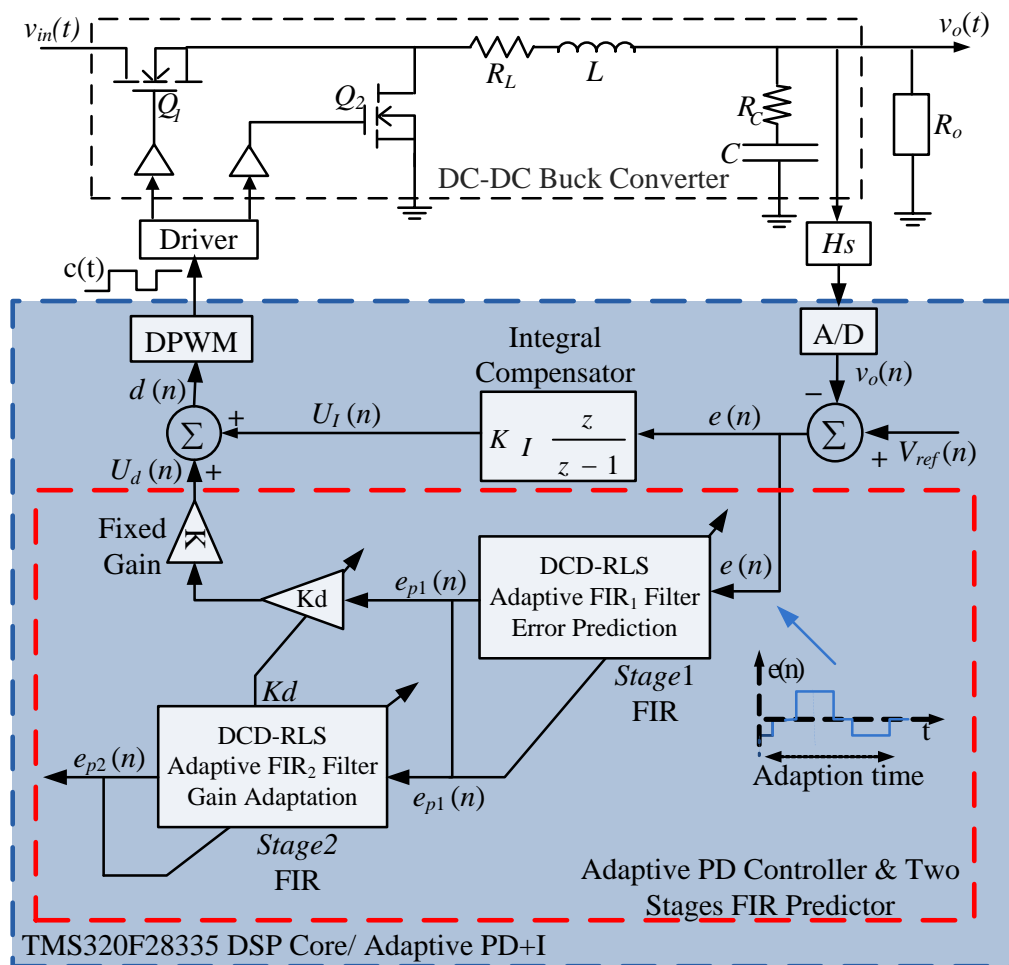


Fig. 5.1 Adaptive PD+I controller using one tap DCD-RLS PEF

Fig. 5.1 shows a block diagram of the proposed control scheme. Here, a similar PD control method to Kelly and Rinne [96, 97] is employed. However, a non-adaptive integral compensator is included in the feedback loop. This replaces a reference voltage feed-forward path in the original scheme. In this way, we look to achieve an

adaptive PD+I controller. The integral compensator has a number of roles. First, during the initial convergence time for the filter tap weight, the integral compensator is used to excite the system. The integral effectively introduces a transient, which is then amplified. This, in turn, initiates an oscillation in the control error signal. The excitation signal improves the convergence time of the adaptive filter, the time to obtain optimal taps weight parameters. It also allows the adaptive controller to work continuously in an on-line mode.

The advantage of this scheme is that the adaptive PEF rapidly “learns” the behaviour of the oscillation created by the integral compensator and rejects it from the control loop. Therefore, for the majority of the time a smooth output response is observed. The oscillation in the output voltage response only appears for a very brief period of time, sufficient for identification purposes. The final purpose of the integral compensator is more obvious; it helps output voltage regulation and ensures zero steady-state error in the system. When actually choosing the value of integral gain K_I , a compromise exists between the magnitude of the excitation signal in the loop and the need to avoid unwanted LCOs. At the output of the PD compensator, a fixed gain (K) is included in the control loop (Fig. 5.1). This gain increases the excitation until the adaptive filter weight converges to the optimal value. For the buck converter system under consideration, $K = L/T$, where T is the switching period and L is the inductor value [96].

5.3 Auto-Regressive / Process Generation, Identification

To implement a PEF as the central controller in the feedback loop requires the realisation of an Auto-Regressive (AR) process generator, followed by an AR identifier (Fig. 5.2).

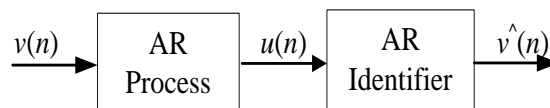


Fig. 5.2 Reconstruction of white noise

The AR process generator is defined as an all pole filter. The input is typically a white noise signal, $v(n)$, whilst the AR process output, $u(n)$, is normally a non-white signal [96]. The difference equation for this filter can be described as:

$$u(n) + a_1u(n-1) + \dots + a_Mu(n-N) = v(n) \quad (5.1)$$

Fig. 5.3 depicts the AR process generator model. To stabilize the AR filter, it is necessary to place all roots of the characteristic equation inside the unit circle of the z -plan (5.2). Therefore:

$$1 + a_1z^{-1} + a_2z^{-2} + \dots + a_Mz^{-N} = 0 \quad (5.2)$$

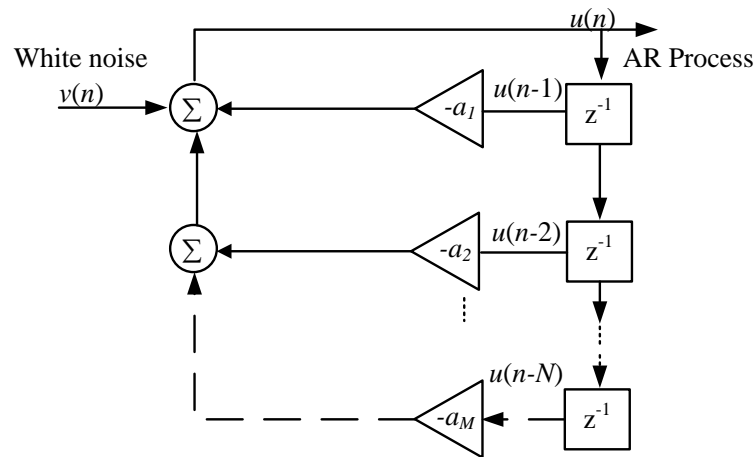


Fig. 5.3 AR process generator

Now, to identify the unknown AR process, and to reproduce the white noise input of the AR filter; a matching inverse filter must be designed; this is known as a Moving Average (MA) filter which is also referred to as an “all zeros filter” or FIR filter. Therefore, the output of the AR process filter is presented to the AR identifier (Fig. 5.4), whose transfer function is described as [80]:

$$H_{ARA} = \frac{V(z)}{U(z)} = \frac{1}{H_{ARG}} = \sum_{n=0}^N a_n z^{-n} \quad (5.3)$$

Here: H_{ARA} is the transfer function of AR analyser, H_{ARG} is the transfer function of the AR generator, and $a_0 = 1$. This filter is intrinsically stable.

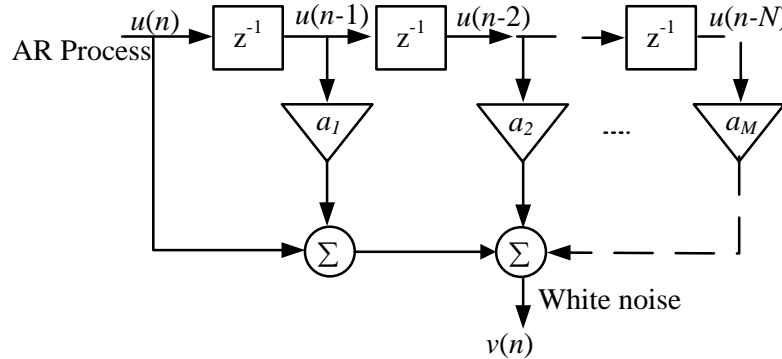


Fig. 5.4 AR process identifier

5.3.1 Relationship between Forward Prediction Error Filter and AR Identifier

A forward prediction filter is defined as a linear predictor that represents the combination of the past samples of the input signal $[u(n-1), u(n-2), \dots, u(n-N)]$. This filter consists of N unit delays and tap weights [80]. As shown in Fig. 5.5 the estimated output $\hat{y}(n)$ of the forward predictor is the prediction of the present input signal $u(n)$. Mathematically, the estimated output can be described as:

$$\hat{y}(n) = \sum_{k=1}^N w_k u(n-k) = \mathbf{w}^T \mathbf{u} \quad (5.4)$$

where:

$$\begin{aligned} \mathbf{w} &= [w_1 \quad w_2 \quad \dots \quad w_N]^T \\ \mathbf{u} &= [u(n-1) \quad u(n-2) \quad \dots \quad u(n-N)]^T \end{aligned} \quad (5.5)$$

The prediction error, $e_p(n)$, is defined as the difference between the desired signal and the estimated output signal, $\hat{y}(n)$. Here, the desired signal is equal to the input signal $u(n)$. Therefore:

$$e_p(n) = u(n) - \hat{y}(n) \quad (5.6)$$

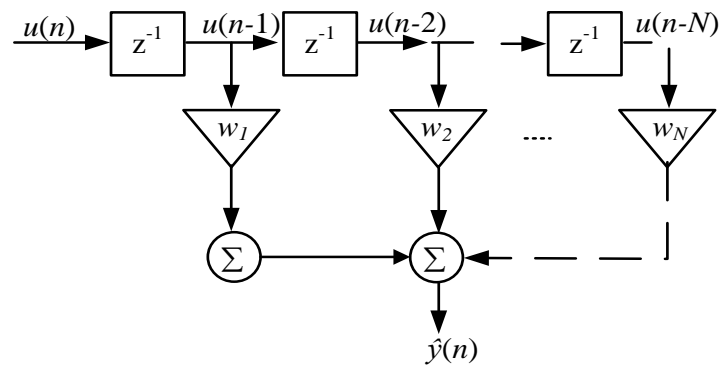


Fig. 5.5 One step ahead forward predictor

By substituting equation (5.4) into equation (5.6), and combining both terms into a single summation, the PEF can be expressed as:

$$e_p(n) = \sum_{k=0}^N w_f^k u(n-k) \quad (5.7)$$

where:

$$w_f^k = \begin{cases} 1 & k=0 \\ -w_k & k=1,2,\dots,N \end{cases} \quad (5.8)$$

This is depicted in Fig. 5.6. The length of the one step ahead forward prediction filter is one less than the length of the prediction error filter [compare equation (5.4) and (5.7)]. However, the number of delay elements and the order of both filters are the same. In such a way, the relationship between the PEF error filter and the AR identifier filter is illustrated as it shows in Fig. 5.7 [80].

In order to define the vector coefficients, \mathbf{w} , of the linear prediction filter, analytical calculation of the linear system equations is required. Adaptive algorithms such as LMS can be used to optimally calculate the vector coefficients (filter tap-weights) and reduce the computational load. Thus, an adaptive PEF can be applied to predict the AR process and reconstruct the original signal. The difference equation for the AR model has the same form as the difference equation of a PEF. Therefore, the forward prediction filter can be applied as the AR identifier [80, 96].

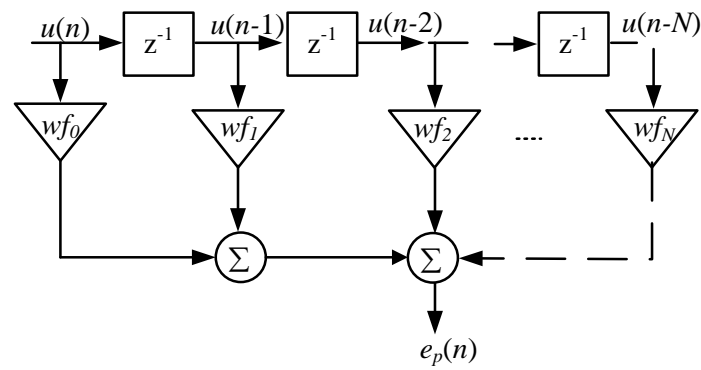


Fig. 5.6 Forward prediction error filter

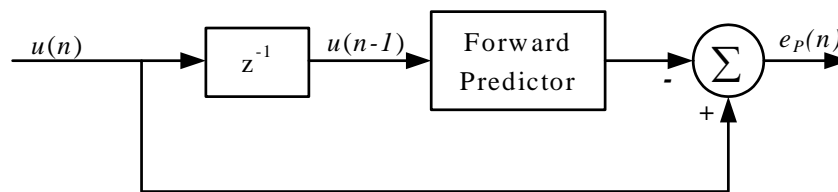
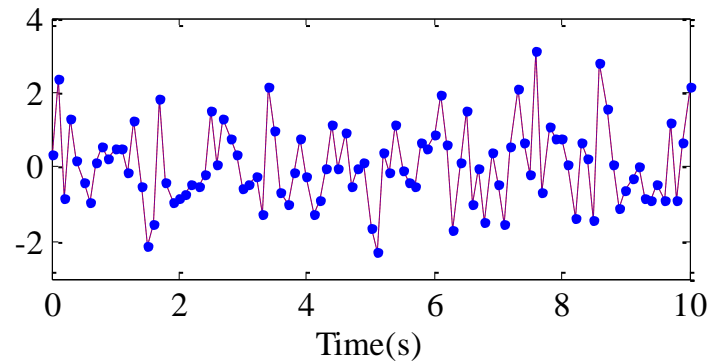


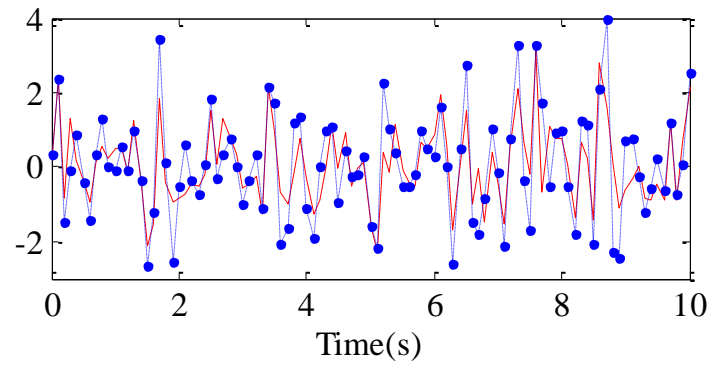
Fig. 5.7 Prediction error filter

To clearly understand the aforementioned description, suppose a second order AR model with constant filter coefficients ($a_1 = -0.1$ and $a_2 = -0.5$). The input of the filter is a random noise and the MA filter is designed to be exactly the inverse transfer function of AR filter (as first shown in Fig. 5.2). Therefore, assuming a perfect design, the poles of the AR filter are cancelled by the zeros of the MA filter. In this case, as shown in Fig. 5.8 (a), the input signal (random noise) and the output signal are identical (In Fig. 5.8 , the dotted line is the estimated output and the solid line is the actual input). However, in practice, the AR process generator is unknown; consequently, the MA filter must identify the process signal and attempt to reconstruct the original signal. Adaptive filter algorithms can also be used to optimally calculate the tap-weight of the MA filter. Therefore, an adaptive PEF can be applied to predict the AR process and reconstruct the original signal. This is clearly demonstrated in Fig. 5.8(b, c). Here, a one/two tap PEF is designed to identify the AR filter coefficients and reconstructed the input signal. It is worth noting that the AR process is assumed to be a second order filter, thus a two tap PEF will provide better prediction results than the suggested one tap PEF. This will lead to a more optimal estimation process. However, the first order PEF filter still produces a

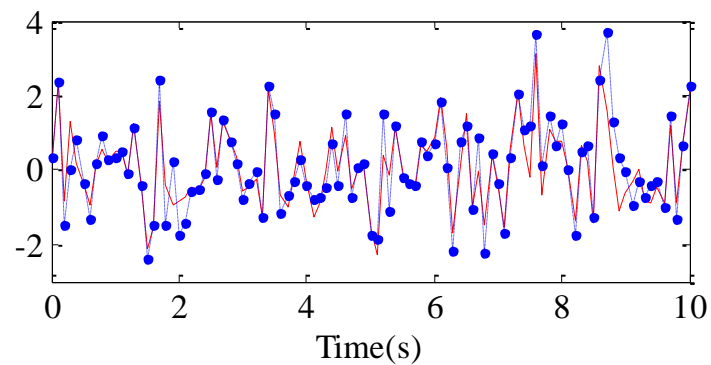
reasonable estimation of the AR model and reduces the computational overhead. In this specific application, this is deemed to be a worthwhile compromise.



(a)



(b)



(c)

Fig. 5.8 AR analyser, a: matched Inverse MA filter, b: one tap adaptive PEF, c: two tap adaptive PEF filter. The dotted line is the estimated output and the solid line is the actual input

5.3.2 One-Tap Linear FIR Predictor for PD Compensation

A digital FIR filter can be described, in difference equation form, by equation (5.1). From this, it is possible to describe the digital filter in the z -domain as:

$$\hat{Y}(z) = U(z)(w_1 z^{-1} + w_2 z^{-2} + \dots + w_N z^{-N}) \quad (5.9)$$

Referring to Fig. 5.6 and using equation (5.7) and (5.8), a FIR-PEF can therefore be represented in z -form as:

$$\frac{E_P(z)}{U(z)} = (1 + w_1 z^{-1} + w_2 z^{-2} + \dots + w_N z^{-N}) \quad (5.10)$$

The order of the digital filter candidate model is application dependent. SMPC systems can usually be satisfactorily compensated with a second order digital filter. However, as described in [57, 96], a second order minimum phase plant, such as a buck converter, can be compensated using a typical MA filter with β parameters only (5.11) of pole placement controller as:

$$G_c(z) = \frac{\beta_o + \beta_1 z^{-1} + \beta_2 z^{-2}}{(1 - z^{-1})(1 + \alpha z^{-1})} \left\langle \frac{D(z)}{E(z)} = \beta_o + \beta_1 z^{-1} + \beta_2 z^{-2} \right. \quad (5.11)$$

By setting the order of the PEF filter to one order lower than the plant, the PEF is equivalent to a controller design based on the pole placement method presented by Kelly and Rinne [57], where the order of the controller is also less than the order of plant by one. By comparing (5.10) with (5.11), a low order approximation FIR-PEF can actually be implemented as a gain controllable compensator [96]:

$$\beta_o + \beta_1 z^{-1} = K_d(1 + w_1 z^{-1}) \quad (5.12)$$

Equation (5.12) is equivalent to a PD controller. Importantly, it only requires one addition and one multiplication operation. A good quality regulator is required to optimally place the poles within the z -plane unit circle [57, 96]. This is the second purpose of the two-stage adaptive linear predictor shown in Fig. 5.1. In the first stage FIR, the adaptive algorithm places a zero (w_1) as close as possible to the dominant

poles of the auto-recursive model to minimise the error in the loop [127]. In the second stage, the adaptive algorithm estimates and adapts the gain (K_d) to minimise the prediction error in the adaptive filter. Conveniently, the adaptation of K_d is performed by the same mathematical process as the stage 1 FIR filter. However, here the FIR filter uses the prediction error signal $e_{p1}(n)$ as an input signal [96], rather than the voltage error signal (Fig. 5.1). Finally, automatic adjustment of (K_d, w_1) reduces the variance of the prediction error and influences the final controller output duty signal. This PD controller is then incorporated with the integral compensator to form the PD+I structure. As a result, a low complexity adaptive controller is achieved. This controller is capable of self-regulation, by finding the optimal control parameters, without explicit knowledge of the actual circuit parameters.

5.4 Least Mean Square Algorithm

Generally, to determine the optimal estimated parameter the adaptive algorithm requires solving a set of normal equations given by (4.10) [$\mathbf{w} = \mathbf{R}^{-1}\boldsymbol{\beta}$]. This can simply be achieved by performing the line search approach in the direction (\mathbf{p}) negative to the gradient vector of the minimisation function (5.13). This technique is known as ‘*Gradient descent*’ method [80].

$$\mathbf{p} = -\mathbf{g} = -\nabla f(\mathbf{w}) = -\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} \quad (5.13)$$

Generally, iterative computation of the filter weights take the following form [80]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu \mathbf{p}_n, \quad n = 0,1,\dots \quad (5.14)$$

Where, \mathbf{w} is the filter vector tap-weights, μ is the step size and \mathbf{p} is a vector direction. Now, by inserting (5.13) into (5.14), the iterative update of the filter coefficients can be written as [80]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu \mathbf{g}_n, \quad n = 0,1,\dots \quad (5.15)$$

In the LMS algorithm, the estimated filter coefficients are calculated based on the minimisation of the mean square error (MSE) [81]:

$$E[e^2(n)] = E[d_r(n) - \mathbf{w}^T \mathbf{u}(n)]^2 = E[d_r^2(n)] - 2\mathbf{w}^T E[d_r(n)\mathbf{u}(n)] + \mathbf{w}^T E[\mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{w} \quad (5.16)$$

Here, $E[.]$ is the expectation operation.

Equation (5.16) can be further simplified as [81]:

$$E[e^2(n)] = E[d_r^2(n)] - 2\mathbf{w}^T \boldsymbol{\beta} + \mathbf{w}^T \mathbf{R}\mathbf{w} \quad (5.17)$$

where:

$$\begin{aligned} \mathbf{R} &= E[\mathbf{u}(n)\mathbf{u}^T(n)] \\ \boldsymbol{\beta} &= E[d_r(n)\mathbf{u}(n)] \end{aligned} \quad (5.18)$$

It can be notice that equation (5.17) is a quadratic function of the filter tap-weights \mathbf{w} ; thus there is only one value that results in a minimum mean square error. This value is founded at the optimal value of the filter tap-weights. The optimal value is computed by setting the derivative with respect to \mathbf{w} equal to zero [81]:

$$\mathbf{g}_n = \frac{\partial E[e_p^2(n)]}{\partial \mathbf{w}} = -2\boldsymbol{\beta} + 2\mathbf{R}\mathbf{w} \quad (5.19)$$

$$\mathbf{g}_n = 2(\mathbf{R}\mathbf{w} - \boldsymbol{\beta}) = 0 \Rightarrow \mathbf{w}_{opt} = \mathbf{R}^{-1}\boldsymbol{\beta} \quad (5.20)$$

Here, \mathbf{w}_{opt} is the optimal solution of the linear equation in (5.20). This solution is known as the Wiener solution [81]. By substituting (5.19) into (5.15), the update coefficients equation can be represented as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \mathbf{R}\mathbf{w}(n) + \mu \boldsymbol{\beta} \quad (5.21)$$

In summary, to find the optimum filter coefficients: 1) at each iteration compute the gradient vector using (5.19) and then 2) update the tap-weights vector using (5.21). However, in real time implementation, the computation of \mathbf{R} matrix and $\boldsymbol{\beta}$ vector is not available. This can be simplified by using the instantaneous value of vector ($\boldsymbol{\beta}$) and the matrix (\mathbf{R}) instead of their actual value [80, 81]:

$$\begin{aligned}\hat{\mathbf{R}} &= \mathbf{u}(n)\mathbf{u}^T(n) \\ \hat{\boldsymbol{\beta}} &= d_r(n)\mathbf{u}(n)\end{aligned}\tag{5.22}$$

where, $\hat{\mathbf{R}}$ and $\hat{\boldsymbol{\beta}}$ are the instantaneous estimation of \mathbf{R} and $\boldsymbol{\beta}$.

From this, equation (5.19) can be written as [81]:

$$\begin{aligned}\mathbf{g}_n &= 2\left(-d_r(n)\mathbf{u}(n) + \mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}(n)\right) \\ &= 2\mathbf{u}(n)[\mathbf{u}^T(n)\mathbf{w}(n) - d_r(n)] \\ &= -2e(n)\mathbf{u}(n)\end{aligned}\tag{5.23}$$

By inserting (5.23) into (5.15), the update coefficients vector can be given as [80]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{u}(n)\tag{5.24}$$

The aforementioned procedure is known as LMS algorithm. Step-by-step operation of the LMS algorithm is depicted in Table 5.1 and Fig. 5.9 [80]. It can be seen that the LMS is a simple and low complexity algorithm, where at each iteration it requires only $N + 1$ multiplications for the error generation (Step 2) and $N + 2$ multiplications (Step 3) for the update of the filter coefficients [81]. However, the major drawback of the LMS algorithm is the speed of convergence, since there is only one parameter (μ) to control the convergence rate. The convergence of the LMS algorithm depends mainly on the step size factor, μ . Generally, the rate of convergence is inversely proportional to the step size. If μ is large, the convergence is relatively fast, but less stability is observed around the minimum value. On the other hand, if the step-size is small the convergence rate will be slow but more stable around the minimum point [80, 81].

Table 5.1 LMS algorithm operation

Step	Equation
	Initialisation: $\hat{\mathbf{w}}(0) = 0$, $\mathbf{u}(0) = 0$, $\mu =$ positive constant value
	for $n = 1, 2, \dots$
1	$\hat{y}(n) = \mathbf{u}^T(n)\hat{\mathbf{w}}(n)$
2	$e(n) = d_r(n) - \hat{y}(n)$
3	$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu e(n)\mathbf{u}(n)$

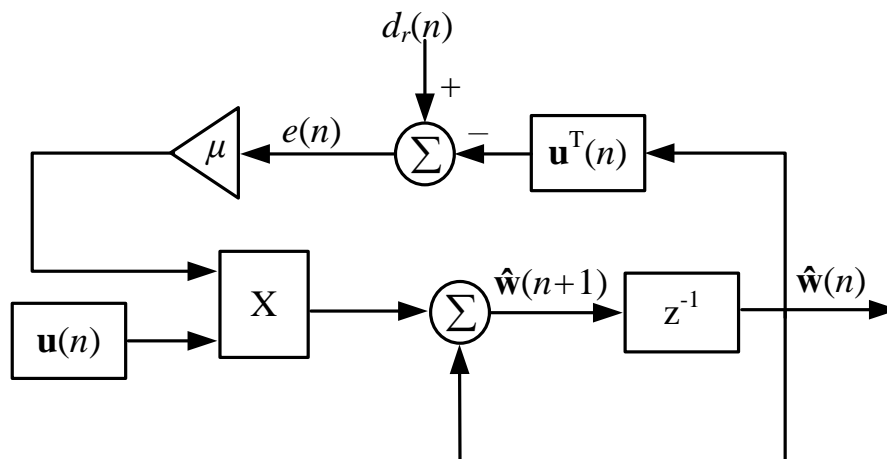


Fig. 5.9 Closed loop LMS system block diagram

5.5 Simulation Results

The proposed DCD-RLS adaptive control scheme (Fig. 5.1), for voltage controlled synchronous dc-dc buck SMPC circuit has been simulated using MATLAB/Simulink. The circuit parameters of the buck converter are the following: $R_o = 5 \Omega$, $R_L = 63 \text{ m}\Omega$, $R_C = 25 \text{ m}\Omega$, $L = 220 \text{ }\mu\text{H}$, $C = 330 \text{ }\mu\text{F}$, $V_o = 3.3 \text{ V}$, and $V_{in} = 10 \text{ V}$. The buck converter is switched at 20 kHz using conventional pulse width modulation. The output voltage is also sampled at 20 kHz. For the DCD-RLS algorithm, the parameters are as follow: $N_u = 1$, $H = 1$, $M = 4$. For completeness, the simulation model includes all digital effects, such as ADC, quantisation, and sample and hold delays (see appendix C). To present the feasibility of the proposed DCD-RLS algorithm, an equivalent system

based on the conventional LMS adaptive controller presented in [96, 97] is also simulated. A second alteration to the original structure in [96, 97] is made by replacing the original LMS-PEF with a DCD-RLS-PEF. The advantages of this change will be demonstrated in the following sections.

5.5.1 Reference Voltage Feed-Forward Adaptive Controller

Initially, the original reference voltage feed-forward structure presented in [96] is simulated and a comparison between the LMS with different step size (μ) values and the proposed adaptive DCD-RLS algorithm is made. The results are shown in is Fig. 5.10.

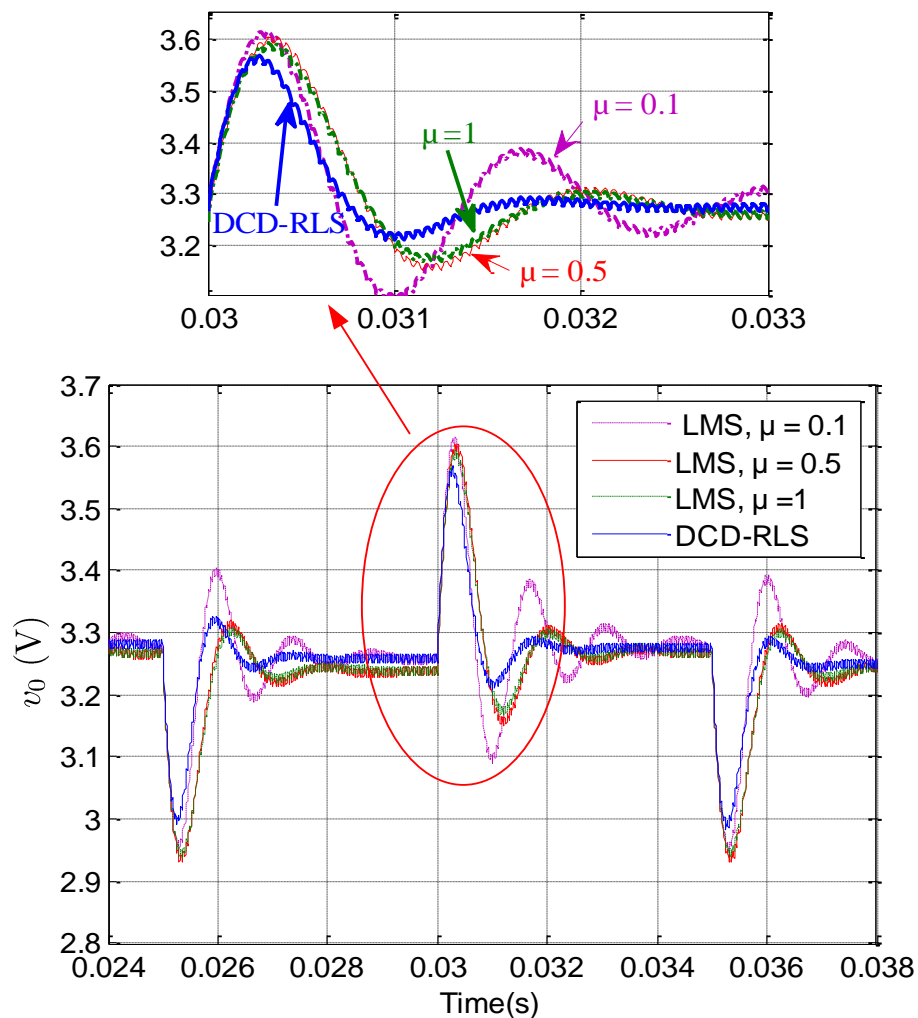
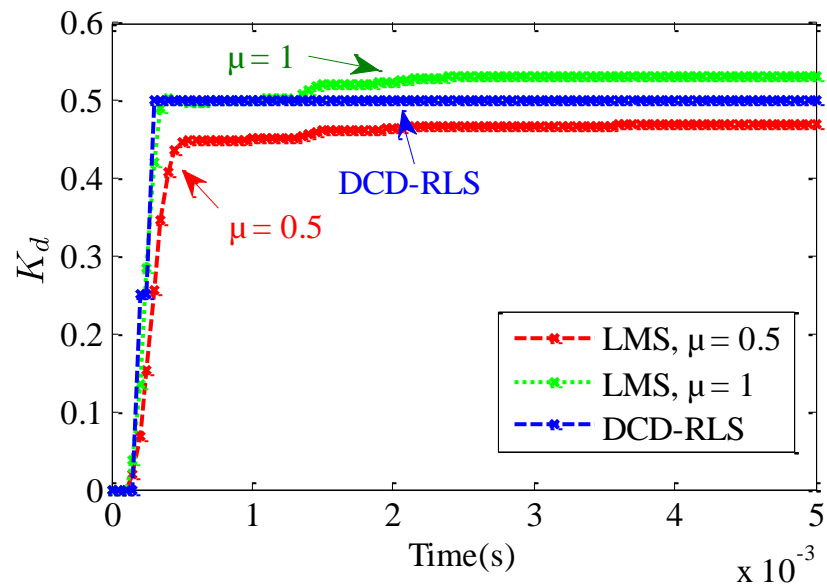


Fig. 5.10 Reference voltage feed-forward: Comparison of transient response between LMS and DCD-RLS. Repetitive load change between 0.66 A and 1.32 A every 5 ms

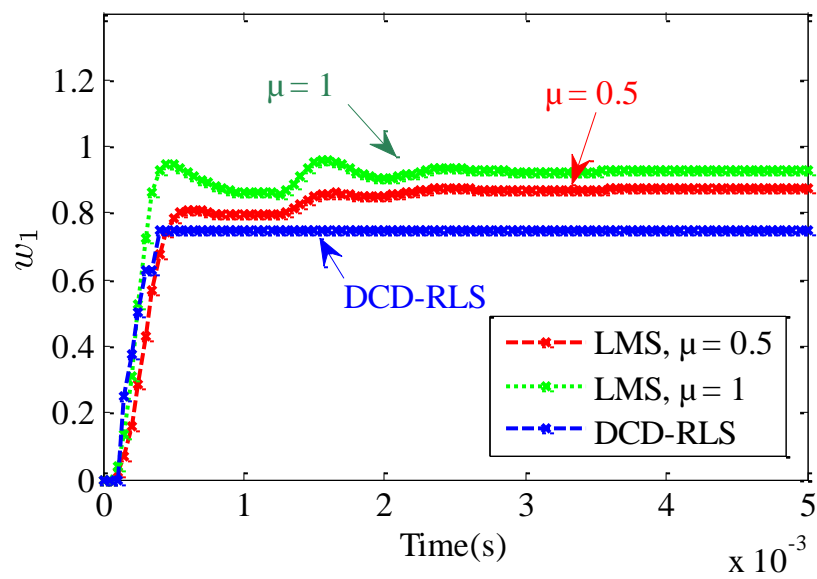
Both methods are able to maintain voltage regulation and recover from abrupt system changes. However, it is clear from Fig. 5.10 that the dynamic characteristics using the proposed DCD-RLS are better than the conventional LMS. There is smaller overshoot and a distinctly faster recovery time after a parametric change or when there is an increase in excitation. From this, we can deduce that the DCD-RLS method yields an overall improvement in the transient response of the system. Clearly, the tracking ability for the abrupt parameters changes is better in DCD-RLS than LMS.

As mentioned earlier, in the LMS algorithm the step-size may give rise to problems; one has to compromise between fast convergence rate and estimation accuracy. It is also compulsory to ensure that μ is within a range that guarantees the filter tap-weights will approach their optimal value. The adaptive gain (tap weight) of the LMS predictor filter, the convergence time, the tap-weight gradient noise, and the stability of the adaptation, all depend heavily on μ . Large values of μ decrease convergence time and improve the dynamic response as shown in Fig. 5.10 but increase the filter gradient noise and vice versa for low values of μ [80, 96]. For this specific example we found that the optimal step size value is when $\mu = 1$.

Fig. 5.11 shows the adaptation performance of the LMS and DCD-RLS algorithms. In both methods, the tap weights approach approximately the same values. However, the DCD-RLS is superior in terms of convergence time. As a result, the choice of step size is important for dealing with unexpected system disturbances. For example, in SMPC applications, one might observe a high control error signal, due to a high initial transient or an abrupt change in load current; if the step size is large, instability may arise. This is because the update of the filter coefficient is directly proportional to the input signal as given in equation (5.24). Therefore a prior knowledge of the variation of the input signal is essential to select an appropriate step size, thus ensure stability and parameter convergence.



(a)

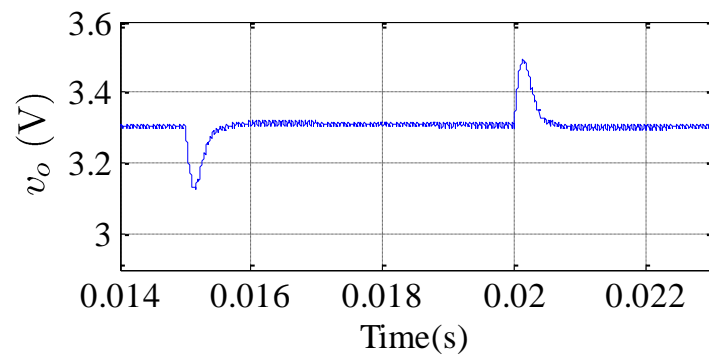


(b)

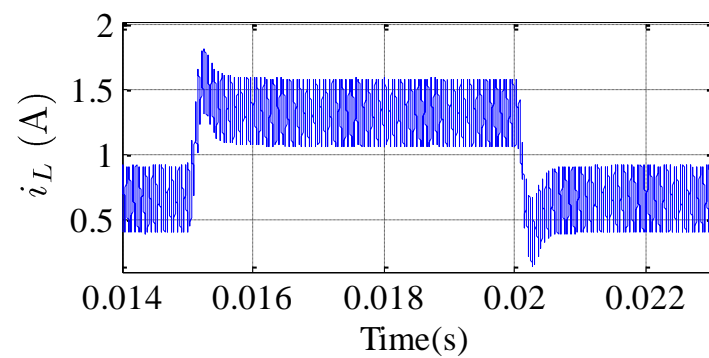
Fig. 5.11 Zoomed adaptation of gain (K_d) and tap-weight (w_1) in the two stage adaptive linear predictor for different step-size values

5.5.2 Voltage Control Using Adaptive PD+I Controller

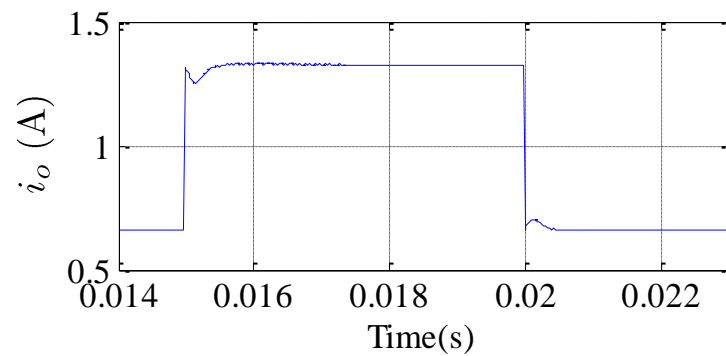
In this section, the adaptive PD+I controller initially discussed in section 5.2 is implemented. Fig. 5.12 shows the performance of placing the integral compensator (Fig. 5.1) in the feedback loop. This increases the excitation of the adaptive filter and drive the steady-state error to zero, hence improving the identification accuracy of the adaptive filter. To investigate the robustness of the algorithm to system disturbances, a load change is introduced into the system. This load change forces the load current to switch between 0.66 A and 1.32 A every 5 ms (Fig. 5.12). Usually the performance of adaptive methods and self-tuning controllers is measured using particular metrics. A cost function is one metric that can be used to describe the performance of a PEF. The benefit of using a PEF is that a cost function naturally exists. The optimum cost function for a PEF is actually the minimisation of the prediction error signal power required to reduce the loop error to zero Fig. 5.13(a). It is clear from Fig. 5.13(b) that the algorithm is capable of minimising the prediction error power; thereby, a well regulated output voltage is ensured. However, the main role of the PEF is to continuously work alongside the adaptive algorithm to minimise the prediction error. This in turn improves the prediction and identification of the input filter. The conventional LMS method can be applied with the adaptive PD+I structure to provide enhanced performance over the previous reference voltage feed forward method (Fig. 5.14). With the introduction of the integrator into the control loop, the loop excitation is increased, and this helps the identification process. However, as mentioned earlier, careful attention must be given to the selection of the step size μ . Fig. 5.14 also shows the equivalent performance of the PD+I structure using the DCD-RLS technique. Once again, it is clear that the DCD-RLS approach provides superior performance than LMS method.



(a)



(b)



(c)

Fig. 5.12 Transient response of the proposed adaptive controller, a: output voltage, b: inductor current, c: load current change between 0.66A and 1.32 A every 5 ms

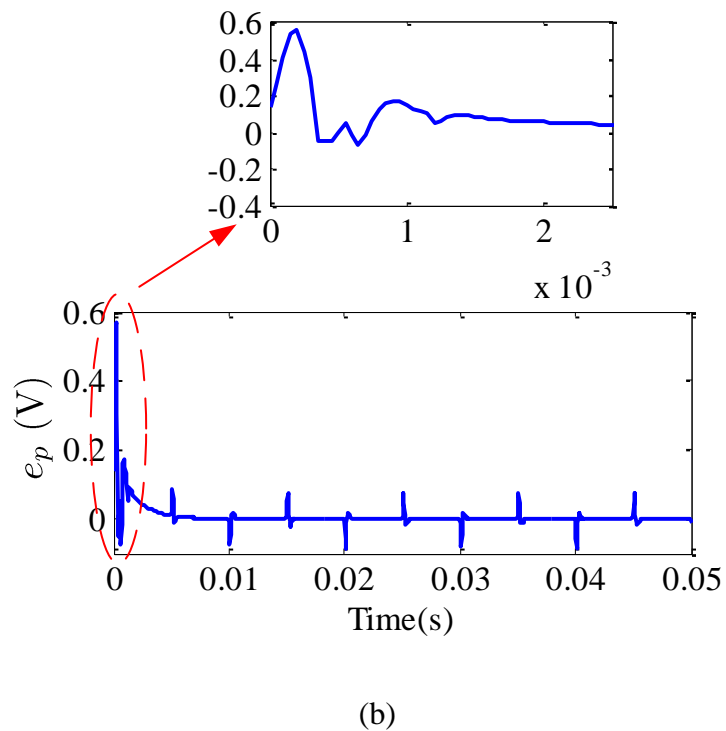
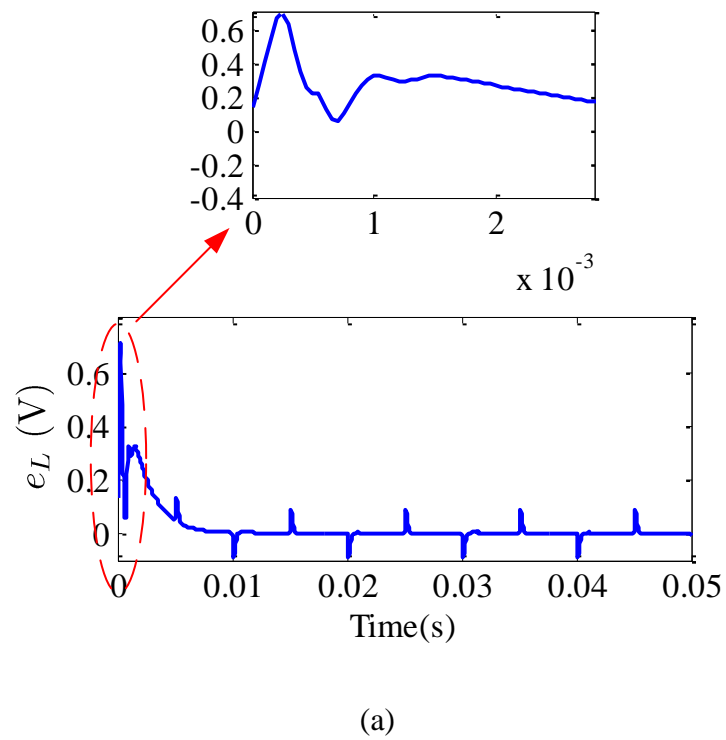


Fig. 5.13 Error signal behaviour during adaptation process, a: loop error (e_L), b: prediction error (e_{p1}). Load current change between 0.66 A and 1.32 A every 5 ms

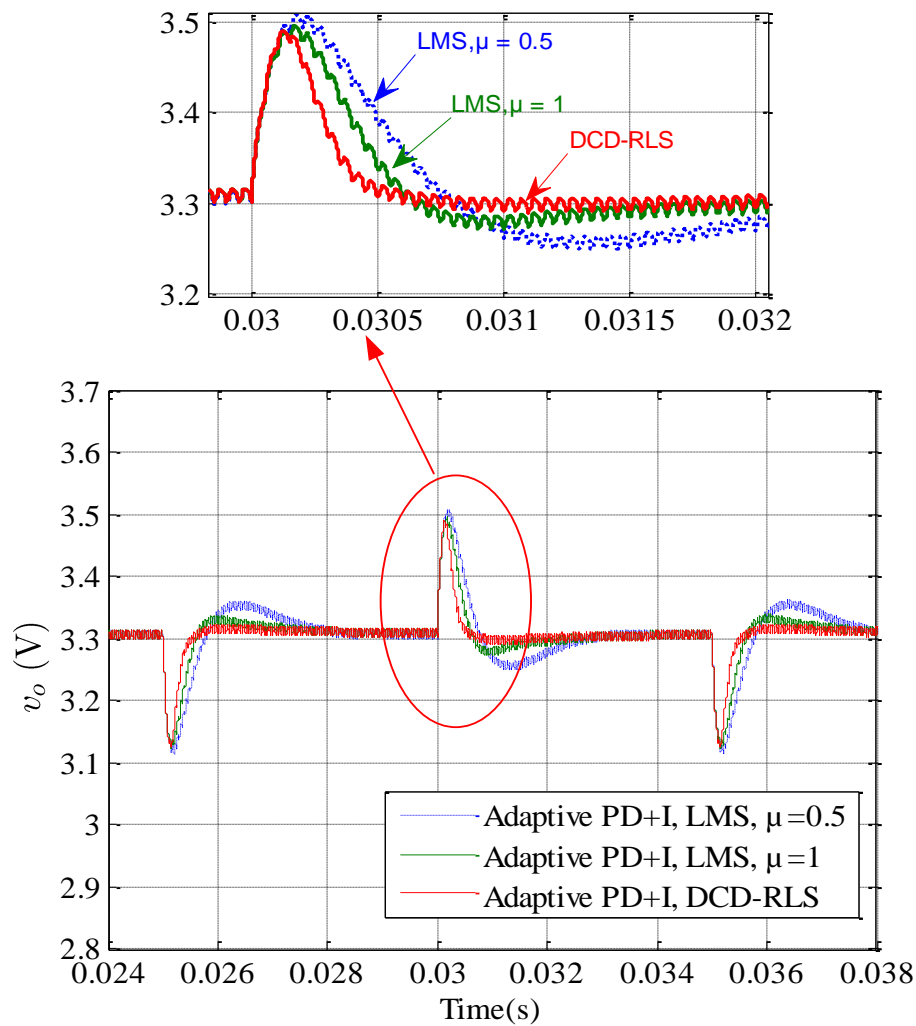
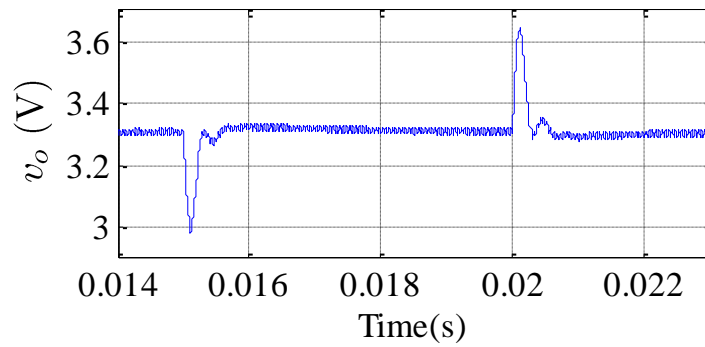


Fig. 5.14 Transient response of the proposed adaptive PD+I controller using DCD-RLS or LMS. Load current change between 0.66 A and 1.32 A every 5 ms

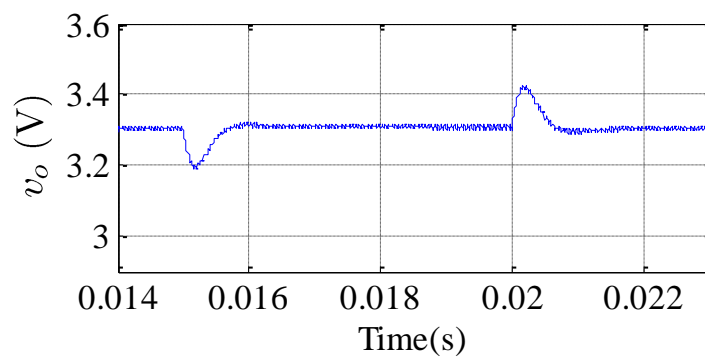
Furthermore, the versatility of the proposed PD+I adaptive controller has been tested with other converter circuit parameters to represent alternative dc-dc converter designs. It has been evaluated by changing the output capacitance with lower and higher values from the original design. To study the dynamic behaviour of the system during these changes, a periodic load change is introduced, Fig. 5.15(a, b). The same procedure then followed with respect to changing the output inductor to a lower value. Fig. 5.15(c) shows the dynamic performance during this change. In each case,

the proposed adaptive controller presents very promising results and can handle a wide range of uncertainty in the SMPC parameters.

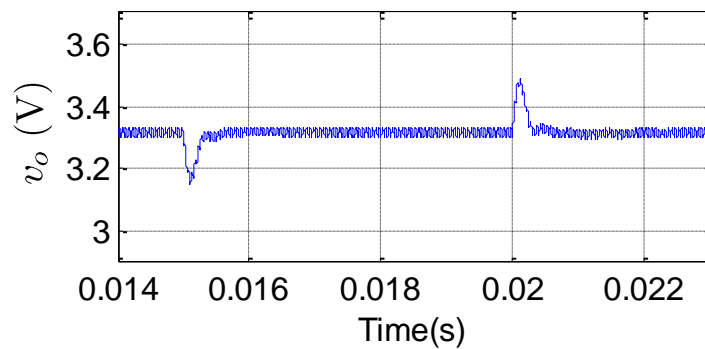
Finally, the adaptive PD+I controller is compared with a conventional PID controller optimally design using the pole-zero cancellation techniques previously presented in section 2.8.1. The adaptive PD+I scheme yields significantly improved transient performance for the same dynamic load change. It demonstrates significantly less oscillatory behaviour and faster recovery time.



(a)



(b)



(c)

Fig. 5.15 Transient response of the proposed adaptive controller during load current change between 0.66 A and 1.32 A every 5 ms, a: output capacitance $C = 150 \mu\text{F}$ and $L = 220 \mu\text{H}$, b: $C = 660 \mu\text{F}$ and $L = 220 \mu\text{H}$, c: output inductor $L = 100 \mu\text{H}$ and $C = 330 \mu\text{F}$

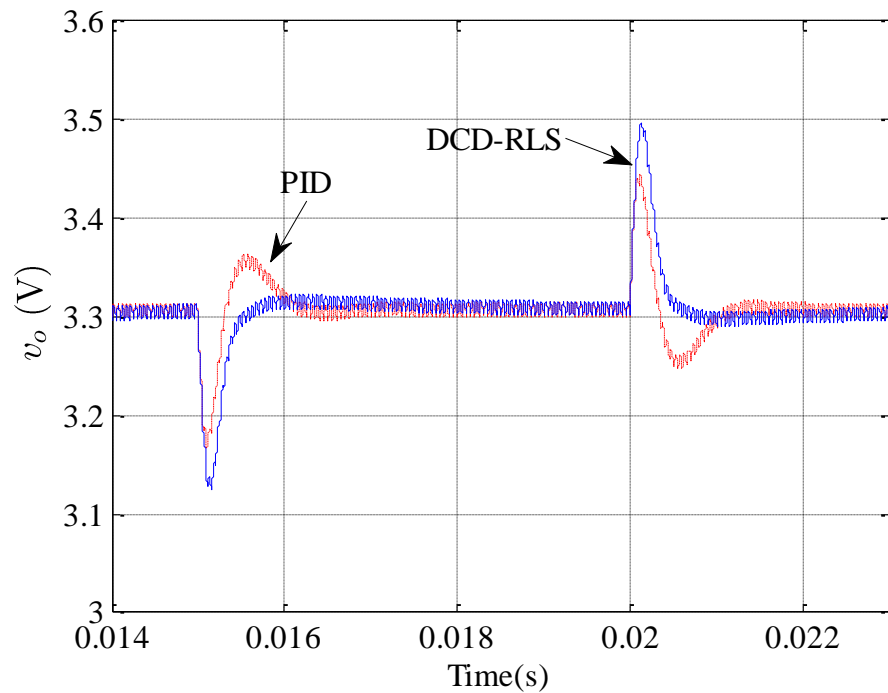


Fig. 5.16 Comparison of transient response results between the proposed adaptive PD+I using DCD-RLS and pole-zero PID control. Repetitive load current change between 0.66 A and 1.32 A every 5 ms

5.6 Robustness and Stability Analysis for the Proposed Adaptive PD+I Controller

SMPC controller behaviour and stability is often expressed in terms of frequency response criteria. The frequency response of the proposed adaptive controller is displayed in Fig. 5.17. Here, it is shown that the phase margin of the compensation system is increased through the introduction of the PD compensator in the loop. The phase margin of the adaptive PD+I compensator is 43° , and the gain margin is 17.8 dB.

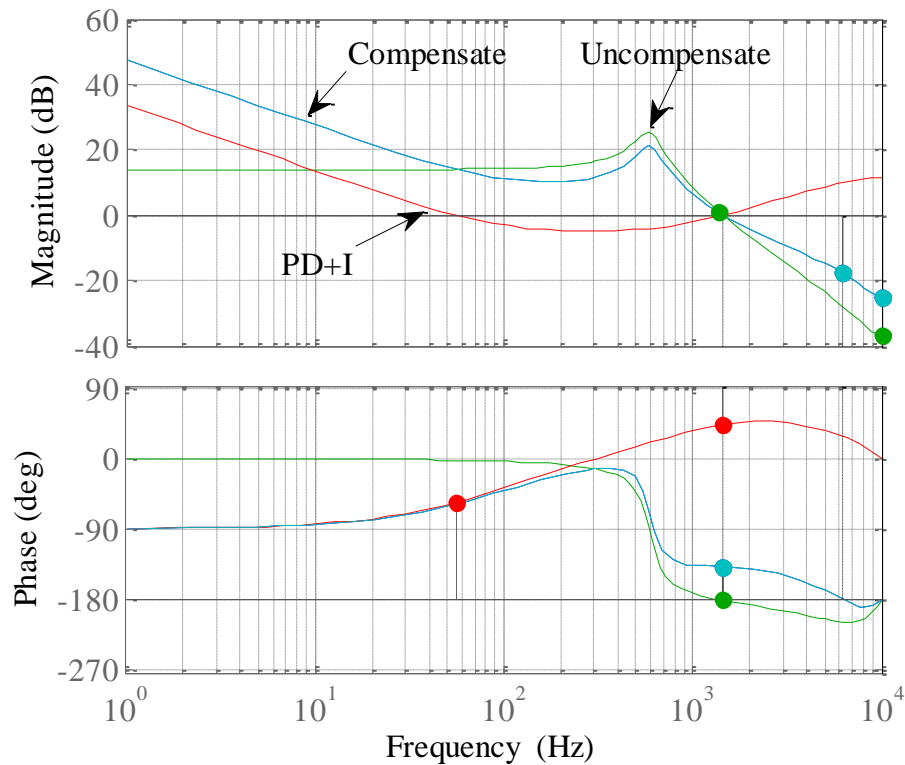


Fig. 5.17 Frequency response of the PD + I compensator and the compensated / uncompensated open loop gains

As shown in Fig. 5.18, three types of disturbance should be considered in closed loop digitally control dc-dc SMPCs [59]; measurement noise, $v_i(n)$, control noise $v_u(n)$, and load disturbance $v_l(n)$. To assess the individual impact of each of these disturbances, and measure noise rejection capability, sensitivity analysis can be used. This analysis can also be used to measure system dynamics and determine the effects of parameter changes in the system. By considering Fig. 5.18 and using sensitivity analysis the overall effect of the disturbances on the SMPC can be expressed in terms of a series of sensitivity functions [59, 72]:

$$y = GV_{ref} + S_{yl}v_l + S_{yi}v_i + S_{yu}v_u \quad (5.25)$$

where:

$$G = \frac{G_c G_{dv}}{1 + G_c G_{dv}} = \frac{G_{Lo}}{1 + G_{Lo}} \quad (5.26)$$

$$S_{yl} = \frac{1}{1 + G_c G_{dv}} = \frac{1}{1 + G_{Lo}} \quad (5.27)$$

$$S_{yi} = \frac{-G_c G_{dv}}{1 + G_c G_{dv}} = \frac{-G_{Lo}}{1 + G_{Lo}} \quad (5.28)$$

$$S_{yu} = \frac{G_{dv}}{1 + G_c G_{dv}} = \frac{G_{dv}}{1 + G_{Lo}} \quad (5.29)$$

Here, G and G_{Lo} are the closed and open loop transfer function respectively. S_{yl} , S_{yi} , and S_{yu} are the output, input, and control sensitivity functions respectively.

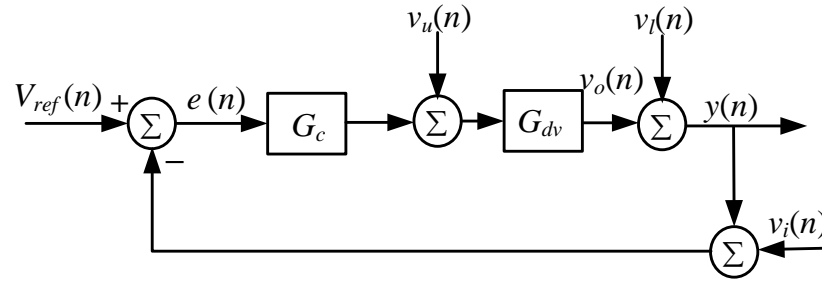


Fig. 5.18 Closed loop scheme of voltage mode control for SMPC

In Fig. 5.18, S_{yl} describes the system performance from a disturbance rejection point of view, S_{yi} highlights the effect of input noise upon the SMPC model, and S_{yu} signifies of control disturbance rejection of the plant [72]. Fig. 5.19 depicts the corresponding sensitivity function of the proposed adaptive controller. It can be observed that the maximum value of S_{yl} is about 2.8 dB. From this, the modulus margin can be determined.

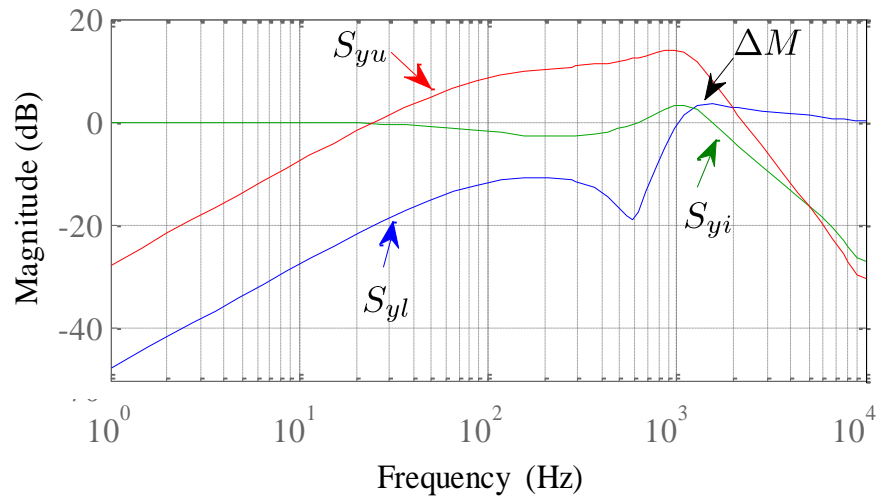


Fig. 5.19 Sensitivity functions of the PD+I controller

Modulus margin is defined as the radius of the circle centred at $(-1, j0)$ on the Nyquist plane required to touch the closest tangent to the plot of the open loop transfer function (G_{Lo}). This is demonstrated in Fig. 5.20. The connection of the critical point to the Nyquist plot of (G_{Lo}) is given by [59, 72]:

$$\min \Delta M = \left| 1 + G_{Lo}(e^{-j\omega}) \right| = \min \left| S_{yl}(e^{-j\omega})^{-1} \right| = \frac{1}{\max \left| S_{yl}(e^{-j\omega}) \right|} \quad (5.30)$$

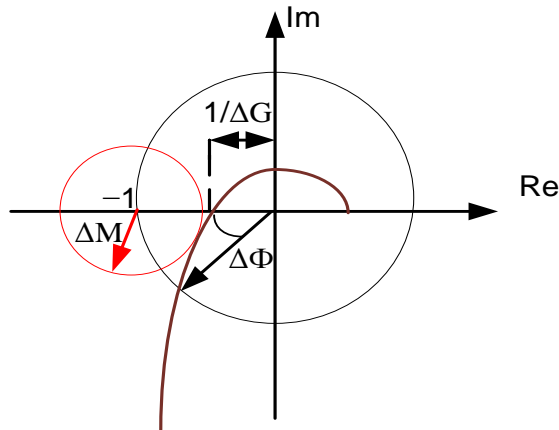


Fig. 5.20 Margins on Nyquist plot

From equation (5.30) it can be concluded that the modulus margin is inversely proportional to the maximum magnitude value of the S_{yl} function. According to [59, 72], ΔM should be kept higher than 0.5 to ensure system robustness, which implies

that the maximum value of $|S_{y_l}|$ should actually be less than 2. Therefore, the lower the maximum value of S_{y_l} the better the output disturbance rejection will be. From Fig. 5.19, it is found that ΔM is approximately equal to 0.72 in this particular system. ΔM is sometimes considered to be an alternative measure of system stability with or instead of gain/phase margin [28].

5.7 Chapter Summary

This chapter presented the viability of incorporating the adaptive PEF as a main controller in the feedback loop. It has demonstrated the mathematical relationship between the AR process and PEF filter. In addition, it described the relation between the PEF and a PD controller, and its suitability for use in an adaptive control design. In consequence, this chapter has demonstrated the feasibility of a new adaptive PD+I controller based PEF for the output voltage regulation of a dc-dc converter. The adaptive control system uses a two-stage/one-tap FIR filter and integral controller. A computationally efficient DCD-RLS algorithm has been used to implement the adaptation mechanism and to overcome many of the limitations of conventional RLS methods, making it well suited for real time power electronic applications. Furthermore, this chapter provided details on the LMS adaptive algorithm. The performance of the proposed adaptive PD+I controller using DCD-RLS was compared with the LMS method. It showed that the adaptive PD+I controller relying on DCD-RLS provided superior performance than the LMS one. The proposed controller has the ability to work continuously in the feedback loop and rapidly minimize the controller error signal by finding real-time tap weights for the FIR filter. The integral controller amplified the oscillation in the feedback loop for a very short period of time to increase the excitation for prediction and identification purposes. The adaptive filter parameters quickly converge and eliminate this oscillation. In this way, the approach is suitable for two important purposes: prediction/identification and controller adaptation. Finally, the robustness/stability analysis of the proposed predictive controller has also been explained in this chapter.

Chapter 6

MICROPROCESSOR APPLICATION BASED SYNCHRONOUS DC-DC SWITCH MODE POWER CONVERTER- EXPERIMENTAL RESULTS

6.1 Introduction

With the advent of increasingly powerful, and cost effective, microprocessor platforms, advanced signal processing algorithms and intelligent adaptive controllers can now readily be implemented on microprocessor based systems to significantly improve the overall dynamic performance of the process. To fully validate the proposed schemes developed in this thesis, a microprocessor based experimental synchronous dc-dc buck converter has been designed and tested for 5 W operation. This chapter describes the laboratory prototype hardware in detail and presents research results validating the novel system identification method using the leading DCD-RLS algorithm presented in Chapter 4 and the digital adaptive control structure described in Chapter 5. Texas Instruments™ TMS320F28335™ *eZdsp* DSP platform has been used in the experimental validation.

6.2 Microprocessor Control Platform

A digital signal processor (DSP) is a dedicated type of microprocessor that is programmed by the user for optimal system operation. The DSP architecture is optimally designed for fast and effective operation of digital signal processing algorithms. The TMS320F28335-DSP platform (Fig. 6.1) is used in this research for parameter estimation and for digitally control of the dc-dc SMPC converter. The TMS320F28335 microprocessor is a member of the Delfino™ C2000 DSP platform

from Texas Instruments (TI) [128]. This chip is a floating point processor which is optimised for digital control applications. It enables high performance computationally advanced algorithms to be implemented using simple system programming. According to [129] from TI, the TMS320F28335 core offers a 50 % performance enhancement over similar fixed point platforms.

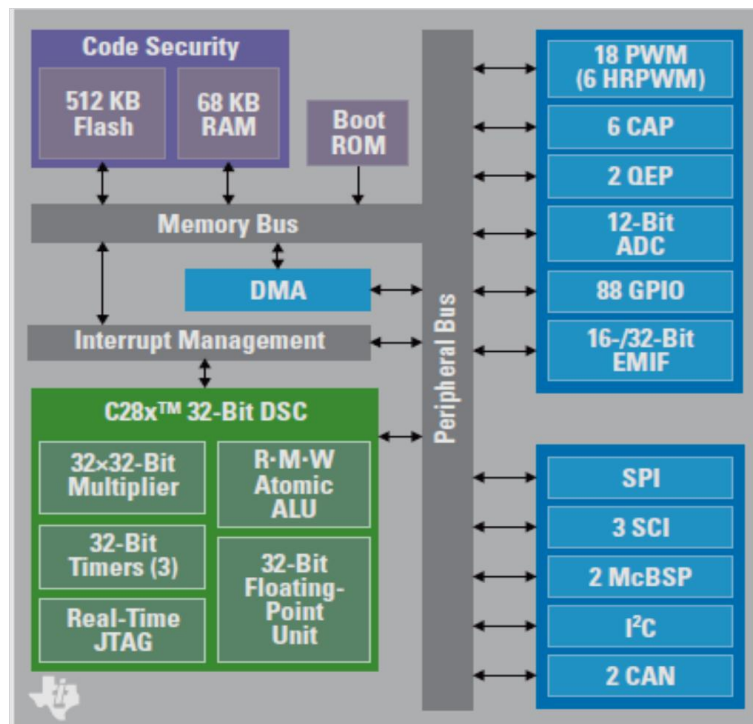


Fig. 6.1 TMS320F28335 *eZdsp* Architecture [129]

The TMS320F28335 based on Harvard architecture design (Fig. 6.1) is similar to the other general purpose microprocessors [128]. This platform includes 512 KB flash memory, 68 KB RAM, and 6 channels direct access memory (DMA). As shown in Fig. 6.1, the processor core consists of three main parts: 1) Arithmetic Logic Unit (ALU), 32×32-bit multiplier, and 2) 32-bit Floating Point Units (FPU). In addition, the TMS320F28335 processor is fully mixed signal core that consists of [128]:

- 12-bit / 16 channel ADC core with conversion time 80 ns at speed up to 12.5 Mega Samples per Second (MSPS). Two built-in analogue multiplexers are integrated with the ADCs to enable connection of 8 channels per multiplexer with dual built-in sampled and hold circuits (S/H). The read operation from the ADC channel can performs simultaneous or sequential conversion from

each multiplexer. The converted values are stored into its dedicated 16-bit results registers. The conversion operation can be started by a trigger signal generated by an event manager or by an external trigger signal through the general purpose input/output (GPIO). Two events (EVA, EVB) are used to trigger the ADCs, these events can work independently.

- The TMS320F28335-DSP has dual 6 channel/16-bit enhanced PWM. Each channel can be independently programmed to generate symmetric and asymmetric PWM. Each event manager module has a 16-bit general purpose timer. The PWM compare registers are used to compare the associated control signal with the timer registers. The timers can be programmed as up/down counters to emulate the PWM operation. The TMS320F28335 processor also has 6 channels/32-bit enhanced capture input (eCAP) that can be configured to generate 6 PWM channels.
- Several communication interface circuits are also integrated into the TMS320F28335 including: Enhanced Controller Area Network (eCAN), Serial Peripheral Interface (SPI), and Serial Communications Interface (SCI).

6.2.1 Microprocessor Code Development

In order to implement and evaluate the proposed system identification and adaptive control algorithms using the TMS320F28335 *eZdsp*; Texas Instruments Code Composer Studio (CCS) based Integrated Development Environment (IDE) is employed on the host PC to write C language programming code and to compile the developed code for download onto the target DSP. In addition, Simulink Embedded Target Support Package (TSP) and Real-Time Workshop (RTW) toolboxes are available for rapid prototyping of the developed adaptive algorithms, automatic C-code generation from Simulink models and for setting the input/output device peripherals (e.g. PWM) as specified by the hardware blocks in the real-time model. This provides a simple, fast, and alternative way to implement and rapidly validate the proposed algorithms in real time using MATLAB/Simulink [130]. After compiling the code, the CCS builds the process and downloads the executable files

onto the DSP core for real time operation. The CCS provides a flexible interface to test, edit, and read the generated code.

6.3 System Hardware Description and Microprocessor Setup

The test platform of the digitally controlled buck dc-dc converter consists of four main parts: single-phase synchronous dc-dc buck converter with dynamic load change circuit, gate drive circuits, signal conditioning/measurement circuits, and the microprocessor core (TMS320F28335 *eZdsp*). Fig. 6.2 shows the whole system setup used in this project and Fig. 6.3 presents the corresponding block diagram of this setup (see appendix B for the circuit schematic).

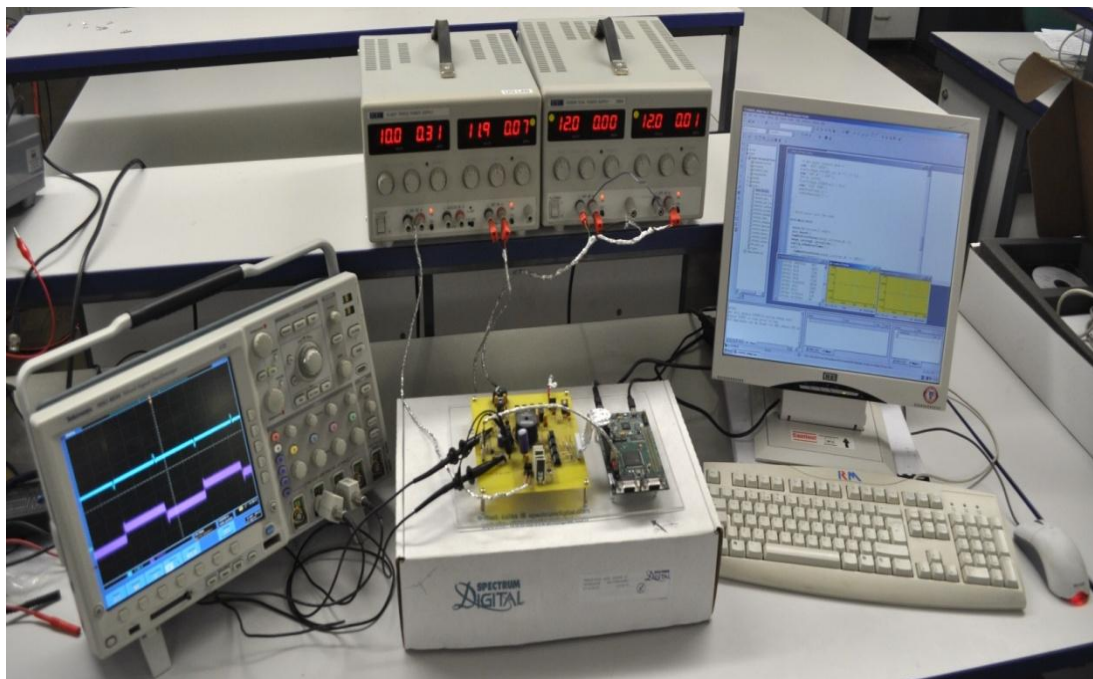


Fig. 6.2 Hardware platform setup

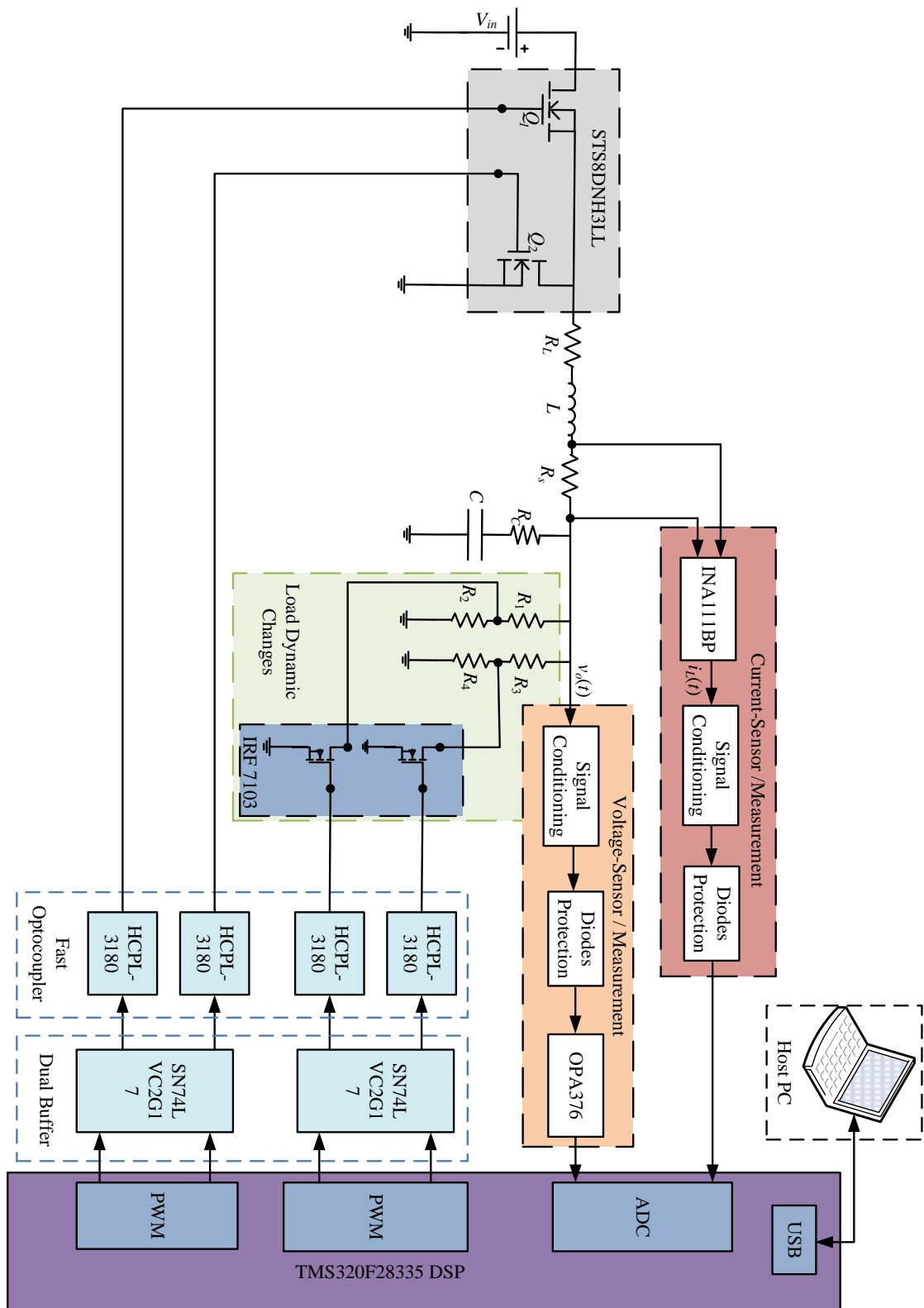


Fig. 6.3 Block diagram of the synchronous dc-dc buck converter based on microprocessor

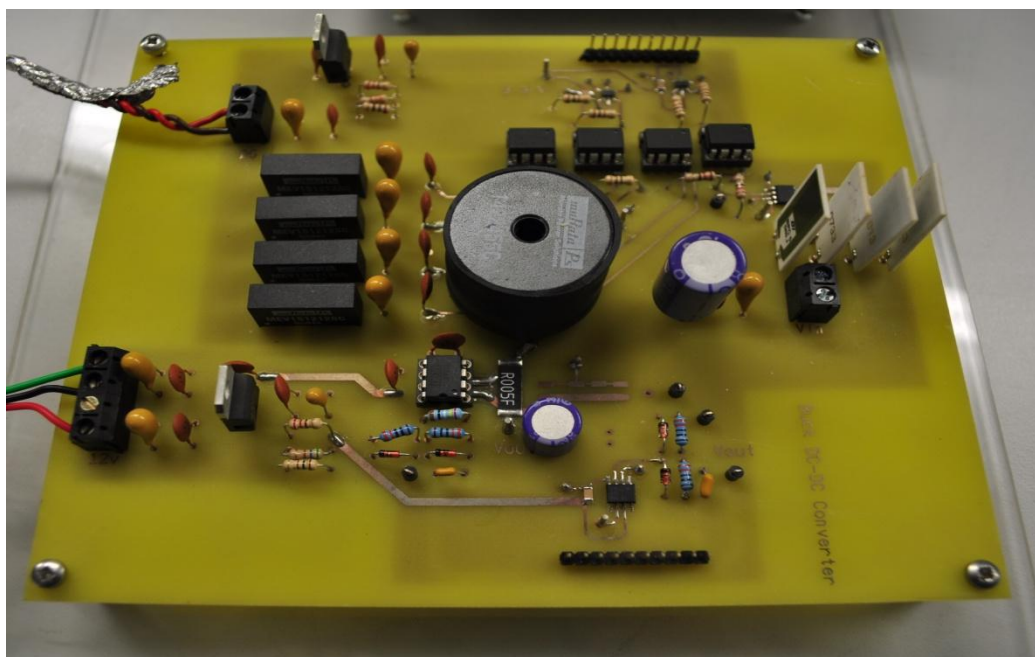
The synchronous dc-dc buck converter includes: two N-channels MOSFETs circuit (ST8DNH3LL) as a switched device, DC-input voltage source with decoupling capacitors, power stage filter (L and C) and output load resistor. In addition, a dynamic load change circuit is designed to test the adaptive controller performance during load changes. As illustrated in Fig. 6.3, two parallel load branches are connected with the output of the buck converter. Each branch includes two series resistors of $5\ \Omega$. In the normal operation, the equivalent load resistance is equal to $5\ \Omega$ ($10\ \Omega // 10\ \Omega$). Therefore, at $3.3\ \text{V}$ regulated output voltage the load current is equal to $I_{out} = 0.66\ \text{A}$. In order to change the load dynamic, a switching circuit (Power MOSFET IRF7103PbF) is included (Fig. 6.3). By closing the switches in each load line, the load resistance seen by the power converter can be reduced, thus increasing the overall load current. In case one, both switches are closed and the total load resistance is reduced to $2.5\ \Omega$ ($5\ \Omega // 5\ \Omega$, $I_{out} = 1.32\ \text{A}$). In case two, one of the switched is closed whilst the other is open and the load is cut to $3.3\ \Omega$ ($5\ \Omega // 10\ \Omega$, $I_{out} = 1\ \text{A}$). The parameters of the prototype synchronous buck converter are shown in Table 6.1. Fig. 6.4(a, b) depicts the prototyped synchronous dc-dc buck converter circuit and the selected digital signal processor platform respectively.

Table 6.1 Prototyped synchronous buck converter parameters

Symbols	Parameters description	Values
V_{in}	Input voltage	10 V
V_o	Output voltage	3.3 V
I_{omax}	Maximum output current	1.32 A
L	Inductor	220 μH
C	Capacitor	330 μF
R_L	Inductor ESR	0.063 Ω
R_C	Capacitor ESR	25 m Ω
R_o	Load resistors	5 Ω , 5 W
R_s	BCS 8, T _T Electronics	8w, 50 m Ω



(a)

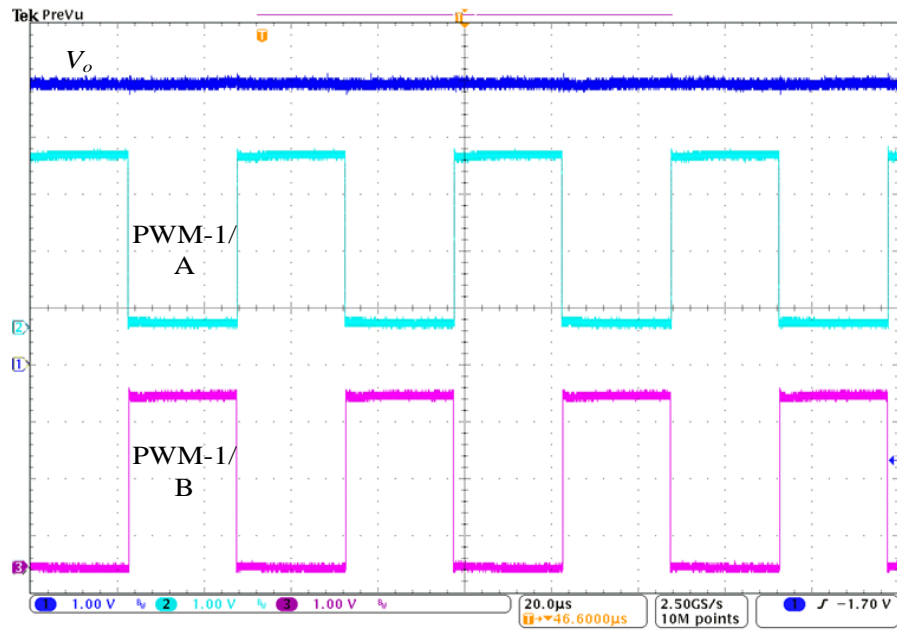


(b)

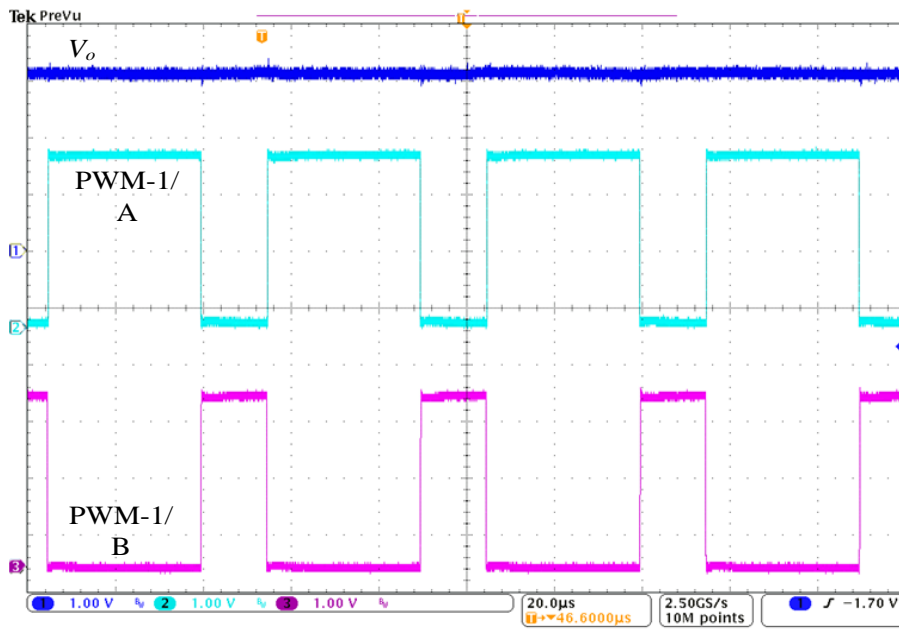
Fig. 6.4 a: TMS320F28335™ DSP platform, b: the synchronous dc-dc buck converter circuit

As presented in Fig. 6.3, two signal conditioning and measurement circuits are designed to measure the regulated output voltage and inductor current. The output voltage generated from the dc-dc power processor is initially scaled down via resistive voltage divider circuit with gain factor equal to 0.5 to accommodate the full dynamic scale of the ADCs which is 3 V. To be confident that the measured voltage does not exceed the ADC full scale, a protection Schottky-Diodes (BAT 85) is included in the measurement circuit. In addition, a buffer protection circuit using a unity gain fast operational amplifier (OPA376) is inserted into the measurement circuit before the ADC chip. A similar signal conditioning and protection circuit is used for the inductor current measurement. In order to measure the current signal, a series shunt resistor is used (Fig. 6.3) with a high speed instrument amplifier (IN111BP). Compare to using a hall effect transducer, this approach reduces the cost and space of the printed circuit board.

Within the microprocessor itself the built-in ADCs sample the input signals and the sampled data is then processed by the software control algorithm. After the control algorithm is executed the duty-cycle signals are updated and a new PWMs signals will be generated. The generated PWM signals are then passed through a dual buffer circuit (SN74LVC2G17) to protect the PWM channels (Fig. 6.3). This buffer circuits are carefully selected to produce a match output levels to the DSP-PWM output voltage. From this, the buffered PWM signals get passed to isolated gate drives (HCPL-3180). Another two PWM channels are configured to activate the dynamic load circuit. Here, the load is configured to repetitively change every 25 ms. Fig. 6.5(a) shows the experimental open loop results of the buck dc-dc converter circuit. The waveforms show the steady-state output voltage (V_o) and the corresponding PWM signals for both N-channels MOSFETs. In this instant the complementary PWM signals have the same duty cycle (50 % duty ratio). Similar results are presented in Fig. 6.5(b) with a different voltage regulation level (33 % duty ratio).



(a)



(b)

Fig. 6.5 PWM waveforms in open loop circuit test, a: duty ratio 50 % , b: duty ratio 33 %

6.4 System Identification Using DCD-RLS / Experimental Validation

This section demonstrates the practical validation of the proposed system identification algorithm presented in Chapter 4. The adaptive leading DCD-RLS algorithm (system identification scheme) and the adaptive PD+I architecture are programmed based on the flowchart shown in Fig. 6.6.

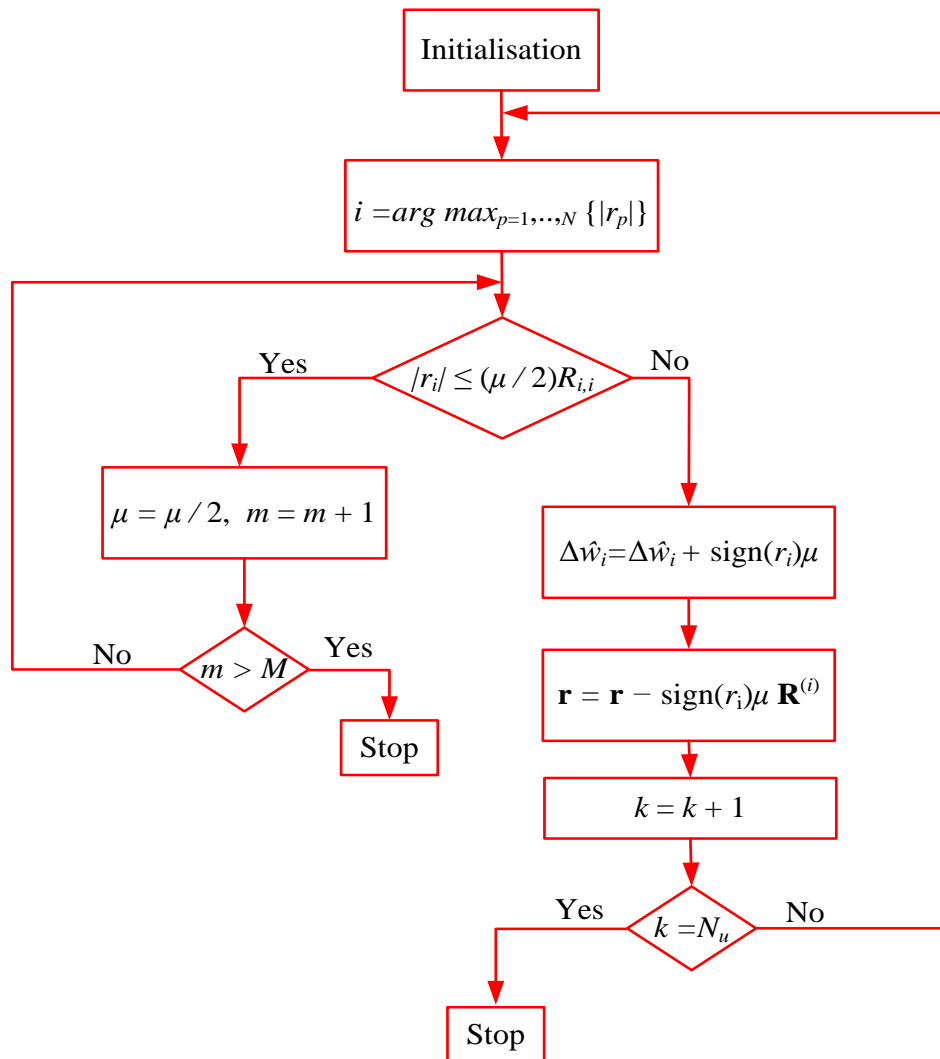


Fig. 6.6 Leading DCD-RLS algorithm flowchart

The designed synchronous dc-dc buck converter has been used to generate real time practical data for direct input into the DCD-RLS algorithm. For easy comparison with the original simulation results, similar parameters and component values to those outlined in section 4.11 are chosen as shown in Table 6.1. The TMS320F28335 platform is used to implement the digital PID controller, to inject the digital PRBS

and then to collect the input/output measurement data. A 9-bit PRBS is generated and implemented in the DSP (first shown in Fig. 4.4). The PRBS amplitude, $\Delta_{PRBS} = \pm 0.008$, and the total data length is 511. Therefore, a complete PRBS sequence is $L / f_s = 25$ ms. The PID gains used in the experimental test are selected to match the simulation setting in section 4.11, where, $q_0 = 4.127$, $q_1 = -7.184$, and $q_2 = 3.182$.

During the practical work, the same procedure as presented in Fig. 4.11 is followed. Fig. 6.7 highlights the output voltage waveform of the experimental buck converter when the PRBS disturbance is injected to allow for system identification. Initially, the SMPC is working under normal conditions (system identification disabled). The system identification process is then enabled; the PRBS signal is injected into the loop and the system begins to estimate the unknown parameters of the buck converter model. The disturbance in the output voltage, created by the PRBS, is clearly visible in Fig. 6.7. The voltage ripple is approximately $\pm 3\%$ with respect to the nominal dc output voltage. However, it can also be seen that this disturbance only exists when the identification process is enabled. After 20 ms, the process is complete, and the buck converter reverts back to normal operation. The PRBS injection time is deliberately increased in this example test to fully demonstrate the convergence rate of the parameter estimation. The actual length of time of the excitation can be significantly reduced in the final optimized solution.

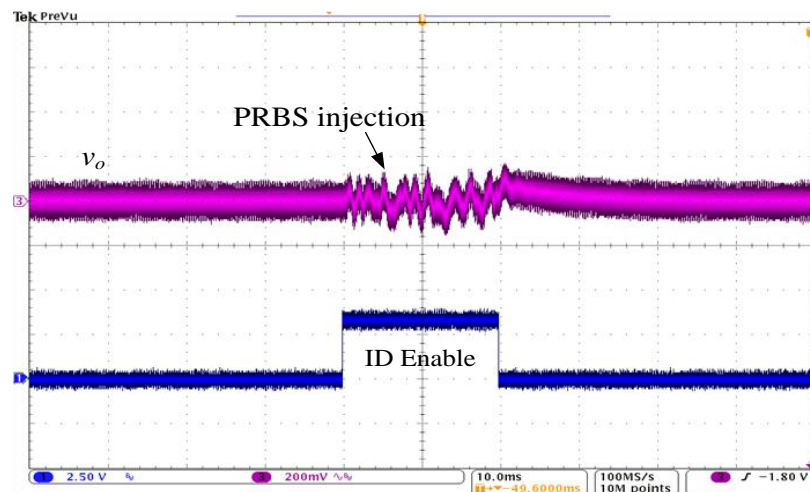


Fig. 6.7 Experimental output voltage waveform when identification enabled. (ac coupled)

Now, the measurement data from the dc-dc converter is stored in the DSP memory, and exported to MATLAB for post-processing after the full test sequence has been applied to the power converter. Practically, in order to focus the identification on the frequency range of interest and remove unwanted high frequency measurement noise; the inputs to the DCD-RLS algorithm require filtering prior to identification. Here, a four tap moving average FIR filter is designed to smooth the input and output data. In addition, offset in the input signals must be removed as the RLS algorithm assumes zero mean input values. In dc-dc SMPC applications it is easier to remove offsets on a cycle-by-cycle basis from the input signals, where steady-state average values of the regulated output voltage and the average duty-cycle ratio are known. At each time instance, the average value of the input signal is directly subtracted from the excited signal. A high-pass filter can also be used to remove the offset from the input signals; however, this will add more computation to the overall system that is not essential in the on-line system identification process. Fig. 6.8 shows the sampled output voltage and duty cycle data from the dc-dc converter during the identification process. From the measured data, the DCD-RLS performs the cycle-by-cycle parameter estimation algorithm previously described to identify the tap-weights of the IIR filter and minimise the prediction error signal. The experimental parameters of the DCD-RLS algorithm are chosen to match the initial buck converter simulation settings and allow for easy comparison of results.

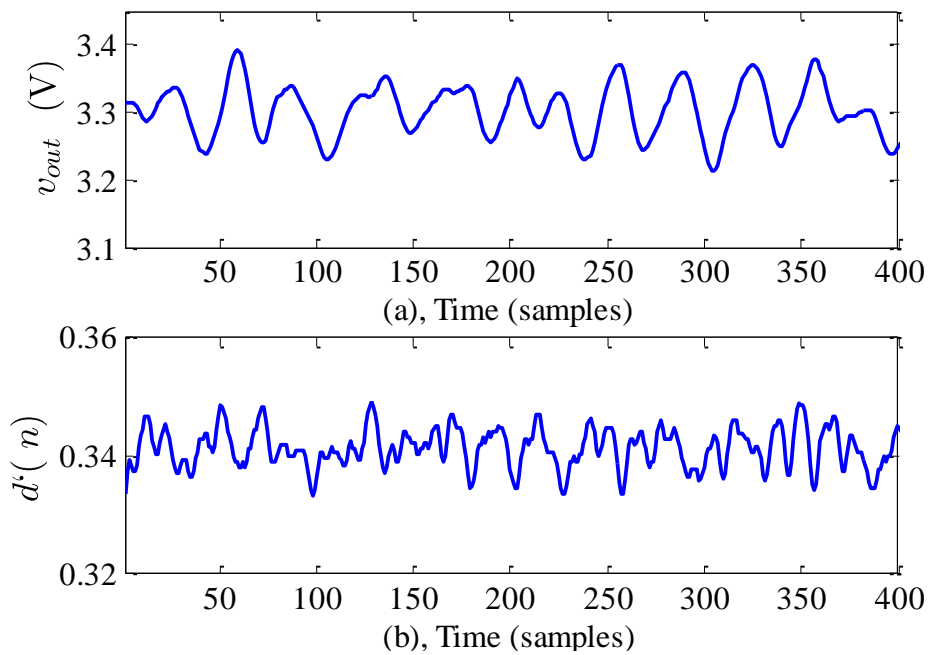


Fig. 6.8 Experimental output voltage and persistence excitation signal (duty signal + Δ_{PRBS}) results during ID, based on sampled data collected from DSP

The results from experimental measurement are shown in Fig. 6.9. Importantly, there is excellent agreement with the original simulation results in Fig. 4.12. The practical based results show both the classical RLS method and the DCD-RLS algorithm converge quickly (< 10 ms) to virtually the same parameter estimation values. Furthermore, it is apparent from Fig. 6.10 that the voltage prediction error signals for both algorithms (RLS and DCD-RLS) converge quickly to zero. In this way, both techniques successfully identify the discrete model of the SMPC from real time experimental data. However, as shown in earlier analysis, the computational effort of the DCD-RLS is substantially lower. It is worth noting that in both methods the convergence time of the pole coefficients (a_1, a_2) is faster and more accurate than the zero coefficients (b_1, b_2). This is re-assuring since in many control systems, including SMPCs, accurate knowledge of the pole locations is important for stability analysis and controller design. Fig. 6.11(a, b) presents the actual estimation error of the classical RLS and DCD-RLS respectively. This result clearly shows that both algorithms reach approximately zero estimation error with a rapid convergence rate. In summary, the performance of the DCD-RLS is comparable to the conventional RLS method.

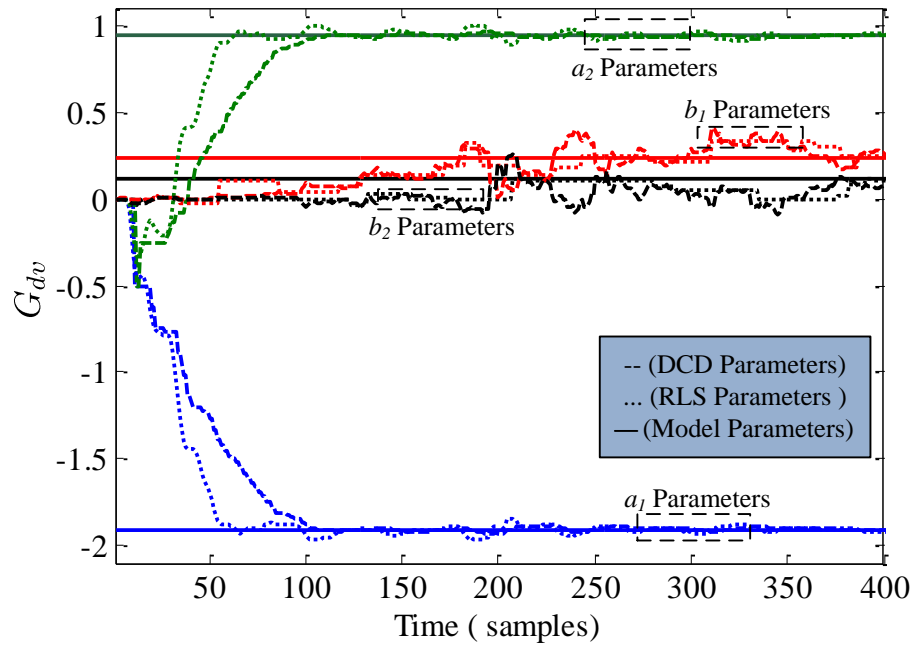


Fig. 6.9 Experimental tap-weights estimation for IIR filter with DCD-RLS and classical RLS methods; compared with the calculated model

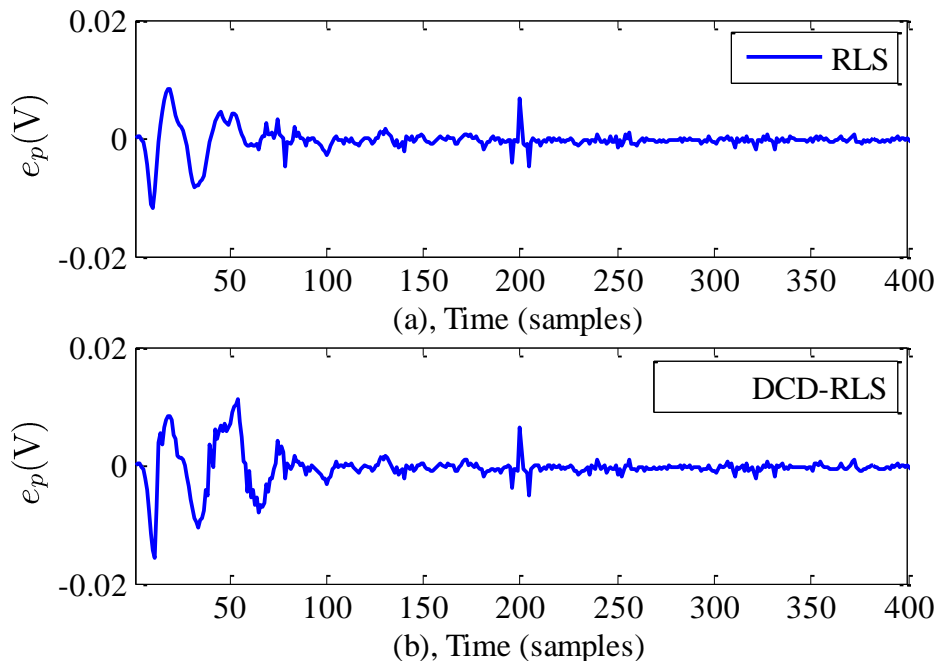
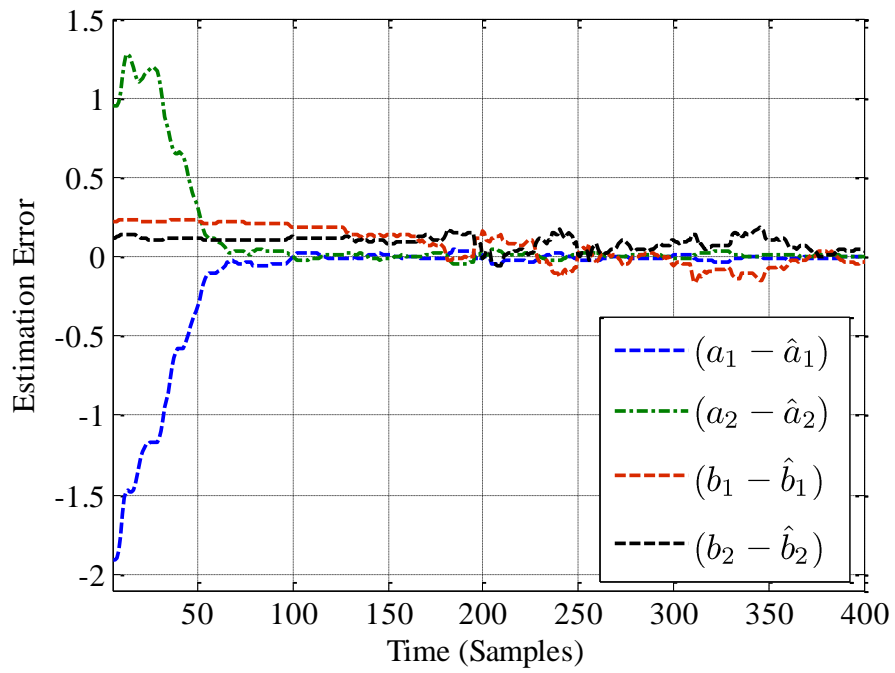
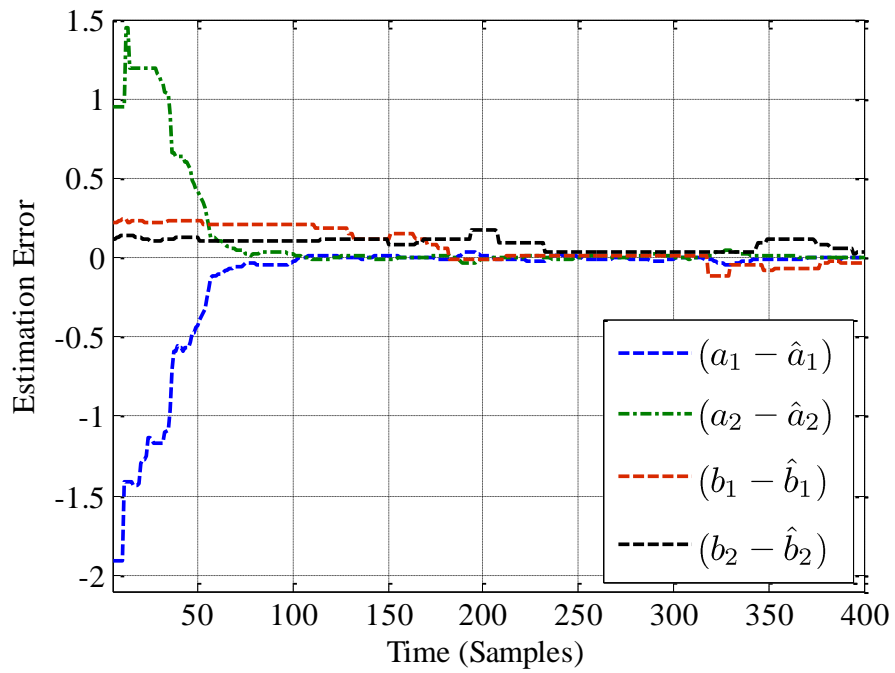


Fig. 6.10 Experimental prediction error results, a: conventional RLS, b: DCD-RLS



(a)



(b)

Fig. 6.11 Experimental parameters estimation error, a: classical RLS, b: DCD-RLS

It has already been noted that the estimation performance of the DCD-RLS algorithm can be improved by increasing the number of iterations-albeit, at the cost of increased computational complexity. Fig. 6.12 compares the mean square error (MSE) performance of the DCD-RLS algorithm with different iteration values (N_u); against the conventional RLS technique. It can be seen that the conventional RLS convergence rate and MSE magnitude are lower than the DCD-RLS, however the convergence rate of DCD-RLS is improved when the number of iterations is increased N_u . As in many applications, a compromise must be made between performance and complexity. In this particular case, $N_u = 1.0$ is sufficient for fast SMPC parameters estimation with acceptable estimation error.

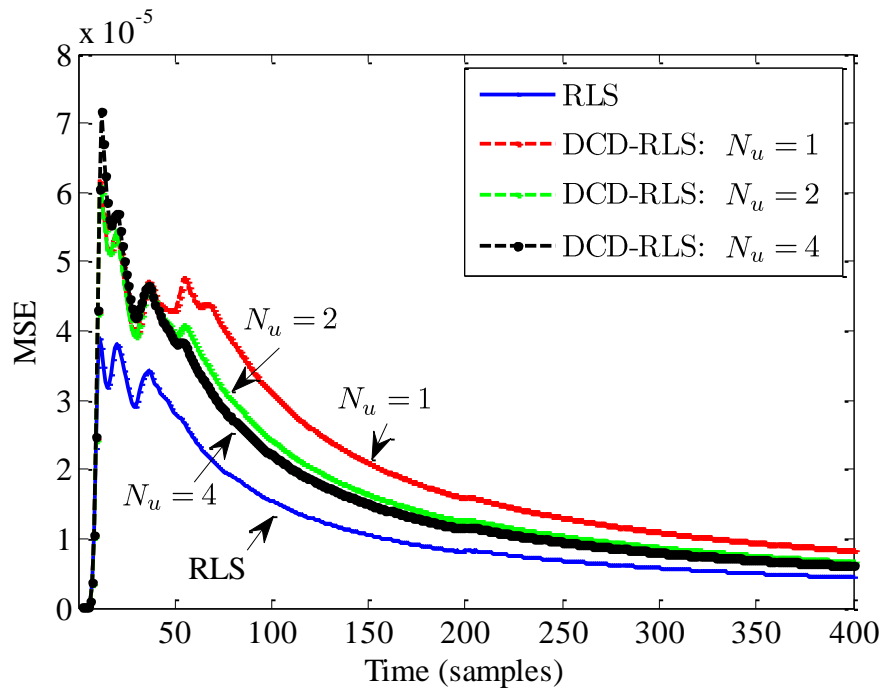


Fig. 6.12 Experimental learning curves comparison results of conventional RLS against DCD-RLS at different iteration values

6.5 Realisation of the Converter Model

In order to confirm the suitability of using a second order model for the dc-dc converter, experimental input and output sample data is collected from the buck converter. The mean value is then removed from the input and output data as shown in Fig. 6.13(a, b). Following this, the input and output data is divided into two parts. The first part is used to construct the system model and consists of 750 samples (37 ms) and the second part is used to validate the resulted model, (a further 750 samples). The real output data of the second part is compared with the estimated output data and when the differences between the measured data and the constructed model are small, the model can be considered as a good fit to the collected data.

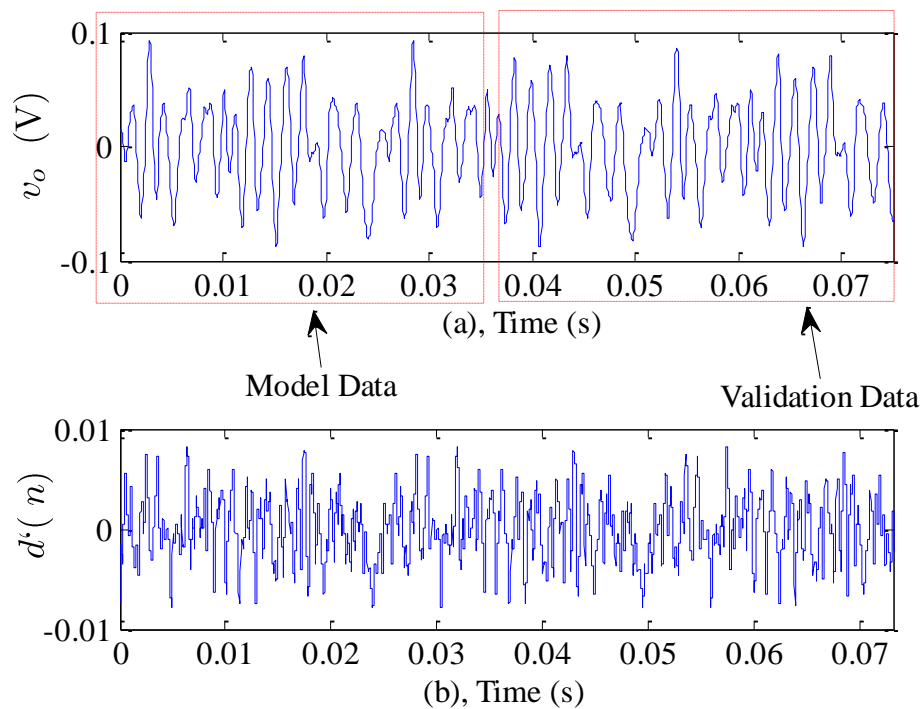


Fig. 6.13 Experimental sampled data collected from DSP, a: output voltage, b: control signal (duty signal + Δ_{PRBS})

Now, two types of the model structure are tested using this evaluation: 1) second and third order equation error model, 2) second and third order output error model. As shown in Fig. 6.14, equation error model provides a better fit than the output error model. Furthermore, increasing the order of the model does not provide any

significant difference in the system data fits: both models provide 98.77% fit with the output data (Fig. 6.14). This result confirms that a second order equation error model is a good choice of candidate model to estimate the system parameters of dc-dc buck converter.

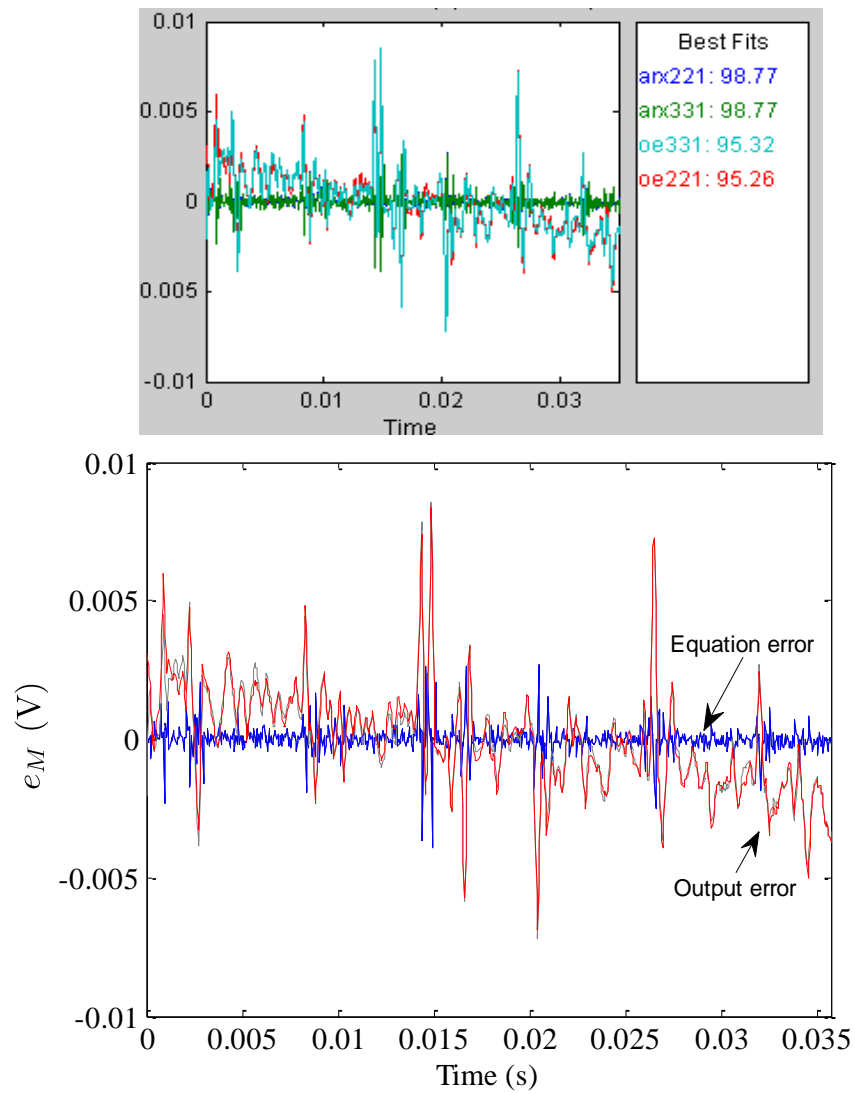


Fig. 6.14 Model errors comparison between third/second order output error and equation error model

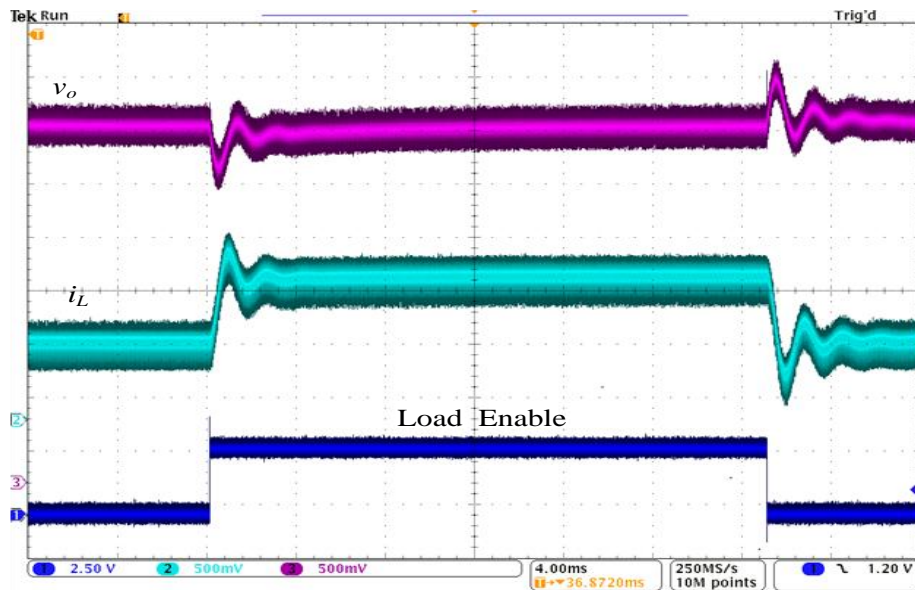
6.6 Adaptive Controller / Experimental Validation

This section presents the practical validation of the proposed adaptive PD+I control system. Initially, a conventional PID voltage controller is implemented on the experimental hardware. The PID is set to control the buck converter output voltage at 3.3 V. This serves as a benchmark for testing the adaptive PD+I controller based on the DCD-RLS method. The PID gains are optimally tuned using the well-recognised pole-zero matching technique previously presented in Section 2.8.1. The transient characteristics of the PID controller are determined by applying a repetitive step change in load to the buck converter. This step change causes the load current to switch between 0.66 and 1.32 A at 25 ms intervals. The results shown in Fig. 6.15 demonstrate that the buck converter is always operating in continuous current-mode (CCM). The output voltage transient shows significant oscillatory behaviour at the points of load change.

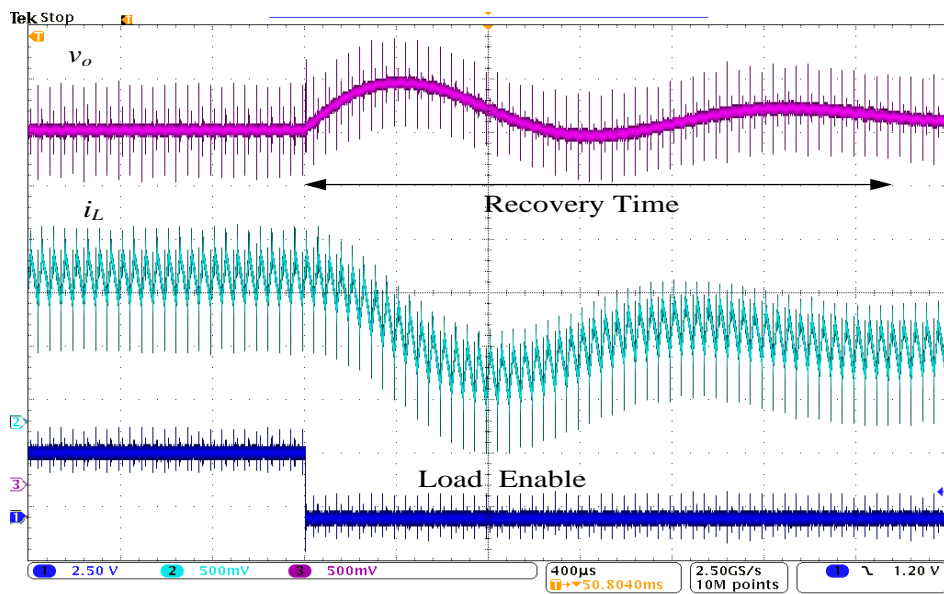
Following this, the DCD-RLS adaptive algorithm is implemented on the DSP for real time operation (Fig. 6.6). For consistency, all circuit parameters remain the same and the buck converter is subjected to the same load change as previously described. The experimental results shown in Fig. 6.16 are in excellent agreement with the simulation results in Fig. 5.12, thus confirming the successful real time implementation of the proposed DCD-RLS control scheme. Compared to the experimental results achieved with the conventional PID controller, the DCD-RLS scheme yields significantly improved transient performance for the same dynamic load change. The DCD-RLS method demonstrates lower transient overshoot, significantly less oscillatory behaviour and faster recovery time.

Finally, the LMS adaptive controller is implemented on the DSP. Here, each DCD-RLS in Fig. 5.1 is replaced with an adaptive LMS filter. As previously described, with the LMS-PEF, there is a need to carefully select an appropriate step size (μ). A range of step sizes have been experimentally tested and in agreement with the simulations, an optimal values of $\mu = 1.0$ is selected. Again, the same set of system parameters is used and the experimental results are shown in Fig. 6.17. These results are a good match to the initial simulation waveforms shown in Fig. 5.14. Compared to the conventional PID controller, the adaptive LMS controller offers improved

transient performance. However, as predicted by the simulation results and confirmed experimentally, the DCD-RLS offers superior dynamic performance over the LMS. In practical systems, the adaptive filter tap-weights can remain at the same value for a long time without changing. This situation is sometimes referred to as “stalling”. This can be caused by insufficient excitation in the signal to cause any change in the estimated filter coefficients. In LMS adaptive filters a high value of step size can be one solution to avoid stalling. Alternatively, a small random noise signal can be added to the filter tap-weights, this may prevent the stalling effects [80]. It is worth noting that the switching frequency effect seen on the experimental waveforms is due to the common mode noise on the oscilloscope probe.

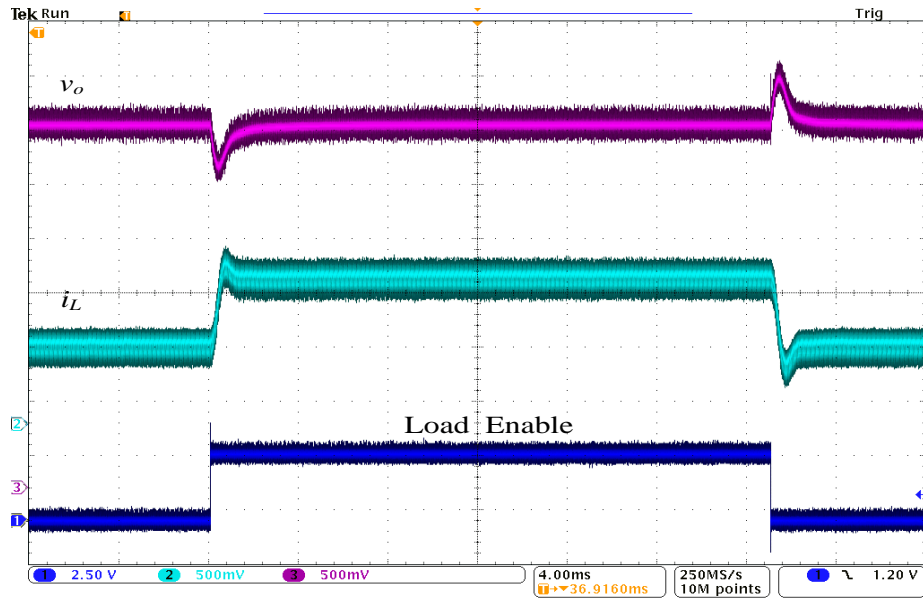


(a)

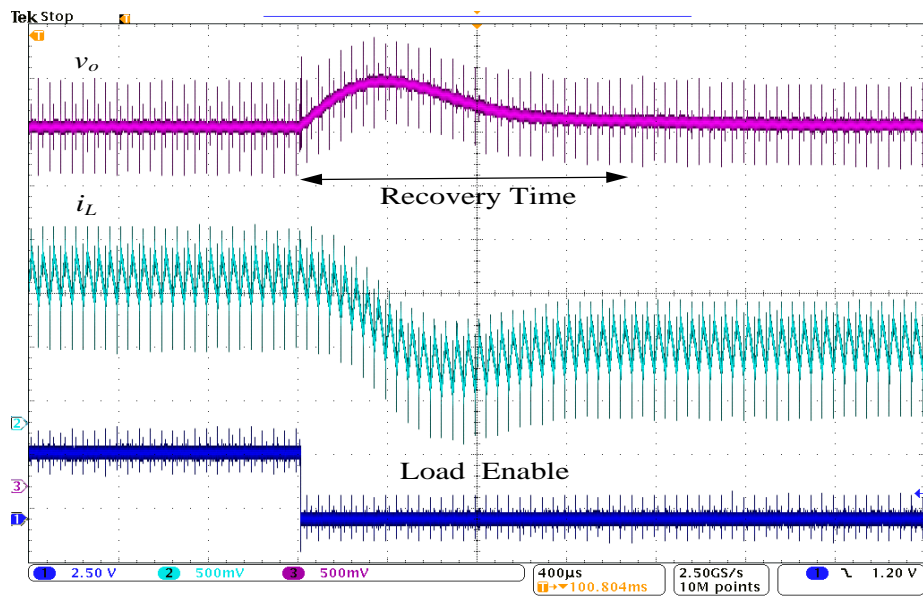


(b)

Fig. 6.15 Transient response of PID controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient

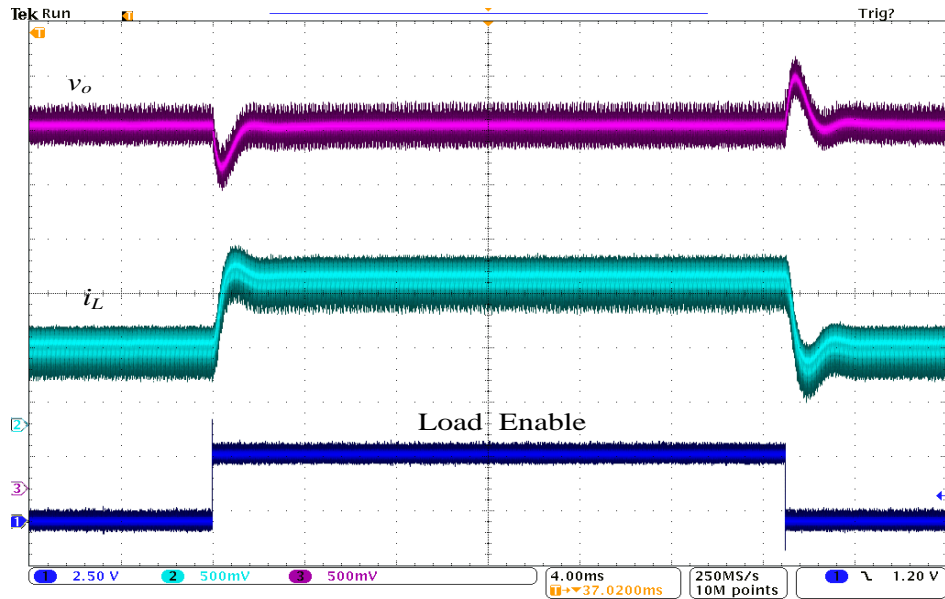


(a)

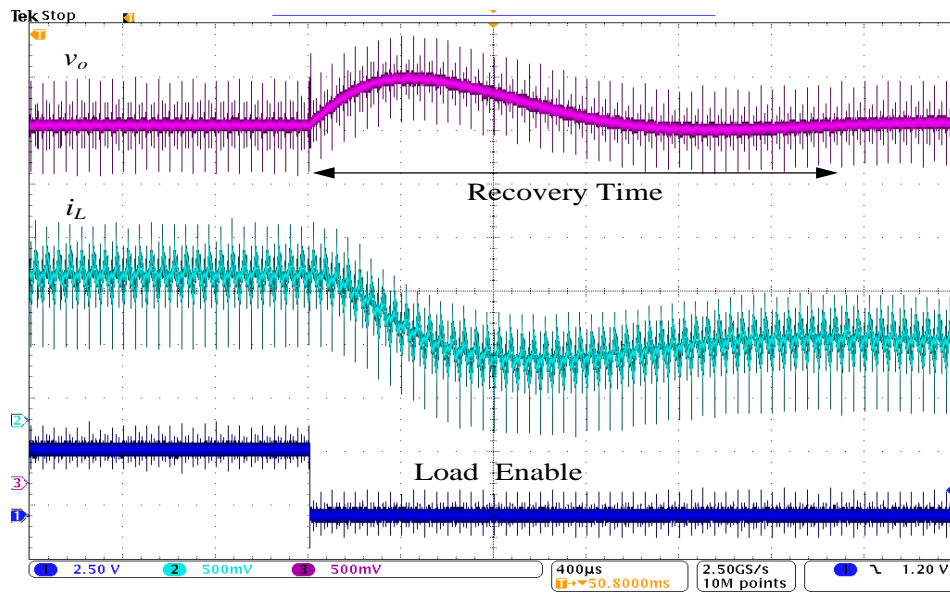


(b)

Fig. 6.16 Transient response of adaptive PD+I DCD-RLS controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient



(a)



(b)

Fig. 6.17 Transient response of adaptive PD+I LMS controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient

6.7 Complexity Reduction

In most applications, there is a trade-off between the dynamic performance and computational complexity (*i.e.*, speed of execution) of the controller. In adaptive PD+I controller two solutions are presented, each giving a different weighting to these two important performance indicators. The LMS is designed for good dynamic performance with low computational complexity, while the DCD-RLS is designed for optimum dynamic performance. The DCD-RLS is a computational-efficient algorithm compared to the classic RLS schemes, but it is acknowledged that a higher computational burden than the LMS exists. For this reason, the overall system complexity of the proposed DCD-RLS scheme (Fig. 5.1) can be reduced by exchanging the second stage DCD-RLS for a classical LMS-PEF. The first stage DCD-RLS still remains in place. In this way, we develop a “hybrid” DCD-RLS: LMS control scheme. This change does not appear to significantly compromise the behaviour of the system response with respect to convergence time, identification accuracy, and control error signal power, even during the initial transient or due to a significant change in the system parameters. When the first stage is faced with a high error signal, the DCD-FIR filter influences the prediction error signal. This prediction error signal is then passed onto the second stage LMS-FIR filter to adapt the tap weights and adaptive gain. The simulation results from the DCD-RLS:LMS system are shown in Fig. 6.18 (load change: 1.32A-to-6.5 A). The experimental results are shown in Fig. 6.19 (load change: 1.32–0.66 A). Here, the same conditions have been used as those originally specified in section 6.6. It can be seen that the dynamic performance of hybrid DCD-RLS: LMS achieves an excellent response.

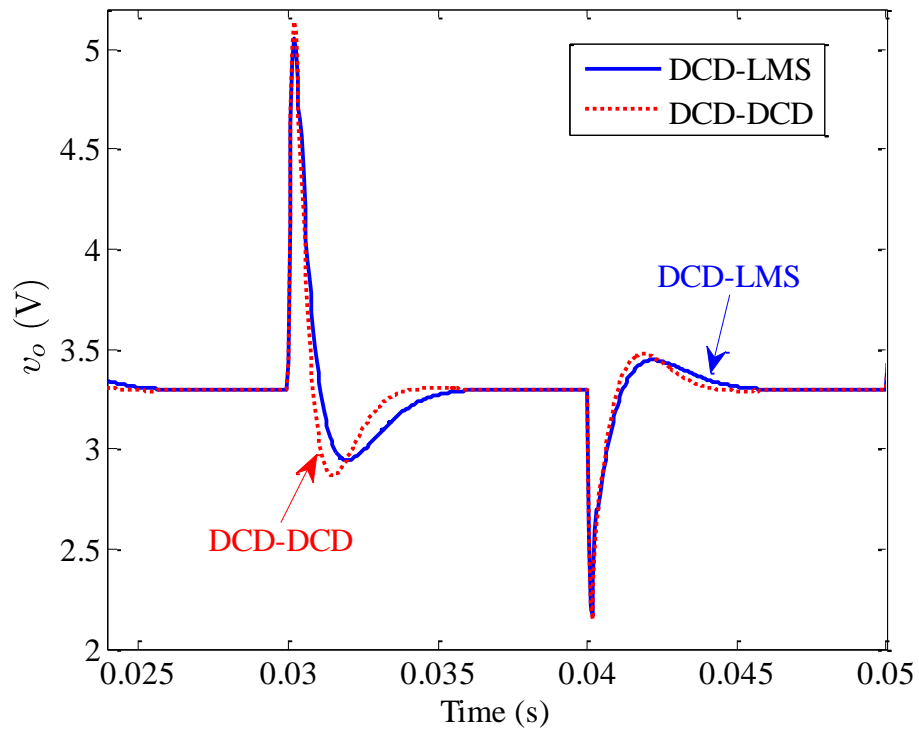
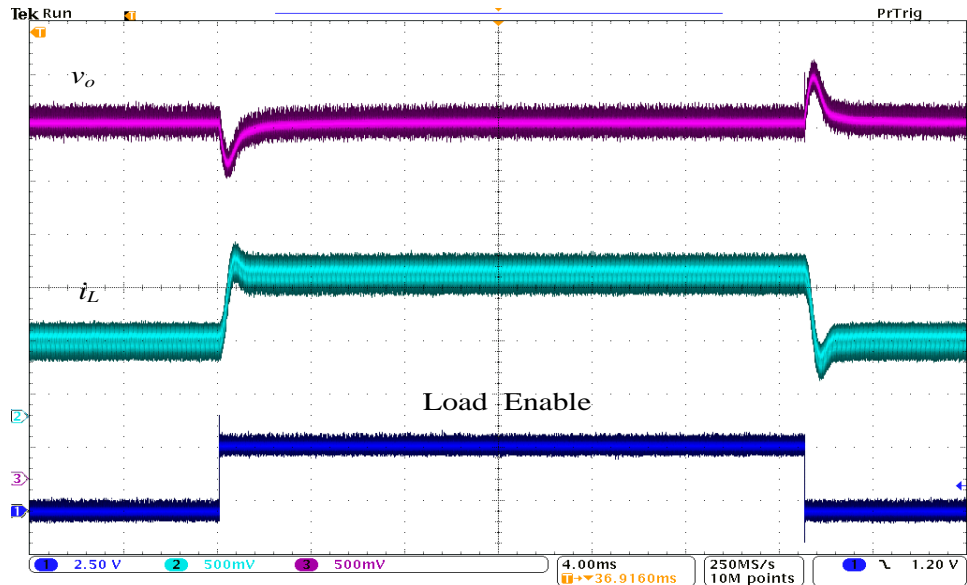
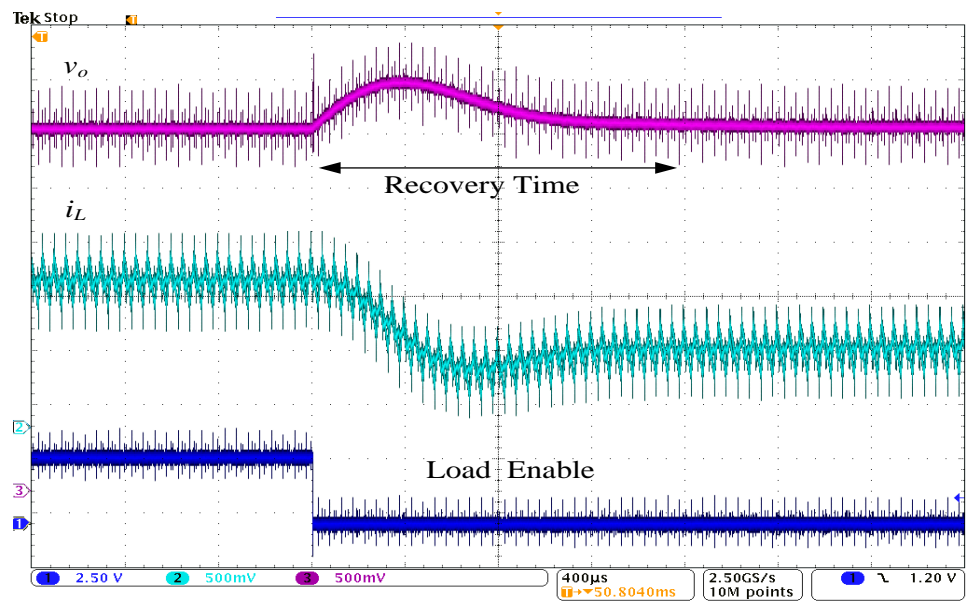


Fig. 6.18 Load transient response at significant change in load current, with two stage DCD-DCD adaptive controller and hybrid DCD-LMS adaptive controller



(a)



(b)

Fig. 6.19 Transient response of hybrid DCD-RLS:LMS ($\mu = 1$) adaptive controller with abrupt load change between 0.66 A and 1.32 A. (a) 4 ms/div: showing two transient changes. (b) 400 μ s/div: “zoom-in” on second transient

6.8 Chapter Summary

This chapter has focused on the experimental validation of the novel leading DCD-RLS system identification algorithm presented in Chapter 4 and the proposed adaptive PD+I controller scheme illustrated in Chapter 5. The experimental results of the system identification scheme are in close agreement to the simulation results presented in Chapter 4, demonstrating the viability of the proposed algorithm for real time application. Furthermore, the results demonstrate that the parameter estimation of the DCD-RLS is comparable to conventional RLS method but with reduced computational complexity. This chapter has also successfully demonstrated that the proposed algorithm can be directly embedded into adaptive and self-tuning digital control systems to improve controller performance. Experimental results show that by applying the DCD-RLS algorithm in the PD+I structure superior dynamic performance and voltage regulation can be achieved compared to the conventional PID controller.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In SMPCs, parameter estimation is essential to acquire an appropriate model of the system and is a first step in developing adaptive and self-tuning controllers. To be successful, any system identification scheme must be able to respond to the characteristics of the system. However, to achieve high levels of accuracy and/or estimation speed typically implies the need for sophisticated identification methods which require significant signal processing to implement. Unfortunately, in applications, such as SMPCs, cost and complexity are a major concern.

Conventional RLS algorithms provide fast convergence speed, small prediction error, and accurate parametric estimation. However, they often have limited application in SMPCs and other low power, low cost applications due to computationally heavy calculations demanding significant hardware resources. Therefore, RLS schemes are not always viable for real time estimation, where it is necessary to keep system costs low and competitive.

For this reason, this thesis makes a research contribution in the area of low complexity parameter estimation algorithms for the system identification and adaptive control of SMPCs. The work specifically presents a system identification/prediction error filter structure based on the DCD-RLS algorithm. Several novel approaches have been presented in this thesis.

❖ DCD-RLS System Identification of dc-dc Converter:

Here, a novel on-line system identification method is proposed to overcome the limitations of many classic RLS algorithms. The proposed algorithm can be implemented in many alternative applications where accurate and efficient parameter estimation is required. In this research, specific attention is given to the parameter estimation of dc-dc SMPCs. The solution based around the DCD-RLS algorithm is proven to be computationally efficient and utilises an IIR adaptive filter as the identification model. The IIR filter parameters are estimated on a cycle-by-cycle basis by superimposing a 9-bit PRBS into the control signal and monitoring the output signal response. Results demonstrate the effectiveness of the proposed solution. The identification method is able to accurately estimate the model parameters and quickly minimise the prediction error power. In addition, it is capable of working continuously in the control loop. Simulation and experimental results, based upon a prototype synchronous dc-dc buck converter controlled by Texas Instruments TMS320F28335™ DSP, show that the DCD-RLS algorithm provide a very good identification metrics (convergence rate, parameters estimation, and prediction error) and the system identification performance is comparable to other complex solutions such as recursive least squares (RLS) techniques. Importantly, the DCD-RLS algorithm reduces the computational complexity of the classical RLS algorithms; thus offering an efficient hardware solution which is well suited to real time applications. As a result, the proposed scheme can be directly embedded into adaptive and self-tuning digital controllers to improve the control performance of a wide range of industrial and commercial applications.

A further research contribution of this thesis is incorporating a new adaptive forgetting factor strategy to the DCD-RLS technique. This scheme is based on fuzzy logic and uses a two input, single output adaptive forgetting factor. The fuzzy logic approach is shown to improve the model estimation during abrupt load changes within the SMPC. The tracking approach relies on monitoring the prediction error signal, where it is possible to detect fast changes in the system.

The results and conclusions of this work have successfully been published in the following journal and international conference papers:

- 1- **M. Algreer**, M. Armstrong, and D. Giaouris, "Active On-Line System Identification of Switch Mode DC-DC Power Converter Based on Efficient Recursive DCD-IIR Adaptive Filter", *IEEE Transactions on Power Electronics*, vol.27, pp.4425-4435, Nov. 2012.
- 2- **M. Algreer**, M. Armstrong, and D. Giaouris, "System Identification of PWM DC-DC Converters during Abrupt Load Changes," in *Proc. IEEE Industrial Electron. Conf., IECON'09*, 2009, pp. 1788 – 1793, Porto, Portugal.

❖ **Adaptive control based on DCD-RLS and LMS PEF:**

The second major contribution of this thesis is the alternative application of the DCD-RLS algorithm for the adaptive control of SMPCs. In this case, the proposed adaptive controller uses a simple two-stage/one-tap FIR adaptive PEF. This two-stage controller is shown to be comparable to a conventional PD controller. A non-adaptive integral controller (+I), is then introduced into the feedback loop to increase the excitation of the filter tap-weight and ensure good output voltage regulation. In this way, the proposed controller applies an adaptive PD+I structure which offers an effective substitute to a conventional PID controller. The DCD-RLS algorithm is employed in this scheme as an adaptive PEF. Again, the main purpose is to reduce the computational complexity of the system which might typically employ a conventional RLS algorithm for this purpose. Simulation and experimental results, based upon a prototype synchronous dc-dc buck converter controlled by Texas Instruments TMS320F28335™ DSP, show that the adaptive PD+I controller, based on the DCD-RLS algorithm, is able to enhance the dynamic performance and convergence rate of the adaptive gains within the controller. As a result, the overall dynamic performance of the closed loop control system is significantly improved. The proposed approach results in a fast adaptive controller with self-loop compensation. In turn, the voltage error signal in the control loop is quickly minimised and will lead to minimise the prediction error signal. Results clearly show the superior dynamic performance compared to conventional PID and adaptive LMS control schemes. Sensitivity analysis shows the PD+I controller to be robust and stable.

Further reduction to the computation complexity of the proposed adaptive controller is also presented in this work. Here, a hybrid DCD-RLS:LMS control structure is developed. The motivation for this study is that whilst the DCD-RLS is a computationally efficient algorithm compared to classic RLS schemes, it must still be acknowledged that it presents a higher computational burden than conventional LMS algorithm. Therefore, the overall system complexity of the proposed DCD-RLS scheme can be reduced by exchanging the second stage DCD-RLS for a classical LMS-PEF. Experimental results show that this modification does not appear to significantly compromise the behaviour of the system response.

The results and conclusions of this work have successfully been published in the following journal and international conference papers:

- 1- **M. Algreer**, M. Armstrong, and D. Giaouris, “Adaptive PD+I Control of a Switch Mode DC-DC Power Converter Using a Recursive FIR Predictor”, *IEEE Transactions on Industry Applications*, vol.47, pp.2135-2144, Oct. 2011.
- 2- **M. Algreer**, M. Armstrong, and D. Giaouris, “Predictive PID Controller for DC-DC Converters Using an Adaptive Prediction Error Filter,” in *Proc. IET International Conf. on Power Electron., Machines and Drives, PEMD 2012*, vol. 2012, Bristol, United Kingdom.
- 3- **M. Algreer**, M. Armstrong, and D. Giaouris, “Adaptive Control of a Switch Mode DC-DC Power Converter Using a Recursive FIR Predictor,” in *Proc. IET International Conf. on Power Electron., Machines and Drives, PEMD 2010*, vol. 2010, Brighton, United Kingdom.

7.2 Future Work

This thesis has concentrated on system identification and adaptive control for a buck dc-dc SMPC. Therefore, application to other power converter topologies should be considered to further validate the application of the proposed techniques for power electronic applications. In particular, the performance of the PEF adaptive controller should be studied on multiphase SMPCs. Here, it is assumed that the order of the PEF will be increased and the impact of this is unknown at present.

Within the area of adaptive PEFs, it is suggested that further research work be carried out into enhancing the LMS algorithm to solve the problem of step-size selection and improve the dynamic performance. For example, a time variable step-size could potentially be used to speed up the convergence rate of the identification process and can be used to improve the overall response of the adaptive control system.

It may be worth investigating more optimal implementations of the adaptive controller based on DCD-RLS algorithm, potentially using dual-core microprocessor technology. Such an implementation could use one core to implement the control loop and the second core for system identification and control loop adaptation. Furthermore, the work on this project can be extended to focus on complete solutions for the purpose of system identification and adaptive control with emphasis on hardware optimisation for efficiency and low cost implementation. The proposed schemes are initially implemented through a DSP; however, more integrated solutions are possible and the algorithms are well suited for application in advanced FPGA and ASIC technologies.

The proposed adaptive algorithm (DCD-RLS) opens several potential topics that would make the on-line parameter estimation more useful for low cost and low complexity applications. For example, one can investigate on-line estimation of the SMPCs parameters based on limit-cycle oscillations (LCOs). With this technique it is possible to continuously identify the parameter of the model without injecting any excitation signal into the loop. The LCO is used as an excitation signal and this in turn could lead to a further reduction in the computation complexity of the identification process. In addition, more emphasis may be considered on inverse model adaptive filter techniques based on the DCD-RLS algorithm. This scheme can be applied for two purposes: system identification and adaptive controller, which may reduce the computation, overhead of the existing adaptive controller.

Alternatively, there is interesting research in the field of non-linear modelling of dc-dc converters. These methods require complicated numerical analysis and extensive off-line testing to develop an appropriate system model. Therefore, there is the potential to explore the application of non-linear adaptive filter algorithms for

system modelling. This will offer an on-line non-linear model of the system that may directly operate alongside the adaptive controller.

APPENDIX A

**DERIVATION OF RLS ALGORITHM BASED ON MATRIX
INVERSION LEMMA**

As described in Chapter 4 that the normal equation of the least square solution can be written as:

$$\hat{\mathbf{w}} = \left[\sum_{k=1}^n \mathbf{u}(k)\mathbf{u}^T(k) \right]^{-1} \sum_{k=1}^n y(k)\mathbf{u}(k) = \mathbf{R}^{-1}(n) \sum_{k=1}^n y(k)\mathbf{u}(k) \quad (\text{A.1})$$

where, $\mathbf{u}(n)$ is the data vector, $y(n)$ is the output signal of the system.

In weighting least square algorithm, the auto-correlation matrix $\mathbf{R}(n)$ and the cross-correlation vector $\boldsymbol{\beta}(n)$ can be given as [63]:

$$\mathbf{R}(n) = \eta(n, k) \sum_{k=1}^n \mathbf{u}(k)\mathbf{u}^T(k) \quad (\text{A.2})$$

$$\boldsymbol{\beta}(n) = \eta(n, k) \sum_{k=1}^n y(k)\mathbf{u}(k)$$

Let assume that the weighting function define as: $\eta(n, k) = \lambda(n)$. For simplicity we denotes to $\lambda(n)$ as λ [63].

Therefore, the solution in equation (A.1) can be reformulated in recursive form by assuming that $\hat{\mathbf{w}}(n-1)$ represents that previous time solution ($n-1$) of least square problem (A.1) [74]. Form this; the auto-correlation matrix can be defined as:

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + \mathbf{u}(n)\mathbf{u}^T(n) \quad (\text{A.3})$$

Now, equation (A.1) can be rearranged as:

$$\hat{\mathbf{w}} = \mathbf{R}^{-1}(n) \sum_{k=1}^n y(k) \mathbf{u}(k) = \mathbf{R}^{-1}(n) \left[\lambda \sum_{k=1}^{n-1} y(k) \mathbf{u}(k) + \mathbf{u}(n) y(n) \right] \quad (\text{A.4})$$

One can write:

$$\sum_{k=1}^{n-1} y(k) \mathbf{u}(k) = \mathbf{R}(n-1) \hat{\mathbf{w}}(n-1) \quad (\text{A.6})$$

By using equations (A.3)-(A.6) the estimated coefficients can be described as follows [63]:

$$\begin{aligned} \hat{\mathbf{w}} &= \mathbf{R}^{-1}(n) \left[\lambda \mathbf{R}(n-1) \hat{\mathbf{w}}(n-1) + \mathbf{u}(n) y(n) \right] \\ &= \mathbf{R}^{-1}(n) \left[\left(\mathbf{R}(n) - \mathbf{u}(n) \mathbf{u}^T(n) \right) \hat{\mathbf{w}}(n-1) + \mathbf{u}(n) y(n) \right] \\ &= \hat{\mathbf{w}}(n-1) + \mathbf{R}^{-1}(n) \mathbf{u}(n) \left[y(n) - \mathbf{u}^T(n) \hat{\mathbf{w}}(n-1) \right] \end{aligned} \quad (\text{A.7})$$

Finally, the recursive solution of the filter coefficients can be written as:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{R}^{-1}(n) \mathbf{u}(n) \left[y(n) - \mathbf{u}^T(n) \hat{\mathbf{w}}(n-1) \right] \quad (\text{A.8})$$

Matrix inversion lemma

As given in equation (A.8), the estimation of the parameters of the system require at each time instant to find the matrix inverse of $\mathbf{R}(n)$. To overcome this issue a matrix inversion lemma can be used [63]:

$$(\mathbf{A} + \mathbf{BCD})^{-1} = (\mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{C}^{-1} + \mathbf{D} \mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{D} \mathbf{A}^{-1}) \quad (\text{A.9})$$

Let suppose:

$$\mathbf{P}(n) = \mathbf{R}^{-1}(n), \mathbf{A} = \lambda \mathbf{R}(n-1), \mathbf{B} = \mathbf{D}^T = \mathbf{u}(n), \text{ and } \mathbf{C} = 1.$$

Thus:

$$\mathbf{P}(n) = \frac{1}{\lambda} \left[\mathbf{P}(n-1) - \frac{\mathbf{P}(n-1)\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{P}(n-1)}{\lambda + \mathbf{u}^T(n)\mathbf{P}(n-1)\mathbf{u}(n)} \right] \quad (\text{A.10})$$

Assuming that [80]:

$$\begin{aligned} \mathbf{S}(n) &= \mathbf{P}(n-1)\mathbf{u}(n) \\ \mathbf{k}(n) &= \frac{\mathbf{S}(n)}{\lambda + \mathbf{u}(n)^T \mathbf{S}(n)} \end{aligned} \quad (\text{A.11})$$

$$e(n) = y(n) - \mathbf{u}^T(n)\hat{\mathbf{w}}(n-1)$$

By inserting equation (A.11) into (A.10) this will result in:

$$\mathbf{P}(n) = \frac{1}{\lambda} \left[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{u}^T(n)\mathbf{P}(n-1) \right] \quad (\text{A.12})$$

Finally, by substituting equation (A.11) and (A.12) into (A.8), this yields [80]:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)e(n) \quad (\text{A.13})$$

APPENDIX B

SCHEMATIC CIRCUIT OF THE SYNCHRONOUS BUCK CONVERTER

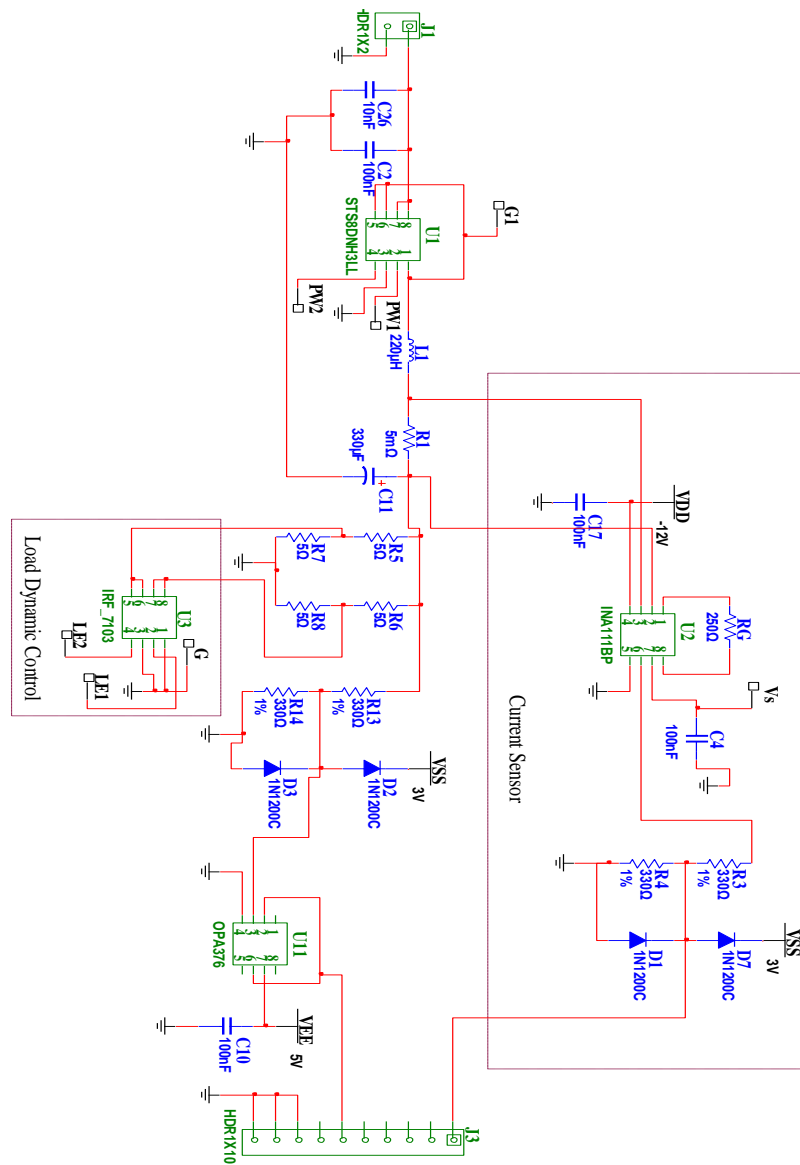


Fig. B.1 Schematic circuit of the buck converter

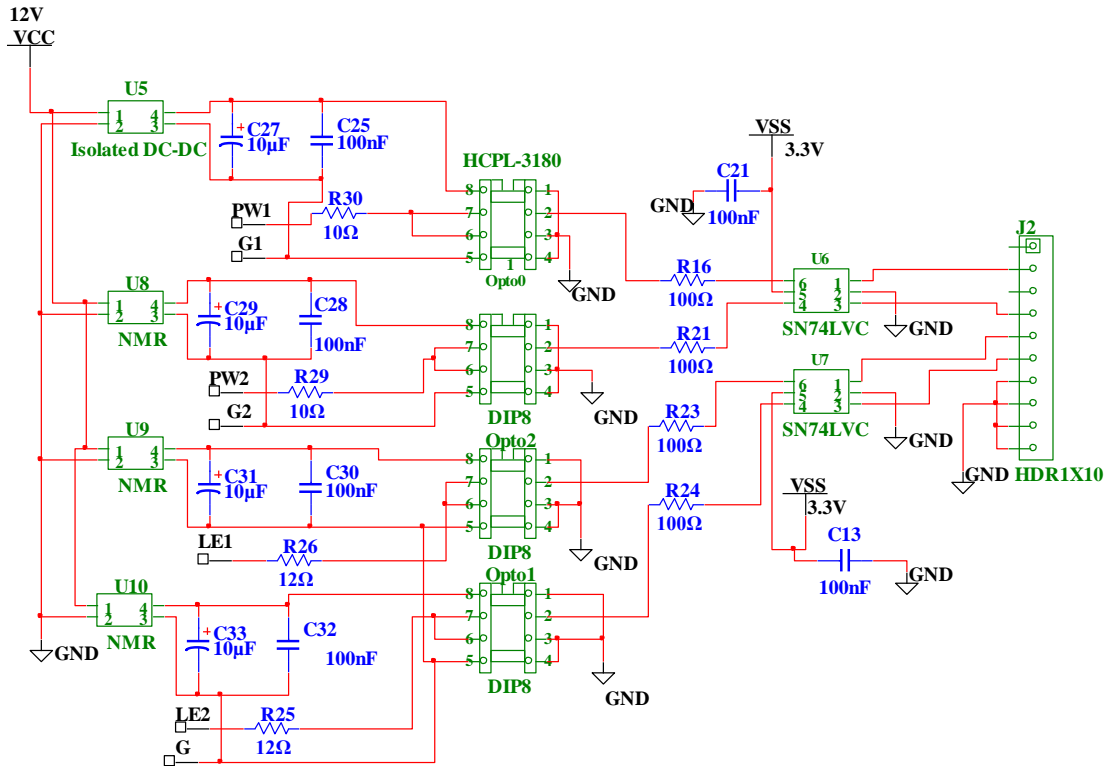


Fig. B.2 Schematic circuit of the isolated gate drive circuit

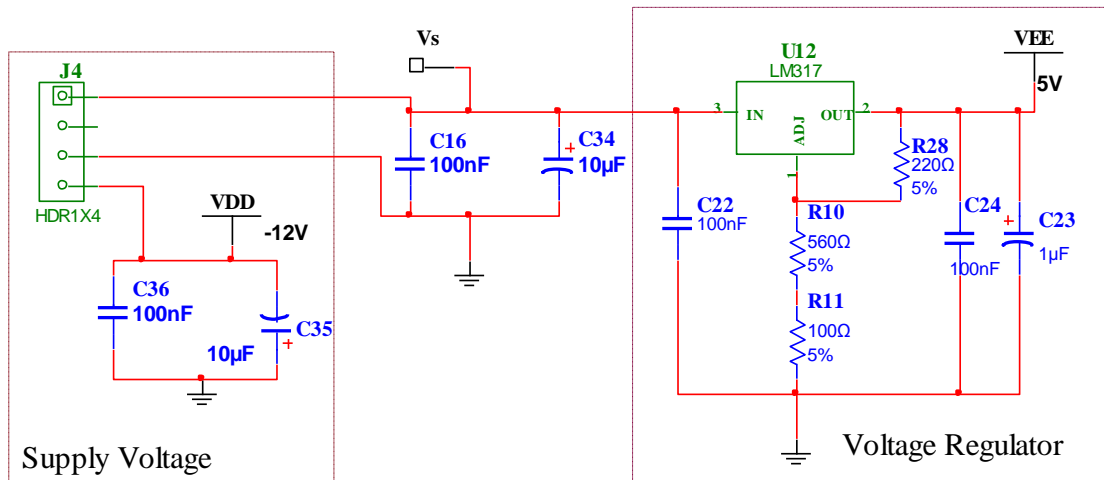


Fig. B.3 Schematic circuit of the analogue side power supply with 5 V voltage regulator

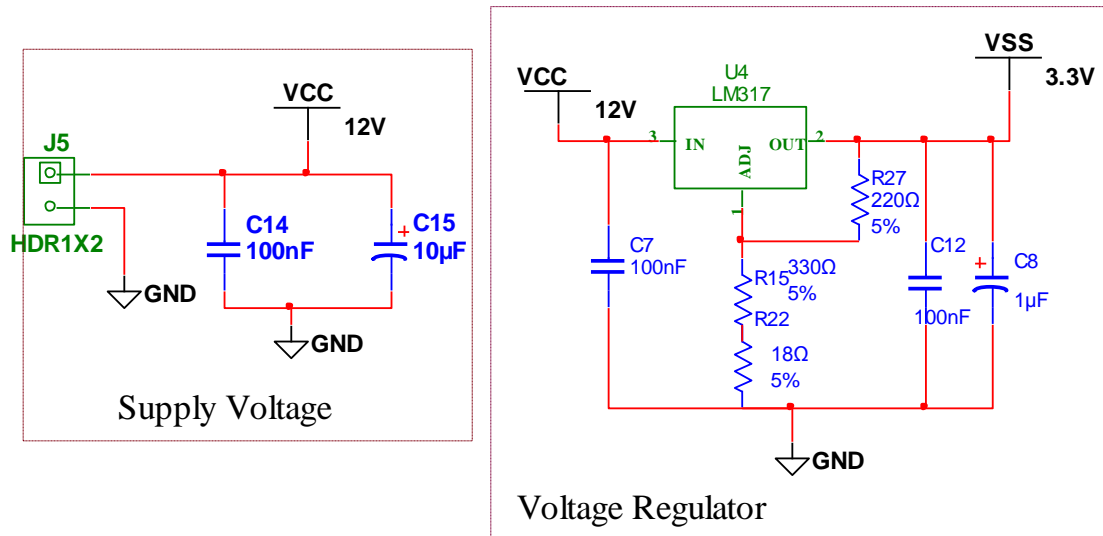


Fig. B.4 Schematic circuit of the digital side power supply with 3.3 V voltage regulator

APPENDIX C
SIMULINK MODEL OF THE PROPOSED STRUCTURES

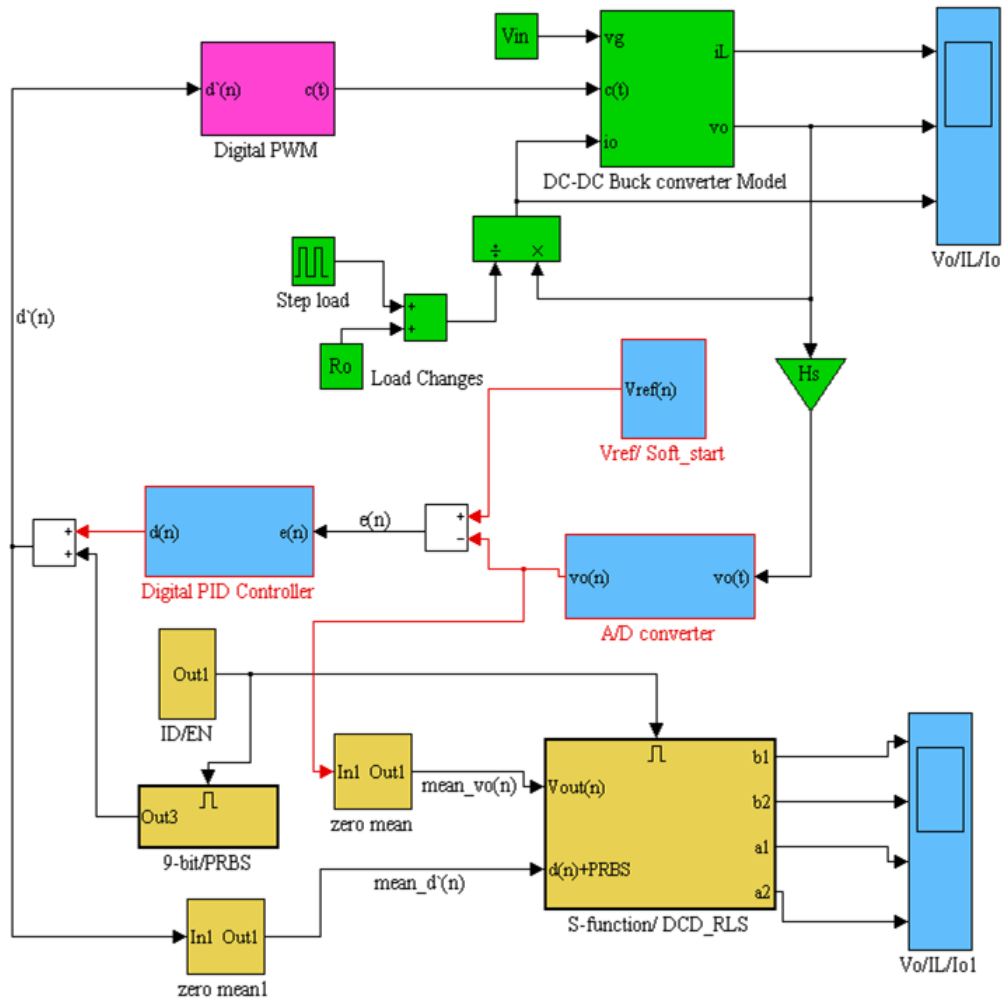


Fig. C.1 Simulink model of the proposed system identification structure

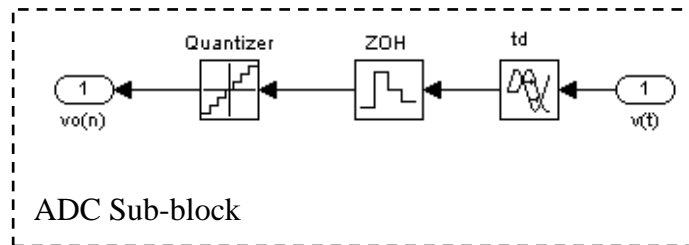
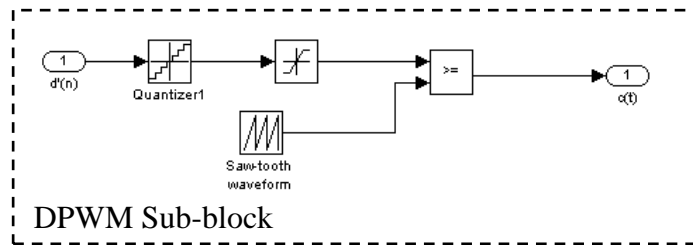
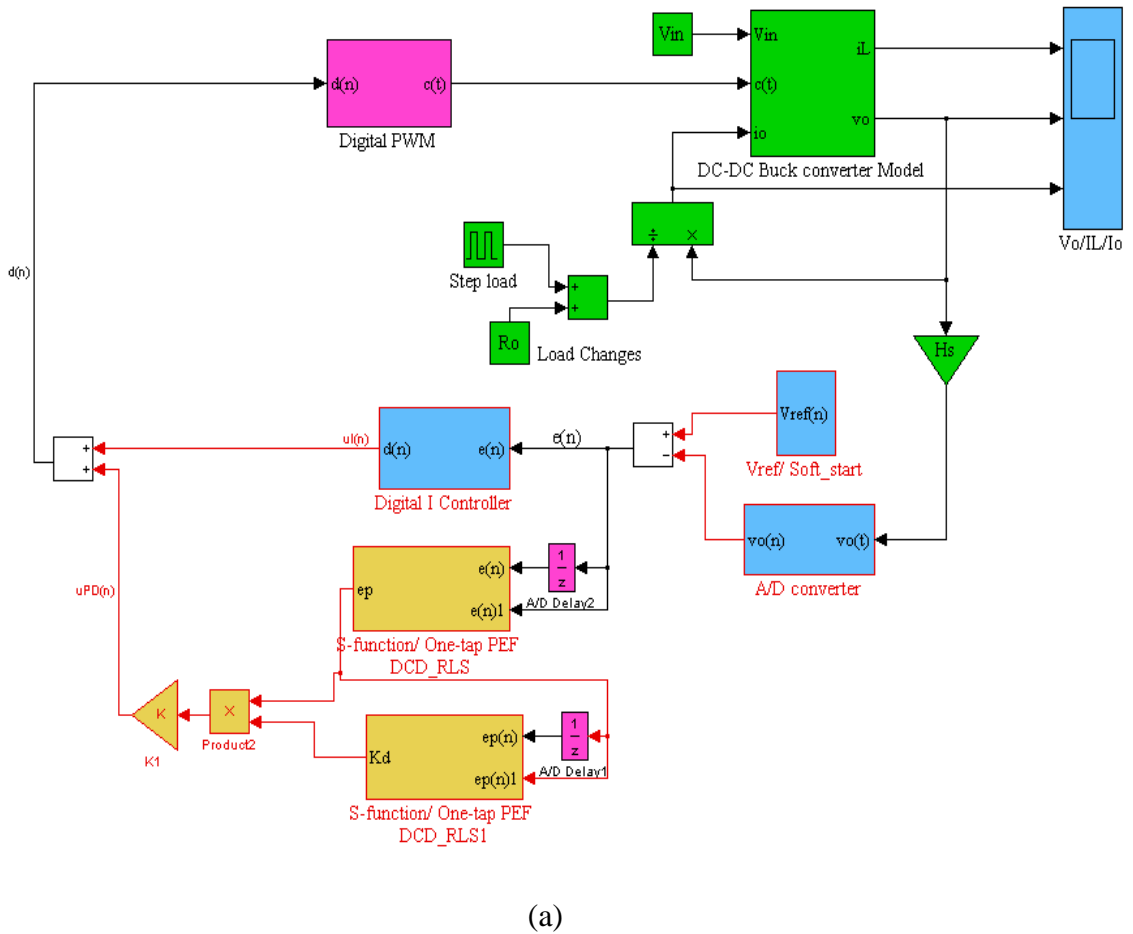


Fig. C.2 a: Simulink model of the adaptive PD+I controller, b: Digital PWM sub block, c: ADC sub block

REFERENCES

- [1] M. M. Peretz and S. Ben-Yaakov, "Time-Domain Identification of Pulse-Width Modulated Converters," *IET Power Electronics*, vol. 5, pp. 166-172, 2012.
- [2] 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, pp. 2578-2588, 2009.
- [3] J. Morroni, R. Zane, and D. Maksimovic, "Design and Implementation of an Adaptive Tuning System Based on Desired Phase Margin for Digitally Controlled DC-DC Converters," *IEEE Transactions on Power Electronics*, vol. 24, pp. 559-564, 2009.
- [4] J. Morroni, R. Zane, and D. Maksimovic, "An Online Stability Margin Monitor for Digitally Controlled Switched-Mode Power Supplies," *IEEE Transactions on Power Electronics*, vol. 24, pp. 2639-2648, 2009.
- [5] B. Miao, R. Zane, and D. Maksimovic, "Automated Digital Controller Design for Switching Converters," in *IEEE Power Electronics Specialists Conference (PESC 2005)*, 2005, pp. 2729-2735.
- [6] N. Kong, A. Davoudi, M. Hagen, E. Oettinger, X. Ming, H. Dong Sam, and F. C. Lee, "Automated System Identification of Digitally-Controlled Multi-phase DC-DC Converters," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2009)*, 2009, pp. 259-263.

-
- [7] 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.
- [8] S. Davis, "Controller IC Employs Real-Time Adaptive Loop Compensation, PMBus " in *Power Electronics Technology Magazine*, 2012
- [9] S. Davis, "Adaptive Control IC Creates Self-Adjusting DC-DC Converters," in *Power Electronics Technology Magazine*, 2009.
- [10] Intersil. Data Sheet (December 2010). Adaptive Digital DC/DC Controller with Drivers and Current Sharing. [Online]. Available: <http://www.intersil.com/content/dam/Intersil/documents/fn68/fn6876.pdf>.
- [11] M. Hagen and V. Yousefzadeh, "Applying Digital Technology to PWM Control-Loop Designs," in *Power Supply Design Seminar (SEM-1800)*, Topic 7, 2008-2009, pp. 7.1-7.28.
- [12] Texas Instrument. (July 2008). UCD 9240 Digital PWM System Controller. [Online]. Available: <http://focus.ti.com/lit/ds/symlink/ucd9240.pdf>.
- [13] M. Algreer, M. Armstrong, and D. Giaouris, "Active On-Line System Identification of Switch Mode DC-DC Power Converter Based on Efficient Recursive DCD-IIR Adaptive Filter," *IEEE Transactions on Power Electronics*, 2012.
- [14] M. Algreer, M. Armstrong, and D. Giaouris, "System Identification of PWM DC-DC Converters During Abrupt Load Changes," in *IEEE Industrial Electronics Conference (IECON 2009)*, 2009, pp. 1788-1793.
- [15] M. Algreer, M. Armstrong, and D. Giaouris, "Adaptive Control of a Switch mode DC-DC Power Converter Using a Recursive FIR Predictor," in *5th IET International Conference on Power Electronics, Machines and Drives (PEMD 2010)*, 2010, pp. 1-6.

-
- [16] M. Algreer, M. Armstrong, and D. Giaouris, "Adaptive PD+I Control of a Switch-Mode DC-DC Power Converter Using a Recursive FIR Predictor," *IEEE Transactions on Industry Applications*, vol. 47, pp. 2135-2144, 2011.
- [17] M. Algreer, M. Armstrong, and D. Giaouris, "Predictive PID Controller for DC-DC Converters Using an Adaptive Prediction Error Filter," in *7th IET International Conference on Power Electronics, Machines and Drives (PEMD 2012)*, 2012.
- [18] R. W. Erickson and D. Maksimovi, *Fundamentals of Power Electronics*, 2nd ed. Norwell, Massachusetts, USA.: Kluwer Academic Publisher Group, 2001.
- [19] M. Ned, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications and Design*, 3rd ed.: John Wiley & Sons, Inc., 2003.
- [20] V. Yousefzadeh, "Advances in Digital Power Control," in *32nd International Telecommunications Energy Conference (INTELEC)*, 2010, pp. 1-8.
- [21] D. Maksimovic and R. Zane, "Small-Signal Discrete-Time Modeling of Digitally Controlled PWM Converters," *IEEE Transactions on Power Electronics*, vol. 22, pp. 2552-2556, 2007.
- [22] B. Hutchings, "SMPS Buck Converter Design Example," Microchip. Web Seminars. (May 2006). [Online]. Available: <http://techtrain.microchip.com/webseminars/ArchivedDetail.aspx?Active=109>. Retrieved: March 2012.
- [23] A. J. Forsyth and S. V. Mollov, "Modelling and Control of DC-DC Converters," *Power Engineering Journal*, vol. 12, pp. 229-236, 1998.
- [24] G. Liping, J. Y. Hung, and R. M. Nelms, "Evaluation of DSP-Based PID and Fuzzy Controllers for DC-DC Converters," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 2237-2248, 2009.
- [25] S. Jen-Ta, L. De-Min, L. Chih-Wen, and H. Chung-Wen, "An Adaptive Control Method for Two-Phase DC/DC Converter," in *International*

- Conference on Power Electronics and Drive Systems (PEDS 2009)* 2009, pp. 288-293.
- [26] S. Abe, M. Ogawa, T. Zaitso, S. Obata, M. Shoyama, and T. Ninomiya, "Power-Stage Frequency Response Cancellation of DC-DC Converter With Digital Control," in *International Symposium on Power Electronics Electrical Drives Automation and Motion (SPEEDAM)*, 2010, pp. 44-49.
- [27] 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.
- [28] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Menlo Park, Calif.: Addison-Wesley, 1998.
- [29] D. M. Van de Sype, K. De Gusseme, F. M. L. L. De Belie, A. P. Van den Bossche, and J. A. Melkebeek, "Small-Signal z-Domain Analysis of Digitally Controlled Converters," *IEEE Transactions on Power Electronics*, vol. 21, pp. 470-478, 2006.
- [30] M. M. Peretz and S. Ben-Yaakov, "Time Domain Identification of PWM Converters for Digital Controllers Design," in *IEEE Power Electronics Specialists Conference (PESC 2007)*, 2007, pp. 809-813.
- [31] G. E. Pitel and P. T. Krein, "Real-Time System Identification for Load Monitoring and Transient Handling of DC-DC Supplies," in *IEEE Power Electronics Specialists Conference (PESC 2008)*, 2008, pp. 3807-3813.
- [32] B. J. Patella, A. Prodic, A. Zirger, and D. Maksimovic, "High-Frequency Digital PWM Controller IC for DC-DC Converters," *IEEE Transactions on Power Electronics*, vol. 18, pp. 438-446, 2003.
- [33] Z. Zhao, "Design and Practical Implementation of Digital Auto-Tuning and Fast-Response Controllers for Low-Power Switch-Mode Power Supplies," PhD Thesis, University of Toronto, 2008.

-
- [34] H. Peng, A. Prodic, E. Alarcon, and D. Maksimovic, "Modeling of Quantization Effects in Digitally Controlled DC-DC Converters," *IEEE Transactions on Power Electronics* vol. 22, pp. 208-215, 2007.
- [35] A. V. Peterchev, X. Jinwen, and S. R. Sanders, "Architecture and IC Implementation of a Digital VRM Controller," *IEEE Transactions on Power Electronics*, vol. 18, pp. 356-364, 2003.
- [36] S. Buso and P. Mattavelli, *Digital Control in Power Electronics* vol. 1: Morgan and Claypool Publishers, 2006.
- [37] Y. Duan and H. Jin, "Digital Controller Design for Switch Mode Power Converters," in *IEEE Applied Power Electronics Conference and Exposition (APEC 1999)*, 1999, pp. 967-973 vol.2.
- [38] T. W. Martin and S. S. Ang, "Digital Control for Switching Converters," in *IEEE International Symposium on Industrial Electronics (ISIE 1995)*, 1995, pp. 480-484 vol.2.
- [39] A. Kelly and K. Rinne, "Sensorless Current-mode Control of a Digital Dead-beat DC-DC Converter," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2004)*, 2004, pp. 1790-1795 Vol.3.
- [40] A. Prodic, D. Maksimovic, and R. W. Erickson, "Design and Implementation of a Digital PWM Controller for a High-Frequency Switching DC-DC Power Converter," in *IEEE Industrial Electronics Conference (IECON 2001)*, 2001, pp. 893-898 vol.2.
- [41] X. Haiping, W. Xuhui, and K. Li, "DSP-Based Digitally Controlled Bi-Directional DC-DC Converter," in *IEEE Industrial Electronics Conference (IECON 2004)*, 2004, pp. 800-804 Vol. 1.
- [42] H. Haitoa, V. Yousefzadeh, and D. Maksimovic, "Nonlinear Control for Improved Dynamic Response of Digitally Controlled DC-DC Converters," in *IEEE Power Electronics Specialists Conference (PESC 2006)*, 2006, pp. 1-7.

-
- [43] K. D. Wilkie, M. P. Foster, D. A. Stone, and C. M. Bingham, "Hardware-in-the-Loop Tuning of a Feedback Controller for a Buck Converter Using a GA," in *International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM 2008)*, 2008, pp. 680-684.
- [44] G. Shuibao, L.-S. Xuefang, B. Allard, G. Yanxia, and R. Yi, "Digital Sliding-Mode Controller for High-Frequency DC/DC SMPS," *IEEE Transactions on Power Electronics*, vol. 25, pp. 1120-1123.
- [45] G. Yanxia, G. Shuibao, X. Yanping, L. Shi Xuefang, and B. Allard, "FPGA-based DPWM for Digitally Controlled High-Frequency DC-DC SMPS," in *3rd International Conference on Power Electronics Systems and Applications (PESA 2009)*, 2009, pp. 1-7.
- [46] A. V. Peterchev and S. R. Sanders, "Quantization Resolution and Limit Cycling in Digitally Controlled PWM Converters," *IEEE Transactions on Power Electronics*, vol. 18, pp. 301-308, 2003.
- [47] N. Yue, G. Yanxia, and G. Shuibao, "A Dual-mode Digital Controller for Switching Mode Power Supply," in *4th International Conference on Power Electronics Systems and Applications (PESA)*, 2011, pp. 1-7.
- [48] L. Corradini, P. Mattavelli, W. Stefanutti, and S. Saggini, "Simplified Model Reference-Based Autotuning for Digitally Controlled SMPS," *IEEE Transactions on Power Electronics*, vol. 23, pp. 1956-1963, 2008.
- [49] V. P. Arikatla and J. A. Abu Qahouq, "Adaptive Digital Proportional-Integral-Derivative Controller for Power Converters," *IET Power Electronics*, vol. 5, pp. 341-348, 2012.
- [50] A. Kiam Heong, G. Chong, and L. Yun, "PID Control System Analysis, Design, and Technology," *IEEE Transactions on Control Systems Technology*, vol. 13, pp. 559-576, 2005.

-
- [51] V. Yousefzadeh and S. Choudhury, "Nonlinear Digital PID Controller for DC-DC Converters," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2008)*, 2008, pp. 1704-1709.
- [52] Y. F. 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.
- [53] H. Al-Atrash and I. Batarseh, "Digital Controller Design for a Practicing Power Electronics Engineer," in *IEEE Applied Power Electronics Conference (APEC 2007)*, 2007, pp. 34-41.
- [54] Z. Zhenyu 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.
- [55] V. Yousefzadeh, W. Narisi, Z. Popovic, and D. Maksimovic, "A Digitally Controlled DC/DC Converter for an RF Power Amplifier," *IEEE Transactions on Power Electronics*, vol. 21, pp. 164-172, 2006.
- [56] A. Prodic and D. Maksimovic, "Design of a Digital PID Regulator Based on Look-Up Tables for Control of High-Frequency DC-DC Converters," in *IEEE Workshop on Computers in Power Electronics*, 2002, pp. 18-22.
- [57] A. Kelly and K. Rinne, "Control of DC-DC Converters by Direct Pole Placement and Adaptive Feedforward Gain Adjustment," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2005)*, 2005, pp. 1970-1975 Vol. 3.
- [58] G. Shuibao, G. Yanxia, X. Yanping, L.-S. Xuefang, and B. Allard, "Digital PWM Controller for High-Frequency Low-Power DC-DC Switching Mode Power Supply," in *IEEE Conference on Power Electronics and Motion Control (IPEMC 2009)*, 2009, pp. 1340-1346.
- [59] L.-S. Xuefang, F. Morel, B. Allard, D. Tournier, J. M. Retif, and G. Shuibao, "A Digital-Controller Parameter-Tuning Approach, Application to a Switch-

- Mode Power Supply," in *IEEE International Symposium on Industrial Electronics (ISIE)*, , 2007, pp. 3356-3361.
- [60] J. B. V.Bobal, J.Fessl, and J.Machacek, *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*: Springer-Verlag London limited 2005.
- [61] H. D. Venable, "Testing Modern Power Supplies," in *Test & Measurement World Magazine*. vol. 5, No. 11, November 1985, pp. 118-121.
- [62] G. P. Rao and H. Unbehauen, "Identification of Continuous-Time Systems," *IEE Proceedings-Control Theory and Applications*, vol. 153, pp. 185-220, 2006.
- [63] L.Ljung, *System Identification: Theory for the User*, 2nd ed.: Upper Saddle River, NJ: Prentice Hall, 1999.
- [64] R. Isermann, *Digital Control Systems*, 2nd ed. New York: Springer-Verlag, 1989.
- [65] B. Johansson and M. Lenells, "Possibilities of Obtaining Small-signal Models of DC-to-DC Power Converters by Means of System Identification," in *Twenty-second International Conference on Telecommunications Energy (INTELEC)*, 2000, pp. 65-75.
- [66] 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, pp. 2603-2611, 2009.
- [67] 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.
- [68] 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, pp. 1093-1099, 2005.

-
- [69] M. Shirazi, J. Morroni, A. Dolgov, R. Zane, and D. Maksimovic, "Integration of Frequency Response Measurement Capabilities in Digital Controllers for DC-DC Converters," *IEEE Transactions on Power Electronics* vol. 23, pp. 2524-2535, 2008.
- [70] K. M. Moudgalya, *Digital Control*. Chichester: John Wiley, 2007.
- [71] K. W. V. To and A. K. David, "On-line Identification and Control of an AC/DC Power System," *International Journal of Electrical Power & Energy Systems*, vol. 18, pp. 223-227, 1996.
- [72] Y. D. Landau and G. Zito, *Digital Control Systems Design, Identification and Implementation*. Berlin ; New York: Springer, 2006.
- [73] V. Valdivia, A. Barrado, A. Lazaro, M. Sanz, D. Lopez del Moral, and C. Raga, "System-Level Black-Box Modeling of DC-DC Converters with Input Current Control for Fuel Cell Power Conditioning," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2012)*, 2012, pp. 443-450.
- [74] K. J. A. a. B. Wittenmark, *Adaptive Control*: Addison- Wesley Publishing Company, 1989.
- [75] B. Widrow and G. L. Plett, "Adaptive Inverse Control Based On Linear and Nonlinear Adaptive Filtering," in *International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*, 1996, pp. 30-38.
- [76] B. Widrow and G. L. Plett, "Nonlinear Adaptive Inverse Control," in *36th IEEE Conference on Decision and Control* 1997, pp. 1032-1037 vol.2.
- [77] M. Shafiq, "Internal Model Control Structure Using Adaptive Inverse Control Strategy," *ISA Transactions*, vol. 44, pp. 353-362, 2005.
- [78] M. Shafiq and S. Akhtar, "Inverse Model Based Adaptive Control of Magnetic Levitation System," in *5th IEEE Asian Conference on Control*, 2004, pp. 1414-1418 Vol.3.

-
- [79] G. A. Dumont and M. Huzmezan, "Concepts, Methods and Techniques in Adaptive Control," in *American Control Conference*, 2002, pp. 1137-1150 vol.2.
- [80] S. S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [81] P. S. R. Diniz, *Adaptive Filtering : Algorithms and Practical Implementation*, 2nd ed. Boston, London: Kluwer Academic, 2002.
- [82] B. Miao, R. Zane, and D. Maksimovic, "A Modified Cross-Correlation Method for System Identification of Power Converters With Digital Control," in *IEEE Power Electronics Specialists Conference (PESC 2004)*, 2004, pp. 3728-3733 Vol.5.
- [83] T. Roinila, M. Vilkkö, 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.
- [84] T. Roinila, M. Vilkkö, 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.
- [85] L. Jun-Yan, Y. Chun-Hung, and T. Chien-Hung, "Correlation-based System Identification of Digitally Controlled SMPS," in *IEEE International Conference on Power Electronics and Drive Systems (PEDS 2011)*, 2011, pp. 1149-1152.
- [86] M. Shirazi, R. Zane, D. Maksimovic, L. Corradini, and P. Mattavelli, "Autotuning Techniques for Digitally-Controlled Point-of-Load Converters with Wide Range of Capacitive Loads," in *IEEE Applied Power Electronics Conference (APEC 2007)*, 2007, pp. 14-20.

-
- [87] A. Costabeber, P. Mattavelli, S. Saggini, and A. Bianco, "Digital Autotuning of DC-DC Converters Based on a Model Reference Impulse Response," *IEEE Transactions on Power Electronics*, vol. 26, pp. 2915-2924, 2011.
- [88] W. Stefanutti, P. Mattavelli, S. Saggini, and M. Ghioni, "Autotuning of Digitally Controlled Buck Converters Based on Relay Feedback," in *IEEE Power Electronics Specialists Conference (PESC 2005)*, 2005, pp. 2140-2145.
- [89] L. Corradini, P. Mattavelli, and D. Maksimovic, "Robust Relay-feedback Based Autotuning for DC-DC Converters," in *IEEE Power Electronics Specialists Conference (PESC 2007)*, 2007, pp. 2196-2202.
- [90] Z. Zhenyu, A. Prodi, and P. Mattavelli, "Self-Programmable PID Compensator for Digitally Controlled SMPS," in *IEEE Workshops on Computers in Power Electronics (COMPEL 2006)*, 2006, pp. 112-116.
- [91] M. M. Peretz and S. Ben-Yaakov, "Time-domain Identification of PWM Converters Aided by the C2000 DSP Family," in *European DSP Education and Research Symposium (EDERS 2008)*, 2008, pp. 185-192.
- [92] F. Alonge, F. D'Ippolito, and T. Cangemi, "Identification and Robust Control of DC/DC Converter Hammerstein Model," *IEEE Transactions on Power Electronics*, vol. 23, pp. 2990-3003, 2008.
- [93] 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, pp. 1210-1221, 2007.
- [94] V. Valdivia, A. Barrado, A. Laazaro, P. Zumel, C. Raga, and C. Fernandez, "Simple Modeling and Identification Procedures for "Black-Box" Behavioral Modeling of Power Converters Based on Transient Response Analysis," *IEEE Transactions on Power Electronics*, vol. 24, pp. 2776-2790, 2009.
- [95] V. Valdivia, A. Barrado, A. Lazaro, C. Fernandez, and P. Zumel, "Black-box Modeling of DC-DC Converters Based on Transient Response Analysis and

- Parametric Identification Methods," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2010)*, 2010, pp. 1131-1138.
- [96] A. Kelly and K. Rinne, "A Self-Compensating Adaptive Digital Regulator for Switching Converters Based on Linear Prediction," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2006)*, 2006, p. 7 pp.
- [97] A. Kelly, "Self Compensating Closed Loop Adaptive Control System," US Patent App. 20,060/276,915, 2005.
- [98] W. Stefanutti, S. Saggini, E. Tedeschi, P. Mattavelli, and P. Tenti, "Simplified Model Reference Tuning of PID Regulators of Digitally Controlled DC-DC Converters Based on Crossover Frequency Analysis," in *IEEE Power Electronics Specialists Conference (PESC 2007)* 2007, pp. 785-791.
- [99] J. Mooney, A. E. Mahdi, A. Kelly, and K. Rinne, "Dual MAC Processor for Adaptive Control of Multiple High Switching Frequency DC-DC Converters," in *IET Irish Conference on Signals and Systems (ISSC 2008)*, 2008, pp. 116-121.
- [100] E. Santi, H. Y. Cho, A. B. Barkley, D. Martin, and A. Riccobono, "Tools to Address System Level issues in Power Electronics: The Digital Network Analyzer Method and the Positive Feedforward Control Technique," in *IEEE International Conference on Power Electronics and ECCE Asia (ICPE & ECCE)*, 2011, pp. 2106-2113.
- [101] A. Barkley, R. Dougal, and E. Santi, "Adaptive Control of Power Converters Using Digital Network Analyzer Techniques," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2011)*, 2011, pp. 1824-1832.
- [102] K. Tae-Jin, L. Jong-Pil, M. Byung-Duk, and Y. Dong-Wook, "A Diagnosis Method of DC/DC Converter Aging Based on the Variation of Parasitic Resistor," in *31st International Conference on Telecommunications Energy (INTELEC 2009)*, 2009, pp. 1-5.

-
- [103] J. Jin-Hong, K. Tae-Jin, K.-S. Kim, and K.-H. Kim, "Parameter Estimation of DC-DC Converter System for Operating Status Diagnosis," in *International Conference on Electrical Engineering, (ICEE 2004)*, 2004, pp. 881-884.
- [104] S. El Beid, S. Doubabi, and M. Chaoui, "Adaptive Control of PWM DC-to-DC Converters Operating in Continuous Conduction Mode," in *Mediterranean Conference on Control & Automation (MED 2007)*, 2007, pp. 1-5.
- [105] M. He and J. Xu, "Nonlinear PID in Digital Controlled Buck Converters," in *IEEE Applied Power Electronics Conference (APEC 2007)*, 2007, pp. 1461-1465.
- [106] H. F. Farahani, "Designing and Implementation of a Fuzzy Controller for DC-DC Converters and Comparing With PI Digital Controller," *Australian Journal of Basic and Applied Sciences*, vol. 5, pp. 276-285.
- [107] A. R. Ofoli and A. Rubaai, "Real-Time Implementation of a Fuzzy Logic Controller for Switch-Mode Power-Stage DC-DC Converters," *IEEE Transactions on Industry Applications*, vol. 42, pp. 1367-1374, 2006.
- [108] G. M. Di Blasi, V. Boscaino, P. Livreri, F. Marino, and M. Minieri, "A Novel Linear-Nonlinear Digital Control for DC/DC Converter with Fast Transient Response," in *IEEE Applied Power Electronics Conference and Exposition (APEC 2006)*, 2006, pp. 705-711.
- [109] S. Meng, S. Jing, and E. Prasad, "A Dual-loop Digital Controller for Switching DC-DC converters," in *IEEE Telecommunications Energy Conference (INTELEC 2006)*, 2006, pp. 1-5.
- [110] E. Meyer, Z. Zhang, and Y. Liu, "Digital Charge Balance Controller to Improve the Loading/Unloading Transient Response of Buck Converters," *IEEE Transactions on Power Electronics*, vol. 27, pp. 1314-1326, 2012.
- [111] J. Liang, W. Dong, E. Meyer, L. Yan-Fei, and P. C. Sen, "A Novel Digital Capacitor Charge Balance Control Algorithm With a Practical Extreme

- Voltage Detector," in *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2010 pp. 514-521.
- [112] Y. V. Zakharov, G. P. White, and L. Jie, "Low-Complexity RLS Algorithms Using Dichotomous Coordinate Descent Iterations," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3150-3161, 2008.
- [113] L. Jie, 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, pp. 2425-2438, 2009.
- [114] Y. V. Zakharov, B. Weaver, and T. C. Tozer, "Novel Signal Processing Technique for Real-Time Solution of the Least Squares Problem," in *2nd International Workshop on Signal Processing for Wireless Communications*, 2004, pp. 155-159.
- [115] Y. V. Zakharov, "Low-Complexity Implementation of the Affine Projection Algorithm," *IEEE Signal Processing Letters*, vol. 15, pp. 557-560, 2008.
- [116] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: Johns Hopkins University Press, 1996.
- [117] D. S. Watkins and Ebooks Corporation., *Fundamentals of Matrix Computations*, 2nd ed. Hoboken: John Wiley & Sons Inc., 2002.
- [118] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. UK: Cambridge University Press, 2004.
- [119] A. J. Fairweather, M. P. Foster, and D. A. Stone, "VRLA Battery Parameter Identification Using Pseudo Random Binary Sequences (PRBS)," in *5th IET International Conference on Power Electronics, Machines and Drives (PEMD 2010)*, 2010, pp. 1-6.
- [120] M. Allain, P. Viarouge, and F. Tourkhani, "The Use of Pseudo-Random Binary Sequences to Predict a DC-DC Converter's Control-to-Output Transfer Function in Continuous Conduction Mode," in *Canadian Conference on Electrical and Computer Engineering*, 2005, pp. 574-577.

- [121] L. Wang and R. Langari, "Identification of Time-Varying Fuzzy Systems" *International Journal of General Systems*, vol. 25, pp. 203 - 218, 1996.
- [122] X. Yuncan, Y. Qiwen, and Q. Jixin, "Combined Algorithm for Systems With Abrupt but Infrequent Parameter Changes Based on Robust Minmax and EW-RLS Estimation," in *IEEE International Conference on Systems, Man and Cybernetics*, 2003, pp. 177-180 vol.1.
- [123] T. H. Hunt, "A Comparative Study of System Identification Techniques in the Presence of Parameter Variation, Noise, and Data Anomalies," in *39th IEEE Midwest symposium on Circuits and Systems*, , 1996, pp. 593-596 vol.2.
- [124] H. Chia-Chang, L. Hsuan-Yu, and W. Jyh-Horng, "An Adaptive Fuzzy-Logic Variable Forgetting Factor RLS Algorithm," in *62nd IEEE Vehicular Technology Conference (VTC 2005)*, 2005, pp. 1412-1416.
- [125] F. H. Ali and M. M. F. Algreer, "Fuzzy PID Control for Positioning Plants with Uncertain Parameters Variation," in *2nd IEEE Information and Communication Technologies (ICTTA 2006)*, 2006, pp. 1428-1433.
- [126] P. Mattavelli, L. Rossetto, G. Spiazzi, and P. Tenti, "General-Purpose Fuzzy Controller for DC-DC Converters," *IEEE Transactions on Power Electronics*, vol. 12, pp. 79-86, 1997.
- [127] P. Prandoni, M. Vetterli, and E. P. F. de Lausanne, "An FIR Cascade Structure for Adaptive Linear Prediction," *IEEE transactions on signal processing*, vol. 46, pp. 2566-2571, 1998.
- [128] Texas Instruments. Data Manual (June 2007). TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 Digital Signal Controllers (DSCs). [online]. Available: <http://www.ti.com/lit/ds/symlink/tms320f28335.pdf>.
- [129] A. Soukup, "Enabling Greener Embedded Control Systems with Floating-Point DSCs," Texas Instruments. White Paper. (2008). [Online]. Available: <http://www.ti.com/lit/wp/spry113/spry113.pdf>.

-
- [130] S. Choi and M. Saeedifard, "An Educational Laboratory for Digital Control and Rapid Prototyping of Power Electronic Circuits," *IEEE Transactions on Education*, vol. 55, pp. 263-270, 2012.