Université de Cergy-Pontoise

Ecole doctorale Sciences et ingénierie – ED n 417

Spécialité: Génie Electrique et Electronique

Università degli Studi di Salerno

Facoltà di Ingegneria
Dipartimento di Ingegneria dell'Informazione, Ingegneria Elettrica e Matematica Applicata

Dottorato di Ricerca in Ingegneria dell'Informazione XIII Ciclo – Nuova Serie

UNIVERSITÉ FRANCO ITALIENNE | UNIVERSITÀ ITALO FRANCESE

PHD THESIS

FPGA-Based Implementation of Real-Time Identification Procedures for Adaptive Control in Photovoltaic Applications

CANDIDATE : **MATTIA RICCO**

SUPERVISORS: **PROF. ERIC MONMASSON**
PROF. GIOVANNI SPAGNUOLO

COORDINATORS: **PROF. DOMINIQUE LAURENT**
PROF. MAURIZIO LONGO

ACADEMIC YEAR 2014 – 2015

JURY

Rapporteurs

Prof. Gabriel Garcera — Universitat Politecnica de Valencia
Prof. Guillaume Gauteau — Université de Toulouse

Examiners

Prof. François Auger — Université de Nantes

Supervisors

Prof. Eric Monmasson — Université de Cergy Pontoise
Prof. Giovanni Spagnuolo — Università degli Studi di Salerno

Defended on 26 May 2015

To my family and my friends.

Ringraziamenti

Sono tante, forse troppe, le persone che vorrei ringraziare, quindi mi soffermerò su quelle che mi hanno segnato rendendo indimenticabili questi ultimi anni.

Non posso che cominciare con le due persone che hanno reso possibile quest'esperienza internazionale, i miei due supervisori, il Professore Eric Monmasson e il Professore Giovanni Spagnuolo. Sono stati le mie guide e mi hanno incoraggiato nei tanti momenti difficili che ho incontrato. Grazie ai loro consigli e alle loro prediche penso di essere cresciuto tanto sia dal punto di vista scientifico che umano. Posso ritenerli senza alcun dubbio due punti di riferimento e due amici.

Voglio ringraziare tutti i membri dei due laboratori dove ho passato tante giornate: il laboratorio di *Circuiti Elettronici di Potenza* all'Universitá di Salerno e il laboratorio di *Systèmes et Applications des Technologies de l'Information et de l'Energie* (SATIE) - équipe SETE (Systémes d'Energie pour les Transports et l'Environnement) all'Universitá di Cergy-Pontoise.

Vorrei fare un ringraziamento particolare al Professore e amico Giovanni Petrone che durante questi tre anni mi ha dato tanti consigli e suggerimenti. Ringrazio i colleghi francesi (in particolare Yuri, Mohamed, Jawad, Herie e Lahoucine), gli amici conosciuti durante questi anni in Francia (Marie, Boris, Marco, Francesca, Cristian, Abdoul, Emanuela, Claudia, Daniele e tanti altri), tutti i ragazzi del laboratorio in Italia e gli amici incontrati in questi anni di Universitá (in particolare Marco, Simona, Gemine e Simone), con cui ho scambiato tante idee e preoccupazioni. E' stato un onore passare le mie giornate con voi. Un grazie particolare va a Patrizio Manganiello che più di tutti è stato al

mio fianco. Abbiamo passato intere giornate a lavorare insieme e sono sicuro di aver trovato un vero amico per la vita.

Voglio inoltre ringraziare tutta la mia famiglia per il supporto che mi ha dato. In maniera particolare mia madre e la mia compagna, nonché amica, Korinzia, che mi è stata vicina e mi ha supportato e (soprattutto) sopportato in questo periodo. A tutti i miei amici di una vita che hanno saputo distrarmi nei momenti difficili dandomi quella spensieratezza di cui a volte avevo bisogno (Alfredo, Christian, Mirko, Giovanni e Vincenzo).

Voglio infine ringraziare tutte quelle persone che, purtroppo, non ho avuto il modo di ringraziare in queste poche righe. Un grazie a tutti voi che avete reso indimenticabili questi anni.

Mattia Ricco

Abstract

In this thesis two adaptive Maximum Power Point Tracking (MPPT) techniques for PhotoVoltaic (PV) applications, which are based on two different real-time identification procedures are proposed. The algorithms are implemented on the same low-cost Field Programmable Gate Array (FPGA) device in charge of controlling the switching converter that processes the power produced by the PV array.

The Perturb & Observe (P&O) algorithm is the most common MPPT technique. Its efficiency is mainly related to two parameters: the perturbation amplitude and the perturbation period T_p. The optimal values of such parameters depend on the PV array type and on the irradiance and temperature conditions thereof, as well as on the parameters of the power processing circuit. Thus, a method for dynamically adapt the P&O parameters would be very useful for increasing the P&O MPPT performances. Several approaches presented in the current literature are focused on the adaptation of the perturbation amplitude. In this thesis, on the contrary, the on-line optimization of the value of T_p is proposed. The effects of such a parameter on both the tracking speed and the stationary MPPT efficiency are pointed out. Besides, the need for a real-time identification technique for identifying the minimum acceptable value of T_p in the actual PV operating conditions is demonstrated.

Two different identification procedures aimed at developing the aforementioned adaptive MPPT controllers have been studied: the Cross-Correlation Method (CCM) and the Dual Kalman Filter (DKF). The first one belongs to the non-parametric techniques and allows identifying the impulse response and the frequency response of the PV system. Instead, the DKF is a model-based approach which esti-

mates the states and the parameters of the system. One of the aims of this thesis is to demonstrate the usefulness of these identification procedures for the optimization of the PV P&O MPPT performances.

In order to achieve a good trade-off between the desired performances and the cost of the controller, hardware digital solutions, such as FPGA, are adopted. They are able to reduce the execution time by exploiting the intrinsic parallelism of the algorithm to be implemented. Then, in this work, the challenging design of a high performances hardware architecture for the identification algorithms is dealt with. Moreover, the implemented identification techniques are compared in terms of accuracy, identification time and used hardware resources.

Several simulations and experimental tests demonstrate the feasibility of the developed identification procedures. In fact, the proposed adaptive MPPT controllers suitably change in few tens of milliseconds the value of T_p ensuring a stable MPPT behaviour. The developed FPGA-based architectures of both the identification techniques is promising for embedding other functions that are of interest in the field of PV systems, e.g. related to on-line monitoring or diagnostic purposes.

The work has been developed in co-tutorship between the *Systèmes et Applications des Technologies de l'Information et de l'Energie* (SATIE) laboratory in the Université de Cergy-Pontoise (France) and the *Circuiti Elettronici di Potenza* laboratory in the Universitá degli Studi di Salerno (Italy). The work has been supported by the Université Franco-Italienne by means the Vinci project 2013 n. C2-29.

Résumé

Dans le cadre de ce travail de thèse deux algorithmes adaptatifs pour le suivi du Point Maximal de Puissance (MPPT) sont proposés pour des applications photovoltaïques. Ces algorithmes sont basés sur deux procédures d'identification en temps réel. Ces procédures sont implantées sur la même plateforme FPGA (Field Programmable Gate Array) qui contrôle le convertisseur continu/continu qui assure le transit de la puissance produite par la source PV.

Parmi toutes les méthodes MPPT, la technique la plus couramment utilisée est l'algorithme Perturbe & Observe (P&O). Son efficacité est principalement liée à deux paramètres : l'amplitude de la perturbation et le temps d'application de la perturbation. Les valeurs optimales de ces paramètres dépendent du type de panneau et des conditions d'irradiation et de température, ainsi que des paramètres du circuit de puissance. Afin d'améliorer les performances de l'algorithme MPPT, différents algorithmes d'adaptions du niveau de variation ont été proposés en littérature. Dans ce travail de thèse, en revanche, l'optimisation du temps d'application de la perturbation T_p est abordée. Les effets de ce paramètre, tant sur la vitesse de suivi que sur l'efficacité énergétique en régime permanent de l'algorithme MPPT, sont montrés. De plus, la nécessité d'une technique d'identification en temps réel pour identifier la valeur minimale acceptable pour T_p est également justifiée.

Deux procédures d'identification différentes sont employées pour obtenir les susmentionnés contrôleurs adaptatifs MPPT : la méthode de corrélation croisée (CCM) et le filtre de Kalman (DKF). La première méthode est une technique non paramétrique qui permet d'identifier la réponse impulsionnelle et la réponse fréquentielle du système photo-

voltaïque. En revanche, le DKF est une approche basée sur un modèle qui estime les états et les paramètres du système. Un des objectifs de cette thèse a été de démontrer l'utilité de ces procédures d'identification pour optimiser les performances de l'algorithme MPPT en temps réel.

Dans le but d'obtenir un bon compromis entre les performances désirées et le coût du contrôleur, des solutions numériques matérielles, telles que les FPGA, sont adoptées. Ces dispositifs sont capables de réduire le temps d'exécution en exploitant le parallélisme inhérent de l'algorithme à implanter. Ainsi, la conception d'architectures matérielles à hautes performances pour algorithmes d'identification est abordée. Ensuite, les techniques implantées sont comparées en termes de précision, de temps d'identification et de ressources matérielles consommées.

Le bon fonctionnement des deux méthodes d'identification proposées est démontré tant en simulation qu'au moyen de tests expérimentaux. En effet, les contrôleurs MPPT adaptatifs proposés changent de façon appropriée et en quelques dizaines de millisecondes la valeur de T_p, assurant ainsi un comportement stable des cycles MPPT. Les architectures développées concernant les deux techniques d'identification peuvent être aussi utilisées pour implanter d'autres fonctionnalités dans les systèmes PV, telles que le monitorage en ligne ou le diagnostic.

Ce travail a été développé en cotutelle entre le laboratoire de *Systèmes et Applications des Technologies de l'Information et de l'Energie* (SATIE) de l'Université de Cergy-Pontoise (France) et le laboratoire de *Circuiti Elettronici di Potenza* de l'Université de Salerno (Italie). Le travail a été supporté par l'Université Franco-Italienne au travers du projet Vinci 2013 numéro C2-29.

Abstract

In questa tesi vengono proposti due algoritmi adattativi per l'Inseguimento del Punto di Massima Potenza (MPPT) in applicazioni FotoVoltaiche (PV) basati su due distinte procedure di identificazione in tempo reale. Gli algoritmi sviluppati sono implementati sullo stesso dispositivo FPGA (Field Programmable Gate Array) incaricato di controllare il convertitore dc/dc che processa la potenza prodotta dal campo PV.

La tecnica MPPT più comunemente utilizzata è l'algoritmo Perturba & Osserva (P&O). La sua efficienza è principalmente legata a due parametri: l'ampiezza della perturbazione ed il tempo di perturbazione T_p. I valori ottimali di questi parametri dipendono dal tipo di pannello e dalle condizioni di irraggiamento e temperatura, così come dai parametri del circuito di potenza. Per questo motivo, allo scopo di migliorare le prestazioni dell'algoritmo MPPT, è necessario adattare tali parametri durante il funzionamento del sistema. Al contrario dei lavori già proposti in letteratura, dove il parametro adattato è l'ampiezza di perturbazione, in questa tesi viene affrontata l'ottimizzazione del valore di T_p. Vengono mostrati gli effetti di tale parametro sia sulla velocità di inseguimento che sull'efficienza stazionaria dell'algoritmo MPPT. E' inoltre dimostrata la necessità di una tecnica di identificazione che lavori in tempo reale per identificare il minimo valore accettabile per T_p nelle attuali condizioni operative.

Due differenti procedure di identificazione sono considerate per sviluppare i predetti controllori adattativi MPPT: il Metodo della Mutua Correlazione (CCM) ed il Filtro Duale di Kalman (DKF). Il primo ricade tra le tecniche non parametriche e permette di identificare la risposta impulsiva e la risposta in frequenza del sistema PV. Invece, il

DKF è un approccio basato su modello che stima gli stati e i parametri del sistema. Uno degli obiettivi di questa tesi è quello di dimostrare l'utilità di queste procedure di identificazione per l'ottimizzazione delle performance dell'algoritmo P&O.

Al fine di ottenere un buon compromesso tra le performance desiderate ed il costo del controllore, sono adottate soluzioni hardware digitali, come l'FPGA. Questi dispositivi sono capaci di ridurre il tempo di esecuzione sfruttando il parallelismo intrinseco dell'algoritmo da implementare. Quindi, in questo lavoro di tesi, è affrontato il progetto di architetture hardware ad alte prestazioni per algoritmi di identificazione. Inoltre, le tecniche implementate sono confrontate in termini di precisione, di tempi di identificazione e di risorse hardware utilizzate.

Il corretto funzionamento dei due metodi di identificazione proposti è dimostrato sia in simulazione che attraverso test sperimentali. Infatti, viene mostrato come i controllori adattativi MPPT proposti cambino in maniera opportuna ed in tempi brevi il valore di T_p assicurando un comportamento stabile dell'MPPT. Gli algoritmi di identificazione implementati possono anche essere utilizzati per integrare altre funzioni nei sistemi PV, come ad esempio il monitoraggio in linea o la diagnostica.

Il presente lavoro è stato sviluppato in cotutela tra il laboratorio di *Systèmes et Applications des Technologies de l'Information et de l'Energie* (SATIE) dell'Università di Cergy-Pontoise (Francia) e il laboratorio di *Circuiti Elettronici di Potenza* dell'Università degli Studi di Salerno (Italia). Il lavoro è stato finanziato dall'Università Franco-Italiana attraverso il Bando Vinci del 2013 progetto n. C2-29.

Contents

List of Figures

List of Tables

General Introduction

In the last years, the high demand for electrical energy, the need for reduction of greenhouse gases, as the carbon dioxide, as well as the limited capacity of fossil fuel reserves have increased the interest in renewable energies. The photovoltaic solar, wind, biomass and geothermal energies are the most important renewable sources on which academic and industrial specialists are working in these decades. In particular, PhotoVoltaic (PV) systems are increasing their diffusion thanks to the accessibility and availability of solar energy and also because of their small environmental impact.

The electrical power vs. voltage curve of any PV array exhibits one Maximum Power Point (MPP). The voltage value at which it occurs changes according to temperature and irradiance. Therefore, in order to continuously track the MPP, the adoption of Maximum Power Point Tracking (MPPT) controllers becomes mandatory. The Perturb & Observe (P&O) MPPT technique is very common because of its robustness and simplicity. Actual literature demonstrates that the P&O tracking performance depends on the operating conditions, of both the PV array and the switching converter that actuates the MPPT algorithm. In turn, the operating conditions of the whole system depend on the irradiance and temperature, as well as on the PV cells type and technology. An important role is also played by the effects of aging, causing drifts on the parameters values of the PV array and of the switching converter.

As a consequence of this variability affecting the PV source and of the related power processing system, the P&O MPPT performances might be very good in some conditions and poor in some others. Thus, a method for dynamically adapt the P&O parameters to the actual sys-

tem operating conditions would be very beneficial.

In literature, several authors focus their efforts on the optimization of one of the two P&O operating parameters, that is the perturbation amplitude. This thesis work is focused on the on-line optimization of the other P&O parameter, i.e. the perturbation period T_p. The effects of such a parameter on both the tracking speed and the stationary MPPT efficiency are firstly pointed out. Afterwards, the way in which the optimal value of T_p depends on the PV array type and on the irradiance and temperature conditions thereof, as well as on the parameters of the power processing circuit, is described. As a consequence, the need for a real-time identification procedure for identifying the minimum acceptable value that the perturbation period can assume for the actual PV operating conditions has been put into evidence in this work.

The implementation of such new functionalities requires suitable digital platforms. Indeed, identification algorithms are very time consuming so that to achieve a good trade-off between the desired performances and the cost of the controller is a challenging task. Nowadays, digital software MPPT solutions, e.g. microcontrollers or digital signal processor, are usually preferred to analogue ones. However, in such devices the architecture is fixed and then the treatment is serialized. As a consequence, the execution time of the identification algorithm is not optimized. For this reason, hardware solutions are preferred to software ones. Many studies have confirmed that the Field Programmable Gate Array (FPGA) technology is a good candidate when high speed performances are required. Indeed, thanks to the exploitation of the intrinsic parallelism of the algorithm to be implemented, a significant reduction of the execution time is achieved. This work affords the challenging design of a high performances hardware architecture for the identification algorithms required by the proposed P&O MPPT controllers. The limited resources offered by low cost target devices, the architecture constraints, and the need of computation speed performances required by the MPPT application need to be kept into account at the same time.

Thesis Objectives and Contributions

This thesis is aimed at demonstrating the usefulness of the identification processes for the optimization of the PV P&O MPPT performances. Moreover, it is demonstrated that FPGA devices are the best candidates for such applications. In particular, FPGA-based implementations of an adaptive MPPT controller have been designed, taking advantage of two different identification techniques. The studied and implemented identification techniques are the Cross-Correlation Method (CCM) and the Dual Kalman Filter (DKF). The first technique is a non-parametric approach that allows identifying the system frequency response for a given range of frequencies. Instead, the DKF is a model-based technique that estimates the states and parameters of the system. In this work they are used for evaluating the settling time of the system in order to achieve the on-line optimization of the MPPT perturbation period T_p. In the final part of the thesis it is also pointed out the potential usefulness of the same identification techniques for other functions to embed in the PV system, e.g. related to diagnostic purposes.

In the following, the author's contributions and the thesis objectives are detailed.

- First of all, the effects of the value assumed by the perturbation period T_p to the steady state and dynamical MPPT performances of the P&O algorithm have been pointed out. Thus, it has been demonstrated that an on-line optimization of the T_p value by means of a real-time identification technique can improve the P&O tracking performances. An in-depth analysis of the non-linearities of the PV system has also been done in order to evaluate their effects upon the identification process.

- A CCM-based adaptive P&O MPPT controller has been developed. The optimal value of the T_p has been evaluated on the basis of the PV system frequency response identified by means of the CCM.

- A DKF-based adaptive P&O MPPT controller has been developed. The evaluation of the proper perturbation period has been

achieved by the estimation of the PV parameters. Before developing the model-based identification technique, different PV models have been considered and the need of a simple model has been emphasized. In order to meet this requirement and to achieve reliable identification results, the small-signal PV model has been chosen. A convergence criterion aimed at putting the DKF in stand-by conditions has been also proposed.

- The proposed adaptive CCM and DKF based MPPT controllers have been both implemented in a low-cost FPGA device. The required timing and area constraints have been duly fulfilled. For this purpose several algorithm and architecture optimizations have been developed.
- A comparison between the two identification procedures in the MPPT PV application in terms of accuracy, identification time and consumed hardware resources has been performed.
- Experimental validations of the developed FPGA-based adaptive MPPT controllers has been done.

Thesis Outline

The outline of the thesis is given in the following.

The first chapter gives an overview of the linear identification techniques. The non-parametric and model-based approaches are described and the main techniques are presented. Their main advantages and drawbacks are pointed out and the main applications are reviewed.

Chapter 2 is aimed at introducing a new approach to the dynamical optimization of the P&O MPPT perturbation period. It is shown that a proper choice of the T_p value improves both the dynamic and the stationary MPPT efficiency. Moreover, it is shown that T_p cannot be directly evaluated in the time-domain and then the usefulness of a real-time identification technique is justified. Moreover, the principles of both the CCM and DKF are recalled and their use in PV applications

is shown. Finally, the FPGA design methodology for the both identification techniques is given and a preliminary system specification is provided.

Chapter 3 is dedicated to explaining the algorithm development. According to the adopted design methodology, the modular partitioning of both techniques is made and the algorithm digital realization is illustrated. Some aspects of the algorithm are optimized for increasing the accuracy of the identification or for reducing the algorithm complexity. Finally, the developed algorithms are validated through simulation tests in Matlab/Simulink and their comparison in terms of accuracy and complexity is provided.

Chapter 4 deals with the FPGA architecture development of the proposed adaptive MPPT controllers. A pre evaluation of the time/area performances is made and then some architecture optimizations are carried out where needed. Once all constraints are fulfilled, the VHDL coding and a time/area analysis are made. In this chapter a comparison between the two proposed identification techniques in terms of execution time and consumed hardware resources is also given.

In Chapter 5, the experimental validation of the designed hardware architectures is presented. The experimental platform is firstly illustrated. After that, the adaptive MPPT controllers based on both the CCM and the DKF are validated and the experimental results are presented and discussed.

Acronyms

A^3 : Algorithm Architecture Adequation
ADC : Analogue Digital Converter
AIC : Akaike's Information Criterion
ASIC : Application Specific Integrated Circuit
BF : ButterFly
CAD : Computer Aided Design
CCM : Cross-Correlation Method
CLB : Configurable Logic Block
DFG : Data Flow Graph
DFT : Discrete Fourier Transform
DKF : Dual Kalman Filter
DPWM : Digital Pulse Width Modulator
DSP : Digital Signal Processor
EDA : Electronic Design Automation
EIS : Electrochemical Impedance Spectroscopy
EKF : Extended Kalman Filter
FDFG : Factorized Data Flow Graph
FFT : Fast Fourier Transform
FPGA : Field Programmable Gate Array
FSM : Finite State Machine
FSSDM : Further Simplified Single Diode Model
FWHT : Fast Walsh-Hadamard Transform
HIL : Hardware In the Loop
IC : Integrated Circuit
IP : Intellectual Property
ISE : Integrated Software Environment
IUKF : Iterated Unscented Kalman Filter
IV : Instrumental-Variable
I/O : Input/Output
JKF : Joint Kalman Filter
JM : Jacobian Matrix
KF : Kalman Filter
KG : Kalman Gain
LAB : Logical Array Block
LE : Logic Element
LKF : Linear Kalman Filter
LS : Least Square
LSR : Least Square Regression
LTI : Linear Time Invariant
LUT : Look-Up-Table

MACC : Multiply ACCumulator
MAP : Maximum A Posteriori
ML : Maximum Likelihood
MLS : Maximum Length Sequence
MMCM : Mixed Mode Clock Manager
MPP : Maximum Power Point
MPPT : Maximum Power Point Tracking
MSPS : Mega Sample Per Second
NRMSE : Normalized Root Mean Square Error
OCF : Observable Canonical Form
PCF : Polynomial Curve Fitting
PEM : Prediction Error Method
P&O : Perturb & Observe
PLL : Phase-Locked-Loop
PR : Pipeline Register
PRBS : Pseudo Random Binary Sequence
PV : PhotoVoltaic
RKM : Runge-Kutta Method
SoC : System-on-Chip
SM : Synchronous Motor
SMPS : Switched Mode Power Supply
SNR : Signal Noise Ratio
SR : Shift Register
SS : State Space
SSDM : Simplified Single Diode Model
STC : Standard Test Condition
VHDL : VHSIC Hardware Description Language
VHSIC : Very High Speed Integrated Circuit

Chapter 1

Identification Procedures

1.1 Introduction

Recently the interest in techniques aimed at identifying some features of an unknown system has grown in many different contexts, such as biology, econometrics, physics and engineering [2]. A wide range of applications can be found in literature. For instance, a number of identification techniques has been proposed in [3]-[4]-[5] to perform model based control in building applications with the aim of increasing the energy efficiency. Identification techniques are also employed in robotics for hurdles or terrain identification [6], whereas in electrical engineering they are adopted for fault detection [7], pattern recognition [8] or in order to implement adaptive control algorithms [9]. In the control systems field, identification methods are used to synthesize regulators [10], to carry out real-time predictive control [11] or to design a monitoring algorithm. Besides, they are commonly applied to the monitoring of electrical motors for fault diagnosis purposes [12] and for the implementation of sensor-less controllers [13]-[14]. Identification techniques have successfully been adopted also in Switched Mode Power Supply (SMPS) design and analysis. For example, in [15]-[16], the identification procedure is aimed at monitoring and diagnosing the power system and to perform adaptive or non-linear control.

Due to its complexity, the digital implementation of such identification techniques is mandatory. At the price of a lower efficiency and limited control bandwidth with respect to analog controllers, embedded systems allow to integrate the identification and the control algorithms taking benefits from re-programmability, ability to execute complex algorithms, better noise immunity, higher communication capability, cost reduction and improved system level integration [17]. The availability of digital devices for the implementation of system identification and control techniques, also opens the field for embedding diagnostic techniques. On-line monitoring evaluates system performance, helps to identify potential component degradations and helps detecting the potential faults in the system. Such information allows making diagnosis for determining the causes of degradation or fault. Hence, an increment of the system reliability can be reached by identifying, localizing the fault, and reducing the inactivity periods. Furthermore, the identification techniques allow characterizing the system in real-time. This helps to tune adequately the digital controller, and as a consequence keeping the desired system performances with respect to variation of component values or operating conditions. The success of these functionalities is based on the validity range and on the goodness of the identification technique, as well as on the promptness in identifying system variations. Moreover, the intrusiveness degree, the efforts required for the implementation of the identification methods and the cost in terms of used resources are other key aspects in these techniques.

The identification algorithms are the cornerstone to achieve all the aforementioned advantages and they requires to build a mathematical model and then infer on it starting from experimental data [2]. The mathematical model has to be able to represent the corresponding dynamic system with the highest possible level of accuracy. A conceptual scheme of an Input/Output (I/O) system is given in Figure 1.1: $u \in \mathbf{R}^m$ represents the input signals and $y \in \mathbf{R}^n$ is the output signals, i.e. the observable signals of interest. The remaining input signals v, that cannot be controlled by the operator nor always be measurable, are considered as disturbances.

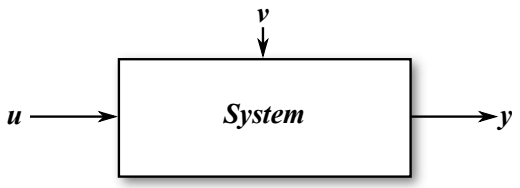


Figure 1.1 Conceptual Scheme of a I/O system.

Regardless of the fact that the identification algorithm is implemented directly on the device in charge of performing the data acquisition and the system control, or on another device where the acquired data are downloaded for *off-line* processing, the identification process can always be decomposed in the following way: *(i)* acquisition of input and output data; *(ii)* choice of a candidate model structure; *(iii)* estimation of the model parameters through a pre-specified rule; *(iv)* validation of the identified model. The output data set can be acquired by disconnecting the system and injecting proper set of input signals (*off-line identification*). This approach is very intrusive and, for this reason, it is not used for embedded applications. On the other hand, *on-line identification* techniques allow performing the identification algorithm during the normal operation of the system despite a poorer identification accuracy or a higher identification time. However, in some cases, a proper input sequence has to be added to the input signals in order to obtain more information regarding the system. In this case the input data set assumes a key role, being appointed to stimulate the system dynamics of interest. In literature several different input signals are employed: impulse function, step function, sine function, sum of sinusoids, and so on [18]. The choice of the optimal excitation waveform depends on the application, on the system characteristics of interest and on the identification method.

The definition of a set of candidate system models is another key factor in the identification procedure, because it might make the identification algorithm less effective. Indeed, an inefficient model can cause wrong identification results, a higher algorithm complexity or a larger identification time. A system model based on known physical laws is called *white box model*. Sometimes such a model can be excessively complex leading to a very time consuming solution, not use-

ful for practical applications. On the other hand if only the structure model, and at most some parameters, are known, the model is called *grey box model*. In this case, thanks to an identification technique, the unknown parameters can be estimated. An overview of these models are given in [2]. When no information about the system is known, it can only be represented through a *black box model*: in this case, not only the system parameters, but also the physical laws governing the system, are unknown. Generally, identification techniques operate on these two last types of models. After having chosen the model structure, an identification rule has to be adopted in order to find the best set of parameters. Finally, the identified model has to be validated.

The choice of the system model to be adopted for the identification procedure also depends on the intrinsic characteristics of the system. If, as it happens in a wide range of applications, the unknown system is non-linear, the model can be linearized for applying a *linear identification procedure*. Otherwise, a *non-linear identification method*, e.g. neural networks or fuzzy sets [19], has to be applied. In this work only linear identification techniques are considered.

In the following, an overview of the so-called *Non-parametric Identification Procedures* aimed at identifying linear time-invariant systems through the impulse response or the transfer function is given. In such cases, a definition of a candidate model set is not required and only an appropriate choice of the stimulus signal is needed. Then, *Parametric Identification Procedures* requiring the knowledge of the system structure and relying on the minimization of a cost function to estimate the model parameters are reviewed.

1.2 Non-Parametric Methods

These approaches require only the knowledge of input and output data and no additional information about the system structure is needed. Hence, such procedures rely on black box models and the choice of the stimulus signal is the most important aspect. The non-parametric approaches peculiarity is that the identified model is described by a curve or a function. In literature, several options relying on different stimulus signals are reported: some of them are time-domain techniques, achieving system identification through analysis of impulse or step response of the system; instead, frequency-domain techniques rely on the system frequency response identification [2]. In Figure 1.2 a general block diagram referring to the non-parametric identification procedure is shown. $p(t)$ is the stimulus signal, required by the adopted non-parametric identification method. The identified sequence $h(t)$ is the identified system response: it can be the impulse response, the step response or the frequency response according with the chosen method. The most common non-parametric methods are now reviewed, underlining their area of interest, their advantages and drawbacks.

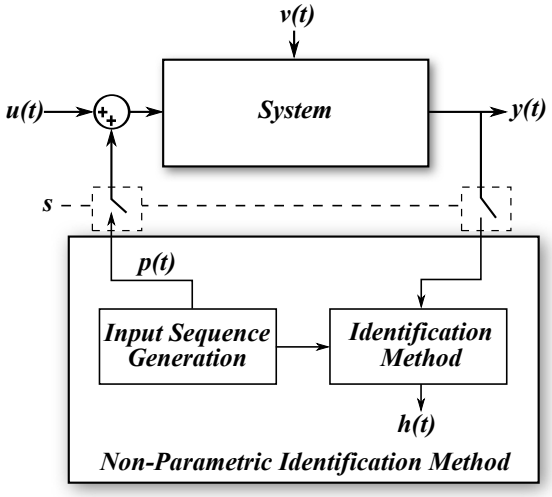


Figure 1.2 Conceptual Scheme of Non-Parametric Identification Methods.

The most simple time-domain technique is the *transient response analysis*. The impulse response and step response analysis both belong to this class: they are obtained by measuring the system output by injecting a pulse or a step signal, respectively, at the input. The transient analysis can be used when only basic characteristics have to be estimated, such as static gain or the predominant time constant. Simplicity is an obvious advantage of these techniques, although it comes at the cost of a high susceptibility to the disturbances present in the system. In fact, large errors on the identified response are found in most practical cases, making this identification procedure not largely used. Nevertheless, a large input signal amplitude provides acceptable results also in the whole response identification. In [20] a step analysis is performed for identifying the impulse response of buck and boost DC/DC converters: good results are reached by using a significant (40%) step amplitude. Unfortunately, especially for non-linear systems, large input variations cannot be applied if a local linearised analysis is required.

Electrochemical Impedance Spectroscopy (EIS) is another widely used non-parametric method. It allows to estimate the impedance of a system in the frequency range of interest. It consists in injecting a sinusoidal perturbation signal, with proper amplitude and phase, at the desired frequency ω_i, and measuring the corresponding output frequency component at the same frequency. The impedance magnitude at that frequency is computed as ratio between the amplitudes of the output and input components, whereas the phase difference is calculated by a phase detector. By performing this procedure at all the frequencies of interest, the Bode Diagram of the system frequency response is obtained. EIS represents a very simple identification method, largely used in Fuel Cell and battery applications. For example, in [21] the EIS is used to collect the impedance data of a proton exchange membrane fuel cell stack. The authors in [22], instead, use this identification method to analyze a Li-ion battery. Its simplicity and the ease in generating the input signal make this technique very attractive. Unfortunately, the time required to perform the identification might be too long for some real-time applications. Indeed, transients due to the application of the sinusoidal perturbations at different frequencies must

be accounted for, especially when the system behaviour in very low frequency ranges is investigated.

Alternatively, a multi-frequency signal composed of several tones at different frequencies can be used as stimulus in order to shorten the identification time. In order to ensure the periodicity of $p(t)$, all the frequency components ω_i must be multiples of the lowest frequency. Moreover, the phase choice of the input signal has to be able to normalize the total signal amplitude in order to avoid the injection of high input values in the system [23]. In [24] a multi-frequency EIS is applied to a PEM fuel cell for analyzing possible humidity-related instabilities. The differences with the single-frequency EIS in a non-stationary system are put into evidence and the shortened identification time, achieved by using the multi-frequency EIS, is documented. It is worth noting that the multi-frequency approach requires a more complex algorithm for Fourier analysis compared to basic EIS procedure.

The *Cross-Correlation Method* is a non-parametric approach allowing to identify the system impulse response by injecting white noise in the system and performing the cross correlation between the output sequence and the white noise input. The cross correlation method does not require a high amplitude of the input stimulus and it is intrinsically robust against disturbances, because the white noise input and the disturbances are uncorrelated. The aforementioned advantages come at the price of higher computational complexity, so that a powerful digital platform is required. This method is largely applied, e.g. in [25], for identifying the frequency response of Switching Power Converters.

Finally, the choice of the input perturbation is a key aspect of non-parametric techniques, which do not require any preliminary knowledge of the system. Concerning monitoring and diagnosis procedures, variations of the system response could be sufficient for realizing if something is changed in the system. On the other hand, the adaptive control requires the estimation of some parameters with the help of the identified responses. Therefore, in this last case, other efforts have to be made when non-parametric identification techniques are used (see Chapter 2 Section 2.2.2).

1.3 Parametric Methods

The Parametric Methods allow to determine a supposed dynamic model from experimental data. They consist in minimizing (maximizing) a cost function by using an optimization algorithm. They require the prior knowledge of the model structure Π parametrized using the unknown parameter vector $\vartheta \in D_\Pi \subset \mathbf{R}^s$. The model set can be defined as in (1.1).

$$\overline{\Pi} = \{ \Pi(\vartheta) \,|\, \vartheta \in D_\Pi \} \tag{1.1}$$

where $\overline{\Pi}$ is the whole set of models with structure Π and a given parameter vector ϑ. The prediction error is defined as difference between the system output $y(t)$ and the model output $\widehat{y}(t)$ obtained by using an assigned parameter vector ϑ^*, as shown in (1.2).

$$\xi(t,\vartheta^*) = y(t) - \widehat{y}(t,\vartheta^*) \tag{1.2}$$

The aim of the parametric methods is to find the parameter vector ϑ^* that minimizes the prediction error ξ. The goodness in the identification of the parameter vector is evaluated according to the following definitions [18].

Definition 1.3.1. The identified parameter vector $\widehat{\vartheta}$ is *un-biased* if:

$$E[\widehat{\vartheta}] = \vartheta \tag{1.3}$$

where $E[\widehat{\vartheta}]$ is the expected value of the identified parameter vector and ϑ is the "true" value (because the chosen model structure is always an approximation of the reality).

Definition 1.3.2. The $\widehat{\vartheta}$ is *consistent* if:

$$\widehat{\vartheta} \to \vartheta \;\; when \;\; N \to \infty \tag{1.4}$$

where N is the number of samples.

These identification methods can be classified as non recursive and recursive procedures. In the following both are illustrated with their advantages and disadvantages. Moreover, the main differences are displayed and some considerations are given.

1.3.1 Non Recursive Procedures

Two types of non recursive parametric approaches can be found in literature [2]. One uses a kind of norm or criterion function of ξ for measuring the accuracy in the model identification. These methods fall into the *prediction error methods* (PEMs). The other approach searches the parameter vector that gives the prediction error uncorrelated with a given sequence. The *Instrumental-Variable Methods* (IV) belong to this latter class.

PEMs use an optimization algorithm in order to search for a parameter vector ϑ^* minimizing a cost function $l(\cdot)$. The latter depends on a filtered version of the prediction error as shown in 1.5.

$$\widehat{\vartheta} = \arg\min_{\vartheta \in D_\Pi} \frac{1}{N} \sum_{t=1}^{N} l(F(z)\xi(t,\vartheta)) \tag{1.5}$$

where $F(z)$ is a linear filter. A number of minimization algorithms, cost functions and filtering methods are presented in literature. The *Least-Squares* (LS) method is the most common optimization algorithm using a least-squares criterion to find the parameter vector ϑ without adopting any filtering of the prediction error. For example, in [26] the method has been used for parameter estimation of controlled autoregressive moving average systems. A drawback in this identification method is the computational effort needed for the matrix inversion required by the method itself. The partitioned matrix inversion lemma allows to reduce the computation effort [27]. The fact that $\widehat{\vartheta}$ is consistent only under restrictive conditions is another disadvantage [18]. In order to overcome these limits, different LS methods have been proposed. Other techniques falls into the PEM category as the *Maximum Likelihood* (ML), *Maximum a Posteriori* (MAP) and *Akaike's Information Criterion* (AIC) method. A block diagram, which resumes the PEM approach, is shown in Figure 1.3. The input and output signal are recorded only when the identification procedure is enabled. Once the data set has been collected, an initial parameter vector is considered and the recorded input sequence is sent to the off-line model. The prediction error is calculated by (1.2), filtered and sent to the minimization algorithm. It calculates a new parameter vector which will

change the off-line model. If the new prediction error is not satisfactory the procedure has to be repeated leading to an intrinsic iterative character of this procedure.

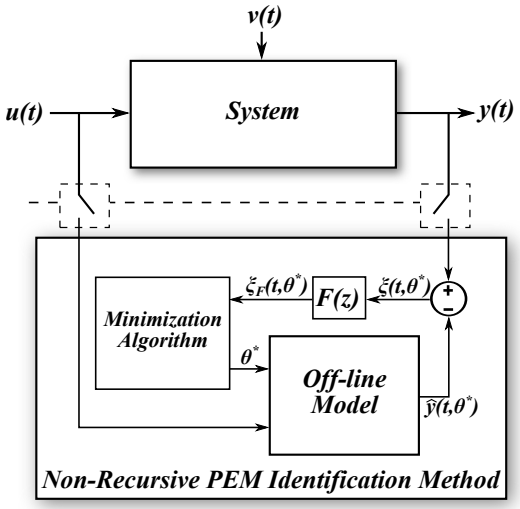


Figure 1.3 Conceptual Scheme of a Prediction Error Method.

The Instrumental Variable approach [18] exploits the uncorrelation property between the input sequence and the disturbance $v(t)$ to the aim of identifying the parameter vector ϑ. It is widely adopted in econometrics and in control engineering. For instance, in [28] this approach is adopted for frequency-domain system identification in industrial motion systems. Figure 1.4 shows the conceptual block diagram of the IV approach. As in PEM approach, the input and output data set is acquired firstly and then the identification method is executed. The IV methods do not minimize a cost function but they are aimed at reducing the correlation between an instrumental variable ζ and the prediction error. The accuracy of the parameter estimation is strictly related to the chosen ζ and then particular attention should be paid in its selection. Indeed, "weak" instrumental variable could involve poor prediction. Moreover, the estimation becomes inconsistent when ζ is correlated with the disturbance [29].

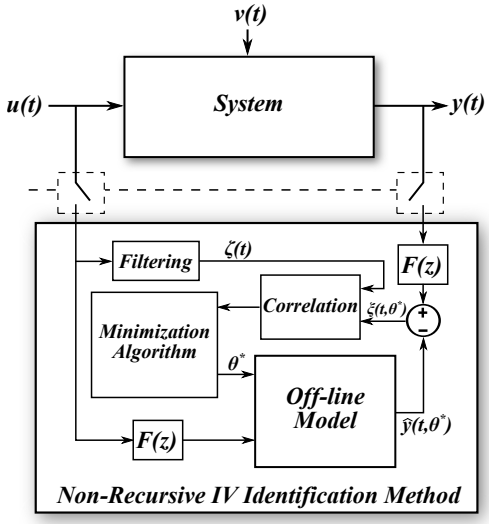


Figure 1.4 Conceptual Scheme of a Instrumental Variable approach.

1.3.2 Recursive Procedures

The *non-recursive* identification methods employ the recorded data to identify the parameter vector and then the system model. *Recursive* identification procedures, instead, use the samples just acquired and are adopted in order to track time-varying parameters. Such methods adapt an on-line model during normal system operation. The main advantage is that this approach works continuously by giving, sample by sample, the most appropriate estimated parameter vector.

The recursive procedures require that the identification algorithm be achieved in one sample, so that the adoption of a powerful digital platform is mandatory. The *recursive least square* (RLS) method [13] is used to identify unknown motor parameters by using voltage and current measurements. The *recursive instrumental variable* method is also very frequent in the recent literature.

Recursive identification algorithm is based on (1.6): the new set of parameters depends only on the old estimation and on the new measurements. This allows a strong reduction of the needed memory in comparison with the non-recursive approaches. The new input and output data are used for calculating the new prediction error, suitably

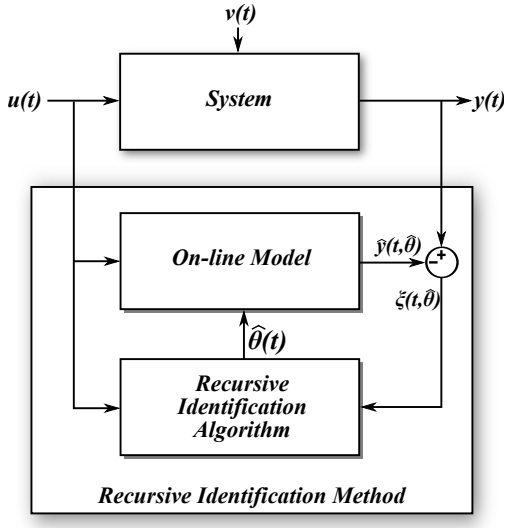


Figure 1.5 Conceptual Scheme of a Recursive Parametric Identification Precedure.

weighted by a variable gain $L(t)$. The digital identification algorithm has to run quickly enough in computing $\widehat{\vartheta}(t)$ before the new measurements are available.

$$\widehat{\vartheta}(t) = \widehat{\vartheta}(t-1) + L(t) \cdot \xi(t, \widehat{\vartheta}(t-1)) \tag{1.6}$$

The *Kalman Filter* (KF) also belongs to the recursive model-based identification method group. It is an optimal state estimator of dynamic systems, presented for the first time by R. Kalman [30]. Its formulation is elegant and quite simple to implement in digital platforms. In addition to the canonical form, new versions of the Kalman Filter have been proposed to estimate also unknown system parameters. It provides also an error bound of the estimation, which is useful in monitoring or diagnosis approaches. It plays a key role in many industrial applications, such as: sensorless control, diagnosis and fault-tolerant control of ac drives; distributed generation and storage systems; robotics, vision and sensor fusion techniques; applications in signal processing and instrumentation; real-time implementation for industrial control systems and so on [31].

Different algorithms based on the Kalman Filter have been pro-

posed in literature. The *Linear Kalman Filter* (LKF) provides an optimal estimate of the state vector for linear system identification. In non-linear cases, the *Extended Kalman filter* (EKF) has to be applied. In [14] the EKF has been employed to estimate the rotor position and speed of a Synchronous Motor (SM) eliminating the mechanical sensors. Both LKF and EKF allow to identify only the state vector of the system. Parametric estimation can be carried out by using the *Joint Kalman Filter* (JKF), thus adding the desired parameters to the state vector. This approach increases the degree of the filter, thus leading to a significant increase of the computational complexity. On the other hand, the *Dual Kalman Filter* (DKF) runs two KF in parallel: one for state estimation and the other one for parameter estimation. In this case the state vector remains unchanged and the filter degree is the higher size between the state and the parameter vector. DKF is used, for example, in battery management systems [32]. In [33] an *Adaptive Kalman Filter* is used for dynamic harmonic state estimation and harmonic injection tracking. Two different models of the noise covariance matrix have been considered depending on the system conditions: steady-state or transient. Therefore, in steady-state conditions the corresponding noise covariance matrix is used in the KF, otherwise the other one is adopted.

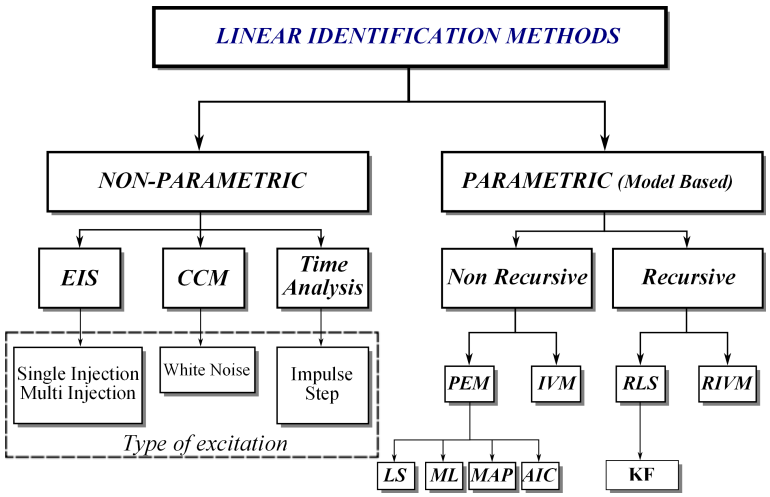


Figure 1.6 Scheme of the Identification Methods.

Figure 1.6 summarizes the most common linear identification methods emphasizing the need of a correct input signal for the non-parametric identification methods and a proper system model for the parametric procedures.

With this overview and after having framed in the next chapter the application treated in this work, it will be possible to understand the motivations for choosing the two adopted identification methods.

Chapter 2

Design Methodology for On-line Identification Algorithms in PV systems

The identification techniques overviewed in Chapter (1) can find many applications also in PV systems. For instance, they are useful for estimating the model parameters or the best operating point. In [34] the resistive, inductive and capacitive parameters of the PV dynamic model are identified. The procedure consists in finding an operating point where the step load transition is characterized by two non-interacting poles. Under this condition the first part of the step current response is due to the inductive contribution and the last part of the capacitive one. Then, the interested parameters can be identified, with only one measurement, using a Least-Squares Regression (LSR) method. The main drawbacks of the proposed approach are the need of a short sampling period and that a part of the procedure is executed off-line. Moreover, the operating point might not be the best for the PV source. In [35], some of the main PV cells parameters values are identified. The procedure requires the irradiation level, the temperature of the photovoltaic array and only one voltage and current measurement. The aim in [36] is to identify the condition of the PV array (normal condition, temporary partial shade and permanent partial shade or malfunction). It uses the Kalman Filter in order to estimate the nominal optimal oper-

ating voltage and a procedure determines the actual PV array condition which requires a measurement of the PV array operating temperature. The authors in [37] propose a real-time estimation of the optimal operating point by using a Polynomial Curve Fitting (PCF) and Newton-Raphson method. Three models have been compared: the Simplified Single Diode Model (SSDM), the Further SSDM (FSSDM) and the PCF, in terms of accuracy and convergence. In [38] a PV model estimation method from the PV module data has been presented. The physical parameters have been identified using the physics equations and the Gauss-Seidel method, being an iterative method for solving transcendental equations. The parameter dependency on temperature and irradiance has been studied and the array optimal operating point has been estimated from the identified parameters. The authors in [39] use the PV manufacturer data curves to estimate the PV parameters and the Iterated Unscented Kalman Filter (IUKF) to identify its best operating point. In [40] a new maximum power point tracking method using the Kalman Filter has been presented and compared with a more classical approach.

One of the objectives of this work is to develop an adaptive PV maximum power point tracking algorithm that is able to face the irradiation and temperature variations as well as the ageing of the components. To this aim, two different identification approach are adopted: the cross-correlation method and the dual kalman filter.

This chapter is organized as follows. At first, a brief presentation of the PV system is made. The importance of using a maximum power point tracker is explained and the need of optimizing the parameters of this algorithm is pointed out. After that, the necessity of on-line identification procedure is stressed and the two identification techniques are presented and their main differences discussed. Finally, an overview of FPGA technology is given and the adopted implementation methodology is illustrated.

2.1 Photovoltaic system

In Figure 2.1, a typical PV system is shown: it consists of a PV source connected to a dc-dc switching converter supplying energy to a DC bus, at which an inverter is connected or a battery. The main function of the converter is to track the time varying MPP.

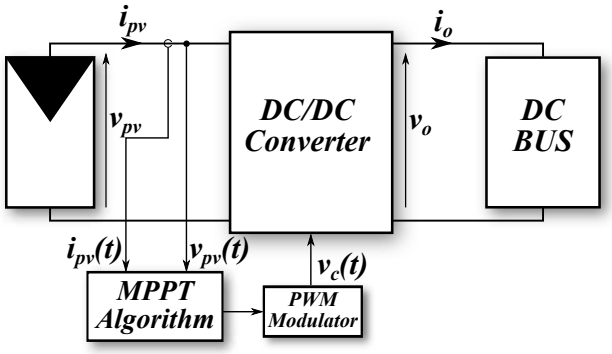
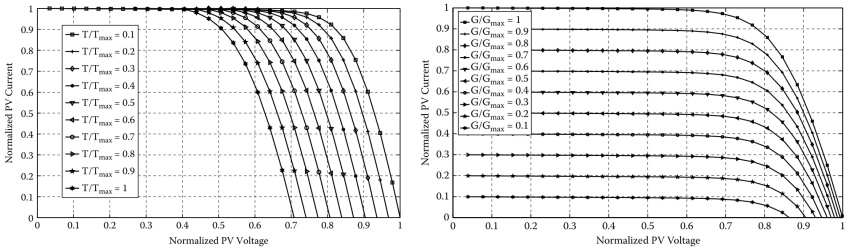


Figure 2.1 Typical scheme of a PV system equipped with the MPPT function implemented by means of a switching converter.

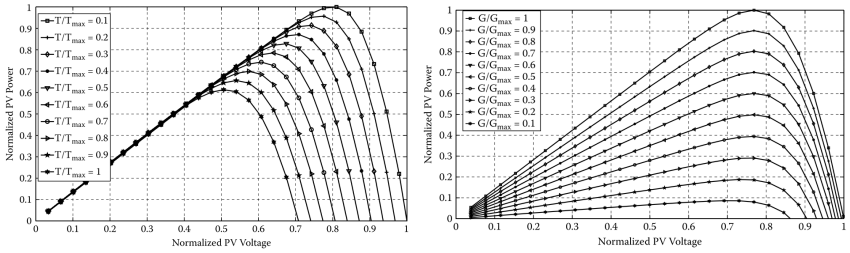
The PV source consists of several PV cells combined in series forming modules, which are in turn connected in series to form a number of PV strings that are connected in parallel. The electrical characteristics of the PV field depend on the cell material, type of cell and electrical connection among the cells. Figure 2.2 shows typical I-V curves of a commercial module, in uniform condition, with several values of temperature and irradiation levels, respectively [1]. T_{max} and G_{max} could be, for example, the values in Standard Test Condition (STC).

Figure 2.3 shows the dependency of the power vs. voltage behaviour on the temperature and the irradiation. Three main points are found: the short-circuit condition, where the PV voltage is zero; the open circuit condition, where the PV current is zero and the MPP condition where the product of PV current and PV voltage assumes its maximum value. In most cases, the power converter is controlled by a MPPT algorithm in order to track the MPP.



(a) (b)

Figure 2.2 Current versus Voltage characteristic: a) at different temperature values T; b) at different irradiation levels G [1].



(a) (b)

Figure 2.3 Power versus Voltage characteristic: a) with different temperature T; b) with different irradiation level G [1].

2.1.1 Photovoltaic Models

A typical I-V curve for a photovoltaic system has been shown in Figure 2.2. It can be obtained from a constant current by subtracting a diode current. Hence, a photovoltaic system can be ideally modelled by a current generator in parallel with a diode. The current generator symbolizes the photo-induced current, depending on temperature, irradiation level, area and material of the PV cell. The diode describes the diffusion and recombination characteristic of the charge carriers in the material. To take into account also the other physical phenomena, the two-diode model, shown in figure Figure 2.4, is usually found in the literature [41]-[1]. The series resistance R_s consists of the internal resistance of the solar cell and the contact resistance; the shunt resistance R_{sh} represents the leakage current and the second diode describes the recombination in the space-charge zone. A good compromise between

the model complexity and accuracy is achieved by the single diode model shown in Figure 2.5 where the contribution of the second diode is neglected.

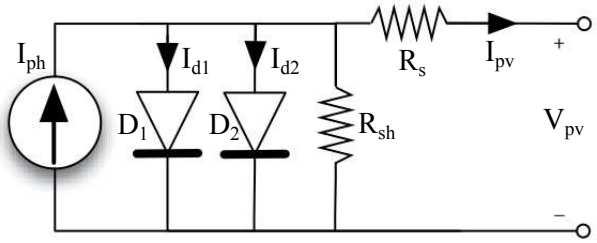


Figure 2.4 Two diode model of a PV cell/module [1].

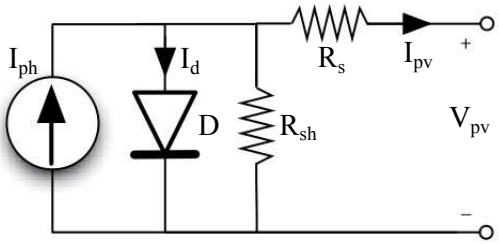


Figure 2.5 Single-diode model of a PV cell/module [1].

The single diode model will be used in the following for modelling the PV source. All the aforementioned parameters are greatly dependent on the irradiance level and temperature as well as on the operating point. This makes the PV source strongly non-linear and depending on the actual operating conditions. Thus, a MPPT algorithm is required to track the maximum power point for ensuring the maximum power production in any condition.

2.1.2 Maximum Power Point Tracker in PV Application

The output power vs. voltage characteristic of a PV array exhibits a unique maximum, called MPP. This is true provided that the PV source works at uniform irradiance and temperature conditions and that the

parametric mismatching among the cells is not significant. Such conditions occur in many practical cases, so that it is assumed that they are hold in the present study [42]. As a consequence of temperature and/or irradiance level variations, the position of the MPP changes as well (see Figure 2.3). Then, it is mandatory to continuously track the MPP in order to extract the maximum power from the PV source whatever the operating conditions are. In the literature several MPPT algorithms are presented. Among these methods, the Perturb & Observe (P&O) is the most common due to low amount of physical resources it requires and for its robustness. Indeed, it consists in perturbing the PV array voltage by means of a step change that is applied through the dc/dc converter duty cycle [1]. If the power extracted from the PV source increases, this means that the operating point has moved towards the MPP, so that the next duty cycle perturbation is applied with the same sign of the preceding one. Otherwise, the sign of the perturbation is reversed. The perturbation can be applied directly to the converter duty-cycle or to the reference voltage if the dc/dc converter operates in closed loop (see Figure 2.6).

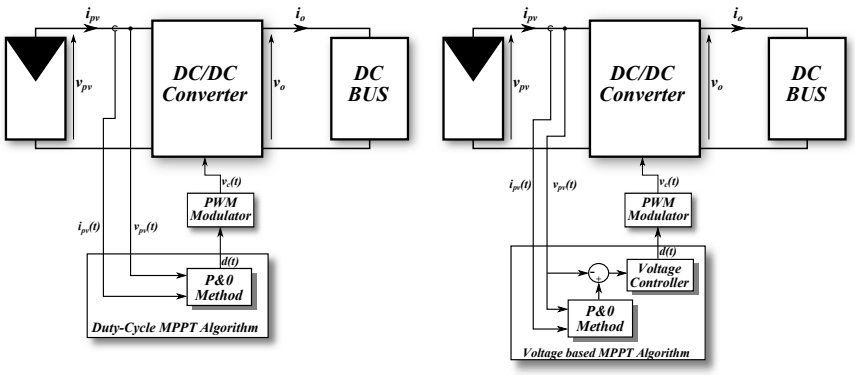


Figure 2.6 Basic schemes for Duty-Cycle MPPT Algorithm and Voltage based MPPT Algorithm.

In the following the stepwise perturbation is straightforwardly applied to the converter duty-cycle for implementing the basic P&O technique, but the results can be extended to any other perturbative MPPT approach.

The MPPT efficiency depends on the two parameters of the P&O

algorithm: the perturbation amplitude $\triangle d$, i.e. the amplitude of the perturbation, and the perturbation period T_p, i.e. the time interval between two consecutive perturbations. In [1] an in-depth discussion about the dependency of such parameters on both the PV source and the switching converter operating points and parameters is given. Some design equations allowing to settle their values in order to achieve the best MPPT performances are also introduced. In particular, the value of the duty cycle perturbation amplitude Δd must be greater than a suitable value Δd_{min}. The lower threshold (2.1) for Δd was determined in [43]

$$\Delta d_{min} = \frac{1}{G_0} \sqrt{\frac{V_{MPP} \cdot K_{ph} \cdot |\dot{G}| \cdot T_p}{\left(H \cdot V_{MPP} + \frac{1}{R_{MPP}} \right)}} \tag{2.1}$$

where $\dot{G}$ is the average rate of change of the irradiance level in the time interval T_p between two consecutive perturbations, G_0 is the dc gain of the transfer function between the PV voltage and the control variable d. V_{MPP}, I_{MPP} and R_{MPP} are the PV voltage, current and differential resistance at the MPP, and [1]

$$K_{ph} = \frac{\partial i_{PV}}{\partial G} \tag{2.2}$$

$$H = -\frac{1}{2} \frac{\partial^2 i_{PV}}{\partial^2 v_{PV}} \bigg|_{MPP} \tag{2.3}$$

The PV power response to a small step perturbation of amplitude Δd allows to define the settling time T_ε. It is the time that such a response needs before being confined within a band of relative amplitude $\pm\varepsilon$ around its steady-state value, shown in Figure 2.7. The smaller the value of the perturbation period T_p, the more prompt is the MPPT algorithm to track the PV power variation occurring when the environmental conditions change abruptly. If its value falls below T_ε, the PV power does not reach its steady state and the MPPT behaviour can

become incorrect, causing an energy loss. Indeed, an incorrect power evaluation can produce a wrong choice of the duty variation direction, causing more oscillations across the MPP in steady state and, in some cases, the instability of the MPPT algorithm.

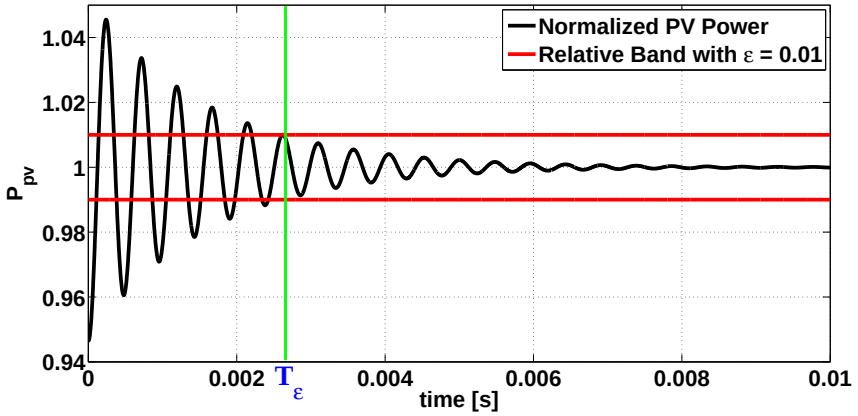


Figure 2.7 PV Power Response to a duty perturbation of amplitude Δd. The red line is the band of relative amplitude $\pm\varepsilon = 0.01$

In literature, some authors propose techniques aimed at adapting the P&O parameters to the varying PV array operating conditions in order to improve the MPPT efficiency. They are generally focused to the optimization of the perturbation amplitude Δd only. This happens in the field of PV systems [44][45][46][47][48], but also for the optimization of the performances of wind energy systems [49] and of other renewable energy systems [50]. The value of the perturbation period has been optimized by means of the approach presented in [51] only. Unfortunately, a systematic procedure for achieving the objective is not given and an empirical equation is used for adapting T_p to the current PV operating conditions.

This chapter is aimed at introducing a new approach to the dynamical optimization of the perturbation period. In the following it is firstly shown that a proper choice of T_p allows an improvement of both the tracking speed and the stationary MPPT efficiency. Moreover, it will be demonstrate the inability to evaluate the T_p in time-domain and then the usefulness of a real-time identification technique.

2.1.3 MPPT efficiency

A suitably small T_p allows the MPPT algorithm to track promptly the MPP variation due to environmental conditions changes, but also leads to an increase of the stationary MPPT efficiency. The system steady state across the MPP is characterized by a sequence of N_L operating points on the left side and of N_R operating points on the right side of the MPP. The MPPT controller moves the operating point from the extreme left side of the MPP up to the extreme right side and vice versa, in a sequence of $N_p = 2 \cdot (N_L + N_R)$ consecutive operating points of duration $T = N_p \cdot T_p$. The stationary MPPT efficiency is defined as

$$\eta_{MPPT} = \frac{< P(t) >_T}{P_{MPP}} \tag{2.4}$$

where P_{MPP} is the MPP power and the numerator requires to calculate the average power delivered by the PV array when it works in each one of the N_p points. The power $P(r), r \in [1,..,N_L]$ and $r \in [1,..,N_R]$, delivered by the PV array in the r-th operating point can be expressed as

$$P(r) = P_{MPP} - \Delta P(r) = P_{MPP} - \alpha \cdot \Delta V_{PV}^2(r) \tag{2.5}$$

where α is a coefficient depending on the PV array characteristics [1] and $\Delta V_{PV}(r)$ is the difference between the MPP voltage and the voltage at which the r-th operating point occurs. By assuming that the operating points are equally spaced, it is $\Delta V_{PV}(r) = r \cdot \Delta V$. Thus, the MPPT efficiency (2.4) can be expressed as

$$\eta_{MPPT} = \frac{2P_{MPP} + 2\left[\sum_{k=1}^{N_R} P(k) + \sum_{j=1}^{N_L} P(j)\right] - P(N_R) - P(N_L)}{2 \cdot (N_L + N_R) \cdot P_{MPP}} \tag{2.6}$$

where $P(N_R)$ and $P(N_L)$ is the power delivered by the PV array in the two extreme operating points. By means of (2.5) it results that

$$\eta_{MPPT} = 1 - \frac{\alpha \Delta V^2 \left[N_R^2 + N_L^2 + 2\sum_{k=1}^{N_R-1} k^2 + 2\sum_{j=1}^{N_L-1} j^2\right]}{2 \cdot (N_L + N_R) \cdot P_{MPP}} \tag{2.7}$$

This expression confirms that the MPPT efficiency increases when the perturbation time T_p decreases. Indeed, due to the relationship

between ΔV and Δd imposed by the dc/dc converter, and due to the fact that Δd is proportional to the square root of T_p (see (2.1)), the second term in (2.7) can be seen as quasi linear with respect to T_p. Additionally, a wrong T_p value might deceive the MPPT algorithm so that both N_R and N_L increase, this leading as well to an increase of the second term in (2.7).

2.1.4 T_p Estimation

A procedure for the real-time identification of the optimal value to be assigned to the perturbation period T_p would be useful in order to have the best MPPT performances in any operating condition, regardless from the type of PV source and the parameters of the switching converter employed for implementing the MPPT algorithm. The PV source type, the irradiance and the temperature greatly change the dynamic characteristics of the PV panel. Similarly, the current and voltage levels at which the dc/dc converter operates, as well as aging effects and tolerances affecting the converter parameters, have a significant influence on the system dynamics, thus also on its settling time.

The settling time T_ε can be measured on the experimental system step responses. If the PV power response, which is obtained as the product of the measured PV current and voltage waveforms, is used to this aim, the noise introduced by the analog-to-digital converter (ADC) makes such an estimation inaccurate. Figure 2.8 shows an example of the PV power and PV voltage step responses. The data have been obtained by using a 12 bit ADC with a quantization error $e_V = 40mV$ and $e_I = 20mA$ for the voltage and current channels respectively. It is obvious that the PV power step response does not allow to evaluate the settling time T_ε because of the finite ADC resolution. Indeed, the accuracy of the PV power measurement is poor and the error propagation makes it dependent on the actual PV operating point

$$\begin{gathered} p_{PV} = v_{PV} \cdot i_{PV} = (V_{PV} \pm e_V)(I_{PV} \pm e_I) = \\ V_{PV} I_{PV} \pm V_{PV} e_I \pm I_{PV} e_V \pm e_V e_I \end{gathered} \tag{2.8}$$

where V_{PV}, I_{PV} are the instantaneous PV voltage and current values respectively.

On the other side, a T_ε estimation based on the voltage step response suffers from the large overestimation or underestimation of the PV voltage settling time $T_{\varepsilon,v}$, due to the ratio between the ADC resolution and the threshold ε used to evaluate the settling time. In conclusion, the estimation of T_ε in the time domain, by means of the power or voltage step response, is affected by a large uncertainty.

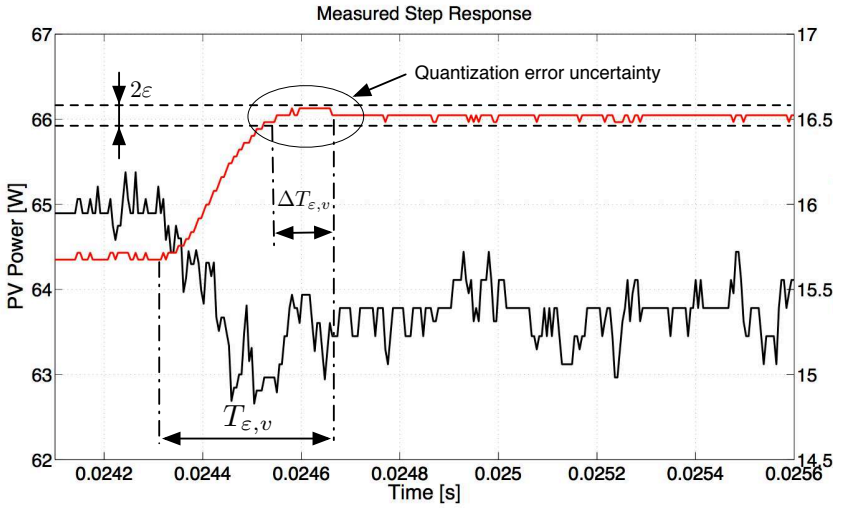


Figure 2.8 Response of the PV power (black) and the PV voltage (red) to a step variation of the duty cycle around the MPP: experimental measurements

Another possibility to evaluate the settling time T_ε is, according to the time-domain expression of the power step response, to use the equation in (2.9). The switching converter topologies commonly used for MPPT PV applications are described by means of a second order model [52]. Otherwise, whenever the converter transfer function assumes a more complex expression, the values of the parameters of the second order dominant dynamic behaviour can be still evaluated numerically, so that the following discussion remains valid. Under this condition, in [1] an expression of the settling time is given.

$$T_\varepsilon \simeq -\frac{1}{\zeta\omega_n} \cdot \ln\left(\frac{\varepsilon}{2}\right) \tag{2.9}$$

where ω_n and ζ are the natural frequency and the damping factor, respectively. Alternatively, T_ε can be indirectly calculated by measuring the PV voltage settling time $T_{\varepsilon,v}$, thus by analyzing the voltage step response, instead of the power step response, so that the following equation is used

$$T_\varepsilon \simeq T_{\varepsilon,v} + \frac{1}{\zeta\omega_n} \cdot \ln(2) \tag{2.10}$$

According to (2.9) and (2.10), an accurate T_ε evaluation requires the knowledge of the values of ω_n and ζ, or at least of their product. The latter value would be easily identified by means of $T_{\varepsilon,v}$ but, unfortunately, as pointed out above, this time is greatly affected by the ADC resolution, so that it cannot be measured accurately.

Unfortunately, the values of ζ and ω_n, and as a consequence T_ε, strongly depend on the whole system operating point, namely the PV array type, the weather conditions and the dc/dc converter parameters, the latter ones being in turn affected by the current and voltage levels, by aging phenomena and by tolerances and uncertainties. Therefore, in order to optimize the MPPT perturbation period, a real time identification is mandatory to estimate the correct settling time T_ε. Running frequently this identification technique, the MPPT perturbation period will be quickly optimized. However, each time the identification procedure is launched, the MPPT algorithm is stopped, causing likely a produced energy reduction. Then, it is mandatory to achieve a suitable compromise.

In this work, two identification approaches have been considered: a non-parametric and a parametric one. The aim is to compare, in photovoltaic applications, these two different identification approaches in terms of accuracy of the results, algorithmic complexity, required hardware resources and execution time. In the following section the real-time identification is presented and the two chosen techniques are deeply illustrated.

2.2 Real-Time Identification

A real time identification procedure finds many applications in PV systems. Real-time means here that the identification must be performed before a significant variation of the system operating point occurs. In PV applications the changes that normally occur concern the power DC/DC converter and the environmental conditions. The ageing of the power converter components, as well as their de-rating, has a very low dynamic, typically in the range of years. This is not the case of the temperature and of the irradiance variations. They show a faster dynamic, in particular the irradiance level that can change more quickly than the temperature. The European standard EN50530 fixes the rate of change for the irradiance variation and the low irradiance level equal to 100 $W/m^2/s$ and 100 W/m^2, respectively. For instance, by supposing 3 % as maximum acceptable variation, the identification process can be considered in real-time if it is executed in less than 30 *ms*, of course by requiring the cheapest digital controller.

One of the objectives of this work is to verify the usefulness of two different identification approaches (parametric and non-parametric) for real-time identification in low-cost applications.

Among the non-parametric procedures, presented in Chapter (1), the Cross-Correlation Method (CCM) has been chosen. In comparison with the other non-parametric techniques, its main advantages are a short execution time and an intrinsically rejection against the disturbances, providing the un-correlation between the input signal and the disturbances. The CCM allows to estimate the system impulse response without any prior knowledge about the PV system. Moreover, by applying a Fourier Transform (FT) to the identified impulse response, the system frequency response can be obtained. From this response, the needed parameters, ζ and ω_n, can be evaluated. This permits to estimate the settling time by means of (2.9).

Among the parametric procedures, the KF has been chosen. As shown in Chapter (1), several Kalman Filter are used in literature. In this work the DKF has been adopted and applied to the PV system for estimating the PV parameters as well as the states of the system. It is largely used in several applications, e.g. battery management [53]-

[54]-[32]. Identifying the PV parameters is possible to estimate the settling time of the PV system. Unlike the non-parametric techniques, the KF requires the knowledge of the PV system structure and then an in-depth study of the PV model has to be achieved. It is worth to note that the KF evaluates also the variances of the estimates. These quantities can be used for on-line monitoring and diagnosis purposes.

In the following, a brief introduction about the input signal, used in identification procedures, is given. Subsequently, the Cross-Correlation Method is presented and its mathematical formulation is reviewed. A fast technique for evaluating the cross-correlation is described, together with the main relations of the Fourier Transform. The identification procedure based on CCM in PV applications is illustrated and the way for estimating the settling time is displayed. After that, the Kalman filter, and in particular the DKF, is reminded. Finally, a consistency test, for parametric approaches, will be presented and the settling time evaluation using DKF is pointed out.

2.2.1 Input Signal

The input signal, having the role of exciting the system dynamics of interest, is a key factor of the identification procedures. In order to quantify the perturbation level of a signal the concept of *persistence* is introduced.

Definition 2.2.1. A signal u(t) has a persistence of order n if there are n different pulsation $\omega_1 \neq \omega_2 \neq \cdots \neq \omega_n$ where the spectral density is not zero. A different and more formal definition can be found in [18], where the signal u(t) is defined persistent of order n if:

1. the covariance function limit exists:

$$r_u(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} u(t+\tau)u^T(t) \tag{2.11}$$

2. the $R_u(n)$ is positive definite, where $R_u(n)$ is the covariance ma-

trix, defined as:

$$R_u(n) = \begin{bmatrix} r_u(0) & r_u(1) & \cdots & r_u(n-1) \\ r_u(-1) & r_u(0) & \cdots & r_u(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_u(1-n) & r_u(-n) & \cdots & r_u(0) \end{bmatrix} \tag{2.12}$$

A non-negligible aspect of the identification method is in the fact that the input signal has to be generated as simple as possible. This is a key feature in view of a software or hardware implementation. Several different input signals have been proposed in literature, e.g. step function, autoregressive moving average process, sum of sinusoids [55]. Among these perturbation signal the *Pseudo Random Binary Sequence* (PRBS) is chosen in this work because of its high persistence order and its simplicity. In the following these two characteristics are demonstrated.

PRBS Signal

The PRBS is a periodic and deterministic sequence assuming only two values [56]-[57]. It can be simply generated by using a n-bit shift register with a XOR gate in feedback. The output sequence can assume a value equal to 0 or 1. However, a PRBS with arbitrary amplitude e can be obtained by manipulating the PRBS sequence (see Figure 2.9). Then, the designer has to select the PRBS value, the clock period and the PRBS length. These choices depend on the application, the ADC resolution, the characteristic of interest and the Signal Noise Ratio (SNR) in the system.

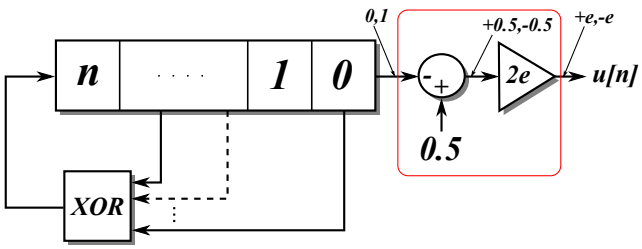


Figure 2.9 Generation of the PRBS signal using a shift register and a XOR gate.

The PRBS length will be less or equal to $2^n - 1$. It depends on the choice of the feedback position besides that the shift register length. A proper choice of the feedback taps can guarantee a *Maximum Length Sequence* (MLS) of the PRBS signal [58]-[59]. It is largely used in identification systems as in [60]-[25]-[61] thanks to its high persistence. This peculiar feature is now demonstrated by analysing its spectral characteristics. The relations (2.13)-(2.14) show the mean $\overline{u}$ and the covariance function $r_{uu}(\tau)$ of $u[n]$.

$$\overline{u} = \frac{1}{M}\sum_{n=0}^{M-1} u(n) = \frac{e}{M} \tag{2.13}$$

$$r_{uu}(\tau) = \frac{1}{M}\sum_{n=0}^{M-1} u(n)u(n+\tau) = \begin{cases} e^2, & \tau = 0, \pm M, \pm 2M, \dots \\ -\dfrac{e^2}{M}, & otherwise \end{cases} \tag{2.14}$$

where e and M are the PRBS amplitude and the PRBS length, respectively. The spectral density is given in (2.15) [18].

$$\phi_{uu}(\omega) = \sum_{k=0}^{M-1} C(k)\delta\left(\omega - \frac{2\pi k}{M}\right) \tag{2.15}$$

where

$$C(k) = \begin{cases} \dfrac{e^2}{M^2} & k = 0 \\ \dfrac{e^2}{M^2}(M+1), & k > 0 \end{cases}$$

Therefore the spectral density is:

$$\phi_{uu}(\omega) = \frac{e^2}{M^2}\left[\delta(\omega) + (M+1)\sum_{k=1}^{M-1}\delta\left(\omega - \frac{2\pi k}{M}\right)\right] \tag{2.16}$$

The (2.16) implies that, with a suitably large value M, the DC component can be neglected and the spectral density is a frequency comb with (M-1) equally-spaced spectral elements. Thus, the PRBS signal has a persistence of order $M - 1$.

For example, with a 10-bit shift register the maximum PRBS length is achieved putting in feedback the 0^{th} and the 3^{rd} register position, as shown in Figure 2.10. In this way M is equal to 1023. With a sampling period equal to 5 μs, the PRBS signal, the PRBS covariance and the PRBS spectral density are those ones shown in Figure 2.11.

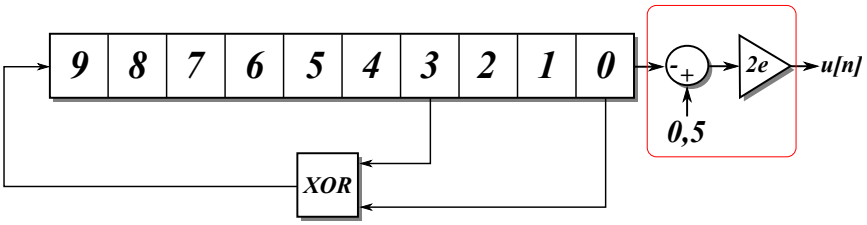
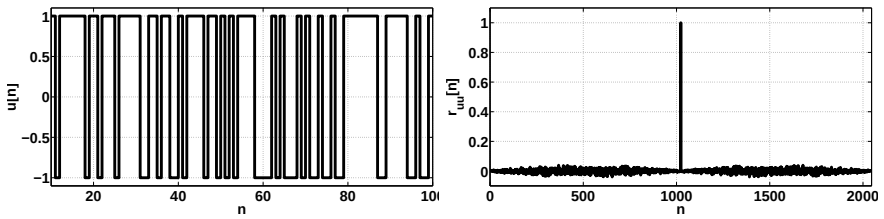
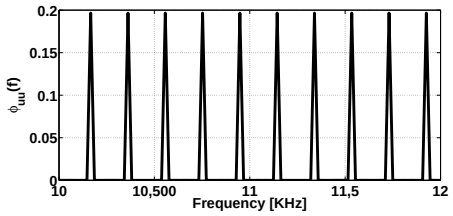


Figure 2.10 PRBS generation using a 10-bit Shift Register.



(a) (b)



(c)

Figure 2.11 PRBS Generation with a 10-bit shift register and T_s equal to 5 μs: (a) PRBS signal (zoom), (b) PRBS covariance, (c) PRBS spectral density (zoom)

PRBS Amplitude in Photovoltaic Applications

In [25] a proper PRBS amplitude is calculated by using an automated procedure that imposes a specified regulation band on the output voltage. This approach is very effective for linear time invariant systems. This is not the case of PV systems, for which the effect of the PRBS amplitude on the system non-linearity has to be studied. Hence, the PRBS amplitude will be chosen in order not to excite the non-linearities of the PV source.

In Chapter 3 (Section 3.3.2), the effect of the PRBS amplitude on the non-linearities of the PV system under analysis is shown. Therefore, a proper amplitude will be chosen.

2.2.2 Cross-Correlation Method

The Cross-Correlation Method is one of the most attractive non parametric methods because of its low identification time and its intrinsic immunity to noise. It is widely applied in the Frequency-domain identification of DC/DC Power Converters. The authors in [62] use this technique for identifying the dynamic responses of a forward converter using low-cost digital hardware. In [63] a modified cross-correlation approach is proposed. In order to achieve reliable identification results multiple periods of PRBS are injected in the system. Using these identified dynamic responses, an automated digital controller is developed [64]. Several improvements are also proposed in [25]-[65]-[66]. This technique is also employed in wide bandwidth application [67]-[66].

The CCM allows to identify the system impulse response and, thanks to a Fast Fourier Transform (FFT), the system frequency response without requiring a prior knowledge of the system structure.

Correlation Analysis

In steady state, for small-signal disturbances, the switching converter can be seen as a Linear Time Invariant (LTI) discrete-system, and the system output can be written as:

$$y[n] = \sum_{k=1}^{\infty} h[k] \cdot u[n-k] + v[n] \tag{2.17}$$

where *y(n)* is the sampled output, *u(k)* is the input digital control signal, *h(k)* is the discrete-time system impulse response and *v(k)* represents disturbances. The cross-correlation between the input control signal *u(k)* and the output signal *y(n)* is:

$$r_{uy}[n] = \lim_{L\to\infty} \frac{1}{2L+1} \sum_{p=-L}^{L} u[p] \cdot y[n+p] \tag{2.18}$$

The equation (2.18) requires an input and output sequence of infinite length or the selection of a window where the cross-correlation can be computed. In order to overcome this issue, the cross-correlation

can be rewritten as a circular cross-correlation [25], as shown in equation (2.19), when the following statements are true:

- $u[n]$ is M-periodic
- the system is in steady-state
- $v[n]$ either decays within M samples or is periodic with frequency multiple of $1/M$

$$\widetilde{r}_{uy}[n] = \frac{1}{M}\sum_{p=0}^{M-1} u[p]\cdot y[n+p] \tag{2.19}$$

where the tilde specifies that the cross-correlation is circular. By replacing (2.17) in (2.19), (2.20) results.

$$\begin{aligned}\widetilde{r}_{uy}[n] &= \frac{1}{M}\sum_{p=0}^{M-1} u[p]\cdot\sum_{k=-\infty}^{+\infty} h[k]\cdot u[n+p-k] + \frac{1}{M}\sum_{p=0}^{M-1} u[p]\cdot v[n+p] \\ &= \sum_{k=-\infty}^{+\infty} h[k]\cdot\widetilde{r}_{uu}[n-k] + \widetilde{r}_{uv}[n]\end{aligned} \tag{2.20}$$

If a white noise with a variance equal to σ^2 is selected as input signal, $\widetilde{r}_{uu}[m]$ is equal to $\sigma^2\cdot\delta[m]$ and the disturbance *v[n]* is uncorrelated with the input signal, therefore, $\widetilde{r}_{uv}[n] = 0$ and the cross-correlation is:

$$\widetilde{r}_{uy}[n] \simeq \sum_{k=0}^{M-1} h[k]\cdot\widetilde{r}_{uu}[n-k] + \widetilde{r}_{uv}[n] \simeq \sigma^2\cdot h[n] \tag{2.21}$$

where σ is the standard deviation of the input signal. Then, the system impulse response can be estimated evaluating the circular cross-correlation between the system output signal with a white noise as input.

In [18] the similarities between the PRBS and the white noise are demonstrated. Therefore the white noise is approximate by PRBS and, according to (2.14), the cross-correlation becomes:

$$\widetilde{r}_{uy}[n] = e^2 \left(h[n] + \frac{1}{M} h[n] - \frac{1}{M} \sum_{k=0}^{M-1} h[k] \right) \approx e^2 \cdot h[n] \tag{2.22}$$

where e is the PRBS amplitude. As can be deduced from (2.22), the bigger the PRBS length M, the more accurate the estimation of the impulse response. Indeed, by increasing M the PRBS becomes more close to the white noise, but the complexity and the execution time of the identification algorithm increase. A suitable compromise between PRBS length and identification accuracy is then required. The circular cross-correlation can be computed using the efficient *Fast Walsh-Hadamard Transform* (FWHT), widely discussed in [68] and in the following section.

Fast Walsh-Hadamard Transform

The MLS symmetry property allows to perform the circular cross-correlation in (2.22) efficiently. To exploit this property it is necessary that u[n], and not y[n], be the circularly shifted signal [101], the $\widetilde{r}_{yu}[n]$ should be used in place of $\widetilde{r}_{uy}[n]$.

$$\widetilde{r}_{yu}[n] = \frac{1}{M} \sum_{p=0}^{M-1} u[n+p] y[p] \tag{2.23}$$

where $u[n]$ is the MLS-PRBS sequence with amplitude e. By applying the non-commutative property of cross-correlation:

$$\begin{aligned} \widetilde{r}_{yu}[n] = \widetilde{r}_{uy}[-n] \simeq \sum_{k=0}^{M-1} h[k] \widetilde{r}_{uu}[-n-k] = \\ = \sum_{k=0}^{M-1} h[k] \widetilde{r}_{uu}[M-n-k] = e^2 \cdot h[M-n] \end{aligned} \tag{2.24}$$

From (2.24), the (2.25) is obtained.

$$h[M-n] = \frac{\widetilde{r}_{yu}[n]}{e^2} = \frac{1}{e^2 M} \sum_{k=0}^{M-1} u[n+k] y[k] = \frac{1}{eM} \sum_{k=0}^{M-1} u[n+k] y[k] \tag{2.25}$$

where u_N is the normalized input sequence, which has value equal to $+1$ or -1. The equation (2.25) in matrix form becomes:

$$\begin{bmatrix} h[0] \\ h[M-1] \\ h[M-2] \\ \vdots \\ h[2] \\ h[1] \end{bmatrix} = \frac{1}{eM} \begin{bmatrix} u_N[0] & u_N[1] & u_N[2] & \cdots & u_N[M-1] \\ u_N[1] & u_N[2] & \cdots & u_N[M-1] & u_N[0] \\ u_N[2] & \cdots & u_N[M-1] & u_N[0] & u_N[1] \\ \vdots & & & \ddots & \\ u_N[M-2] & u_N[M-1] & u_N[0] & \cdots & u_N[M-3] \\ u_N[M-1] & u_N[0] & u_N[1] & \cdots & u_N[M-2] \end{bmatrix} \begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[M-2] \\ y[M-1] \end{bmatrix} \tag{2.26}$$

This can be implemented directly in hardware, but it requires approximately $(2^b-1)^2$ additions, for a b-bit PRBS sequence. In order to reduce the complexity, the symmetry and periodicity properties of the MLS PRBS input and a proper manipulation of the matrix are exploited. Firstly, an additional row and column are added to the square matrix. Accordingly, the output vector and the resulting impulse response are adjusted. In particular a zero is added to the output sequence and the identified impulse response will hold an irrelevant element, labelled $\star$, which will has to be discarded.

$$\begin{bmatrix} \star \\ h[0] \\ h[M-1] \\ h[M-2] \\ \vdots \\ h[2] \\ h[1] \end{bmatrix} = \frac{1}{eM} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & u_N[0] & u_N[1] & u_N[2] & \cdots & u_N[M-1] \\ 1 & u_N[1] & u_N[2] & \cdots & u_N[M-1] & u_N[0] \\ 1 & u_N[2] & \cdots & u_N[M-1] & u_N[0] & u_N[1] \\ 1 & \vdots & & & \ddots & \\ 1 & u_N[M-2] & u_N[M-1] & u_N[0] & \cdots & u_N[M-3] \\ 1 & u_N[M-1] & u_N[0] & u_N[1] & \cdots & u_N[M-2] \end{bmatrix} \begin{bmatrix} 0 \\ y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[M-2] \\ y[M-1] \end{bmatrix} \tag{2.27}$$

The (2.27) is called M-transform [68] and it can be rewritten using the matrix $\underline{\underline{M}}$ (not to be confused with the PRBS length M), as shown in (2.28).

$$h^\star[M-n] = \frac{1}{eM}\, \underline{\underline{M}}\, y^\star[n] \tag{2.28}$$

where $h^\star$ and $y^\star$ are the augmented impulse response and output vector, respectively. The expression (2.27) has a computational order equal to $(2^b)^2$. A significant complexity reduction can be reached by exploiting the permutationally equivalence between the previous $\underline{\underline{M}}$ matrix with a Hadamard matrix.

The Hadamard matrix [68] is an orthogonal and symmetric $n \times n$ matrix which elements can only take two values (± 1). For a Hadamard matrix of order $n = 2^b$, the following relation holds:

$$H_n H_n^T = H_n^T H_n = nI_n \tag{2.29}$$

where I_n is the identity matrix of order n. The expression (2.29) also holds after a row or column permutation applied to the H matrix. Natural-ordered Hadamard matrices can be generated either using the Kronecker product [69] or element by element using the AND operator between the binary representations of the element' row and column numbers, as shown in (2.30):

$$h(r,c) = h(c,r) = (-1)^{p(r,c)} \tag{2.30}$$

where

$$p(r,c) = \left\{ \sum_{i=0}^{b-1} B_i(r) B_i(c) \right\}_{mod2}$$

r and c are the row and column numbers (between 0 and $2^b - 1$), respectively. $h(r,c)$ represents the element of the H matrix in position r and c. $B_i(r)$ and $B_i(c)$ are the i^{th} bit of the b-bit binary representations for r and c, respectively. All the other Hadamard matrix with the same order can be obtained by applying row or column permutations to the natural ordered HM.

The M-Transform Matrix is a symmetric orthogonal matrix and then it is a Hadamard matrix [68]. Hence, it can be also generated element-by-element, as shown in equation (2.31), using a generating and tap registers, shown in Figure 2.12.

$$m(r,c) = m(c,r) = (-1)^{p(r,c)} \tag{2.31}$$

where

$$p(r,c) = \left\{ \sum_{i=0}^{b-1} B_i(t(r)) B_i(g(c)) \right\}_{mod2}$$

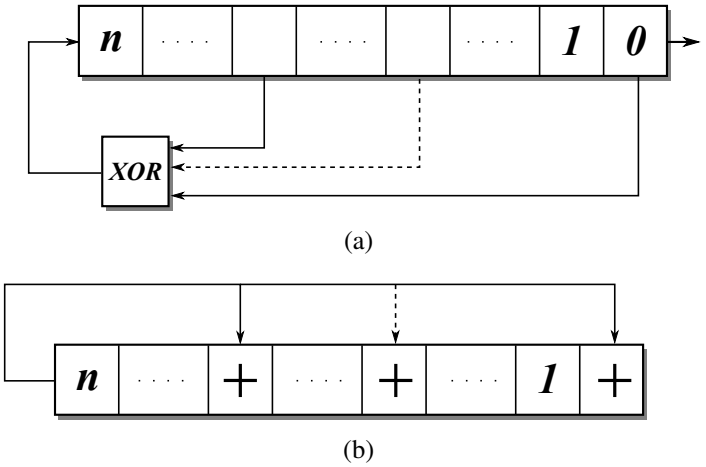


Figure 2.12 (a) Generating Register, (b) Tap Register.

$m(r,c)$ represents the element of the $\underline{\underline{M}}$ matrix in position r and c. $t(r)$ and $g(c)$ are the binary representations of the r^{th} and c^{th} configurations of the tap and generating register, respectively. Therefore, the $\underline{\underline{M}}$ matrix is related to the H matrix by the relation (2.32).

$$\underline{\underline{M}} = P_g^T H P_t \tag{2.32}$$

where P_g and P_t are the permutation matrix defined by the generating register and by the tap register, respectively. Accordingly the relation between the $\underline{\underline{M}}$ matrix and the H matrix, the M-Transform can be computed performing the H-Transform and the equation (2.28) becomes:

$$h^\star[M-n] = \frac{1}{eM}\left\{P_g^T\left[H\left(P_t y^\star[n]\right)\right]\right\} \tag{2.33}$$

Therefore, the impulse response can be computed following these steps: (i) re-order the output sequence $y^\star$ accordingly the matrix P_t; (ii) compute the H-Transform on the re-ordered output; (iii) re-order the transformed sequence in accordance to P_g^T and divide by (eM); (iv) discard the first term in $h^\star$ and invert it keeping fixed the second term. Between all these steps, the one that requires more efforts is surely the computing of the H-Transform. Such a transform is part of the Generalized Fourier Transformer class, then it can be carried out in a very efficient way using a Fast transform, i.e. the FWHT. Its

complexity is now reduced to $O(b2^b)$. It has the same main butterfly structure of the FFT. Figure 2.13 shows an example of FWHT butterfly structure with $b = 3$.

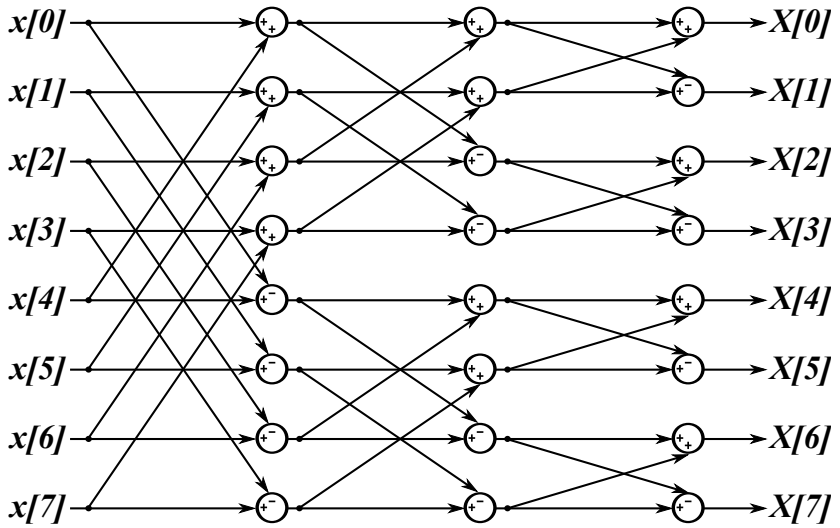


Figure 2.13 FWHT basic butterfly structure.

An in-depth discussion about the permutationally equivalence between the $\underline{\underline{M}}$ matrix and the Hadamard matrix and their generation is given in [101].

Fast Fourier Transform

The FFT is an algorithm that allows to compute the Discrete Fourier Transform (DFT) (2.34). It changes the time representation of a signal into its frequency representation. For this reason, it is largely used to study the frequency property of the signals. It is:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \;\; 0 \leq k \leq N-1 \tag{2.34}$$

where

$$W_N = e^{-j2\pi/N}$$

is the twiddle factor. $X(k)$ and $x(n)$ are the transformed signal and the initial signal, respectively. Therefore, to perform the DFT, N complex multiplications and N additions are necessary, then, its complexity is $O((2^b)^2)$. In order to reduce its complexity, the FFT algorithm can be used [70]. It can be adopted in order to compute the DFT of a sequence with length $N = 2^b$. Its complexity is $O(b2^b)$ and several FFT algorithms are presented in literature. The most common are:

- Radix-2 decimation in-time
- Radix-2 decimation in-frequency

As shown previously, the WHT is a generalized class of Fourier Transform and the FWHT is based on the same butterfly structure as the FFT. Thus, in order to reduce the required resources, both the transformations can be performed on the same butterfly module. For this reason the chosen FFT algorithm is the Radix-2 decimation in-frequency, because its structure is similar to the FWHT one. Figure 2.14 shows the basic structure of the FFT Radix-2 decimation in-frequency with $b = 3$.

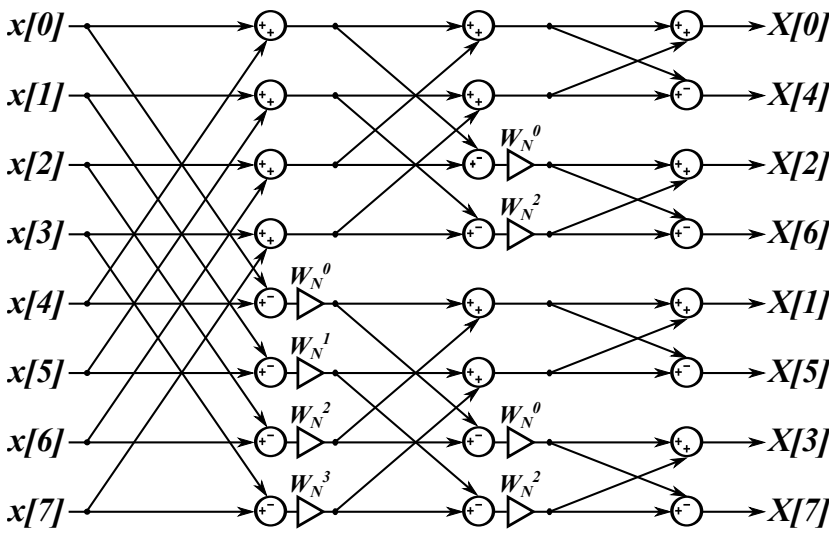


Figure 2.14 FFT basic butterfly structure.

It is worth to note the large similarity between the FFT butterfly structure and the FWHT butterfly structure in Figure 2.13. The input

sequence is in normal order, however, in the FFT the output vector is in bit-reverse order. Then, a re-ordering is required in order to obtain the correct output sequence. In addition to this difference, in the FFT butterfly structure the coefficients W_N^k are also present. They are the twiddle factors shown in equation (2.34). Moreover, the CCM requires first the FWHT, for identifying the impulse response, and after the FFT, for calculating the frequency response. Thus, the two transformation have to be executed in series. For these reasons, both transformation can share the same butterfly structure, allowing a significant resources reduction without increasing the execution time. This choice does not allow to use the standard FFT algorithms, because they are not able to perform the Hadamard Transformation. Therefore, an ad-hoc FFT and FWHT block needs to be developed. An in-depth discussion about this implementation is presented in Chapter 4.

In the following section the CCM application to PV systems is described in order to estimate the P&O settling time.

Identification in PV Applications

The CCM method allows to identify the frequency response of the whole PV system without any prior knowledge about it. The FWHT allows to compute in an efficient way the cross-correlation between the PRBS sequence and the system output signal. The Hadamard Transformation and the FFT can be performed by the same block and lead to the identified system frequency response. Such method has been designed for linear time invariant system. Thus, in the case of PV applications, it has to be applied around a given operating point, e.g. the MPP. As for any non linear system, an appropriate choice of the perturbation signal amplitude has to be done. More details on this point will be given in Chapter (3).

Figure 2.15 summarizes the steps of the procedure that allows to perform the identification of the PV system:

1. The MPPT algorithm is stopped by freezing the duty cycle to the value that ensures that the PV array works at its MPP.

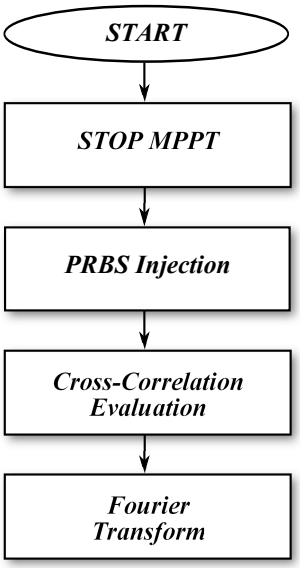


Figure 2.15 Flow Chart of the Cross-Correlation Method.

2. The PRBS sequence is superimposed on the duty cycle of the DC/DC converter in charge of controlling the PV system. In order to ensure a periodic steady-state of the system, two PRBS sequences are injected and the output data obtained during the second burst are collected.

3. The cross-correlation between the PRBS input and the collected output data is performed by means of the FWHT. Thus, the system impulse response is identified.

4. The FFT algorithm allows to calculate the Fourier transform leading to the identification of the system frequency response.

Once the frequency response of the whole PV system is identified, the system settling time is calculated as follows.

Settling Time Estimation

The settling time is calculated on the basis of the system frequency response by means of (2.9), only if the PV system transfer function

$G_{vp,d}$ is a second-order one. Some assumptions have to be done in order to demonstrate that this model holds for the PV system under study.

Without loss of generality, it is assumed that the dc/dc converter is of boost type. In this case, the small signal model of the PV system in the neighborhood of an operating point is shown in Figure 2.16. The PV differential resistance r_d is defined as in (2.35).

$$r_d = \frac{\tilde{v}_{pv}}{\tilde{i}_{pv}} = -(R_p + R_s) \tag{2.35}$$

where R_s is the series resistance, already defined above, whereas, R_p is the parallel between the R_{sh} and the internal resistance of the diode (see Figure 2.5). $\tilde{v}_{pv}$ and $\tilde{i}_{pv}$ are the small signal PV voltage and the small signal PV current, respectively. The value assumed by r_d is strongly depending on the irradiance level and on the PV operating point. When the irradiation is high its value is equal to few Ohms, whereas, if the irradiation decreases, its value can reach tens or hundreds Ohms [1].

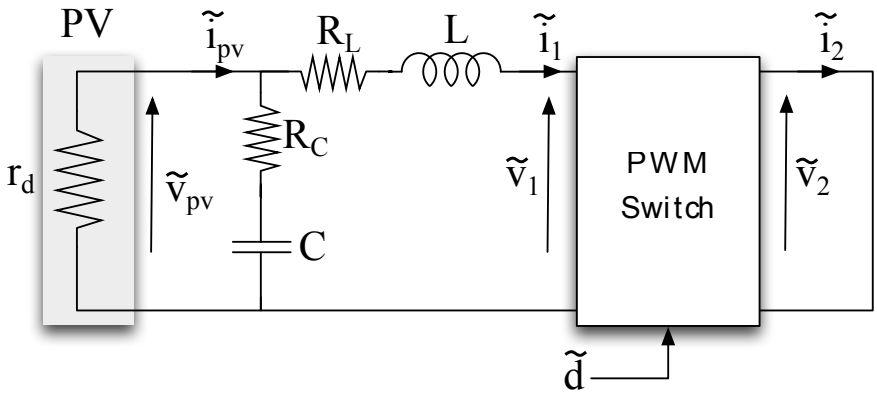


Figure 2.16 PV small signal model in the neighborhood of an operating point.

As for the dc-dc converter, the switching cell is replaced with a linearized small-signal model [71] described by the following equations:

$$\begin{cases} \tilde{i}_2 = (1-D)\cdot\tilde{i}_1 - I_1\cdot\tilde{d} \\ \tilde{v}_1 = (1-D)\cdot\tilde{v}_2 - V_2\cdot\tilde{d} \end{cases} \tag{2.36}$$

Such equations describe the averaged behaviour of the block *PWM SWITCH* shown in Figure 2.16. By using the scheme presented in Figure 2.16 and by keeping into account (2.36), the following $G_{vp,d}$ transfer function, between the duty cycle d and the PV voltage v_{PV}, is obtained

$$G_{vp,d}(s) = \frac{\tilde{v}_{pv}}{\tilde{d}} = \mu \left(\frac{\omega_n^2}{\omega_z} \right) \frac{s + \omega_z}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.37}$$

where

$$\begin{aligned} 2\frac{\zeta}{\omega_n} &= \frac{L}{R_L + r_d} + CR_c + C(R_L // r_d) \\ \mu &= -\frac{V_b r_d}{R_L + r_d} \\ \omega_z &= \frac{1}{CR_c} \\ \omega_n^2 &= \frac{R_L + r_d}{LC(R_c + r_d)} \end{aligned} \tag{2.38}$$

ω_n, μ, ζ and ω_z are the natural frequency, the DC Gain, the damping factor and the zero of the system, respectively. If the dominant pole approximation is fulfilled, i.e. $\omega_z > 10\ \omega_n$, the system can be considered as a regular second-order one. The last two equations in (2.38) allow to write the inequality $\omega_z > 10\ \omega_n$ as follows:

$$\frac{L}{CR_C^2} \cdot \frac{R_C + r_d}{R_L + r_d} > 100 \tag{2.39}$$

During the design of the switching converter, the passive components are designed so that the parasitic resistances R_L and R_C are as small as possible in order to achieve the highest possible conversion efficiency. Thus, it is reasonable to assume that R_L and R_C assume a negligible value with respect to the minimum value assumed by the dynamic resistance r_d [1]. As a consequence, the inequality (2.39) is simplified as follows

$$\frac{L}{CR_C^2} > 100 \tag{2.40}$$

In a boost converter for PV applications the value of the ratio L/C is usually kept close to one [1] and R_C assumes a value of few $m\Omega$, so that the dominant pole approximation holds and the dynamic behavior of the PV voltage $\tilde{v}_{pv}$ with respect to the duty cycle perturbation $\tilde{d}$ is well approximated by a dominant second order transfer function (2.41)

$$G_{vp,d}(s) = \frac{\tilde{v}_{pv}}{\tilde{d}} = \mu \cdot \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.41}$$

In this case, the equation (2.9) holds and the settling time is estimated by extrapolating the values of ζ and ω_n from the identified frequency response.

Finally, the following equalities hold.

$$\begin{aligned} |\mu| = G_0 &= \left|G_{vp,d}(j0)\right| \\ \underline{/G_{vp,d}(j\omega_n)} &= \underline{/G_{vp,d}(j0)} - \frac{\pi}{2} \\ \left|G_{vp,d}(j\omega_n)\right| &= \frac{|\mu|}{2\zeta} \end{aligned} \tag{2.42}$$

$\left|G_{vp,d}(0)\right|$ and $\left|G_{vp,d}(j\omega_n)\right|$ are the transfer function module at the frequency values 0 and ω_n, respectively. $\underline{/G_{vp,d}(0)}$ and $\underline{/G_{vp,d}(j\omega_n)}$ are the transfer function phase at the same two frequency values. It is worth to note that the phase in 0 can assume only two values (0 or $-\pi$) depending on the sign of the DC gain μ. These quantities can be obtained from the identified transfer function. In Figure 2.17 an example of the experimental $G_{vp,d}$ transfer function is shown: the four values, which are needed for estimating T_ε, have been marked with a red dot.

Therefore the equations in (2.42) allow to determine the value of ω_n and ζ. Indeed, according to the second equation in (2.42), ω_n is the frequency at which the phase delay is $\pi/2$ smaller than the value it assumes in DC. Having the value of $|G_{vp,d}(j\omega_n)|$, from the third equation in (2.42), the relation (2.43) is achieved.

$$\zeta = \frac{|G_{vp,d}(j0)|}{2 \cdot |G_{vp,d}(j\omega_n)|} \tag{2.43}$$

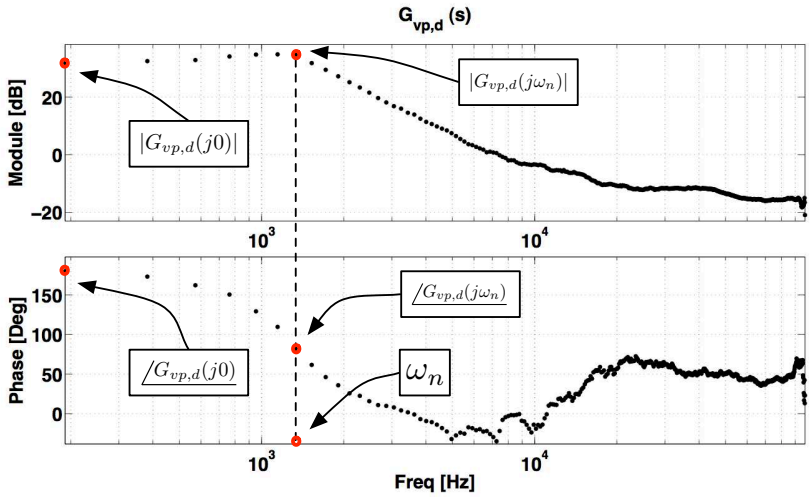


Figure 2.17 Estimation of the $G_{vp,d}$ parameters

Therefore, T_ε is estimated by means of (2.9).

In this section the Cross-Correlation Method has been recalled and its main features have been pointed out. Its usefulness in identifying both impulse and frequency response of a PV system has been confirmed. Moreover, on the basis of some reasonable assumptions on the PV system, the system settling time has been evaluated. In the following section the second identification procedure, belonging to the group of the parametric (or model-based) approaches, is presented. The differences between the application of the two proposed identification techniques to the PV system under study will be pointed out.

2.2.3 The Kalman Filter

The cross-correlation method allows identifying the frequency response of the system under analysis without the need of the knowledge of the system model. Instead, if the system model is known, a parametric identification approach can be adopted. The recursive approaches working during the normal operation of the system, presented in Chapter 1, look very interesting. Such techniques do not require complex calculations or huge amount of memory, and they work with the data just acquired. In particular, the kalman filter has been chosen in this thesis.

The KF is an optimal recursive estimator of a set of states for dynamic systems presented for the first time in [30]. Different kalman filters, as shown in Chapter 1, have been proposed in literature in order to estimate the states and/or the parameters, in linear or non-linear applications, using only noisy measurements. In all cases the knowledge of the system structure, at least, is a requirement.

In this work some unknown parameters of the PV systems have to be identified. For this purpose, the DKF is adopted. In the following, a first overview on the basic LKF is given in order to introduce the fundamental concept then, the main relations of the DKF are reviewed. After that, a test of filter consistency is given and the settling time estimation is demonstrated showing its usefulness for adaptive MPPT algorithms in PV applications.

Linear Kalman Filter

The LKF allows to estimate the observable states of a linear dynamic system. Its mathematical formulation is firstly reviewed. The discrete-time state-space model of a linear dynamic system is shown in (2.44):

$$\begin{cases} x(k+1) = A(k) \cdot x(k) + B(k) \cdot u(k) + w(k) \\ y(k) = C(k) \cdot x(k) + D(k) \cdot u(k) + r(k) \end{cases} \tag{2.44}$$

where $x(k) \in \mathbf{R}^p$, $u(k) \in \mathbf{R}^m$ and $y(k) \in \mathbf{R}^n$ are the states vector, the input signal and the output signal, respectively. The first equation in (2.44) is the state equation and it describes the dynamic of the system. The system stability, as well as its controllability, depend on this equation. The second equation in (2.44) is the output equation and, with the state equation, governs the observability of the dynamic system. $w(k)$ and $r(k)$ are the process noise and measurement noise, respectively. They are assumed to be zero-mean white Gaussian random processes and reciprocally uncorrelated. $A(k) \in \mathbf{R}^{pxp}$, $B(k) \in \mathbf{R}^{pxm}$, $C(k) \in \mathbf{R}^{nxp}$ and $D(k) \in \mathbf{R}^{nxm}$ are the state-space matrix, the input matrix, the output matrix and the feedforward matrix, respectively. They are time-varying entities. The aim of the LKF is to make an estimation $\widehat{x}(k)$ of the unmeasured state $x(k)$ in a dynamic system using the noisy input and output signals. This is made by minimizing the mean squared error between the estimate $\widehat{x}(k)$ and the real state $x(k)$, defined in (2.45).

$$E[(x-\widehat{x})^T (x-\widehat{x})] \tag{2.45}$$

The solution of such a problem is well known in literature [53]. It consists in estimating the state vector $\widehat{x}$ and the error covariance matrix $P_{\widehat{x}} = E[(x-\widehat{x})(x-\widehat{x})^T]$ by means of a set of efficient recursive equations, summarized in Table 2.1. The error covariance matrix $P_{\widehat{x}}$ indicates the confidence in the estimation. When it reaches significantly high values, then the state estimation is not reliable. On the other hand, a low value involves a good quality of $\widehat{x}$. In the following the estimation $\widehat{x}(p|s)$ represents the estimation at time p conditioned by the measurement data up to time s. The KF consists of two distinct steps that have to be performed at each instant k:

1. **Prediction**: a *priori* prediction of the state vector $\widehat{x}(k|k-1)$ by using the previous estimation $\widehat{x}(k-1|k-1)$ and the previous input data $u(k-1)$. In addition, the error covariance matrix $P_{\widehat{x}}(k|k-1)$ is also calculated.
2. **Update**: using the new measurement data the previous prediction $\widehat{x}(k|k-1)$ is corrected by using the Kalman Gain (KG) $L(k)$. Thus, a *posteriori* evaluation of the state vector $\widehat{x}(k|k)$ is made. Moreover, the error covariance matrix is also updated.

Table 2.1 summarizes the main recursive equations of the LKF.

Table 2.1 Linear Kalman Filter

State Prediction	
$\widehat{x}(k\|k-1) = A(k-1)\cdot\widehat{x}(k-1\|k-1) + B(k-1)\cdot u(k-1)$	
$P_{\widehat{x}}(k\|k-1) = A(k-1)P_{\widehat{x}}(k-1\|k-1)A^T(k-1) + W$	
State Update	
KG	$L(k) = P_{\widehat{x}}(k\|k-1)C^T(k)\left[C(k)P_{\widehat{x}}(k\|k-1)C^T(k) + R\right]^{-1}$
$\widehat{x}(k\|k) = \widehat{x}(k\|k-1) + L(k)\left[y(k) - C(k)\widehat{x}(k\|k-1) - D(k)u(k)\right]$	
$P_{\widehat{x}}(k\|k) = \left(I - L(k)C(k)\right)P_{\widehat{x}}(k\|k-1)$	

The transpose operator is indicated with a T to the exponent. W and R are the process noise covariance and the measurement noise covariance, respectively:

$$W(i,j) = E[w(i)w^T(j)]$$
$$R(i,j) = E[r(i)r^T(j)]$$

where $E[\cdot]$ is the mean operator.

The LKF allows to estimate the state vector of the dynamic system under analysis, but, for the PV application under study, the estimation of some parameters is of interest. Thus, the DKF has to be adopted.

Dual Kalman Filter

By using the same relations given for the LKF, a new increased state vector is created with the addiction of the unknown parameters. This approach is the well-known JKF and it allows to estimate at the same time and with the same filter both the states and the parameters. However, such an approach increases the degree of the filter, causing complex algorithms and problem of convergency in high degree filter case [32]. For these reasons, a different approach, the DKF, is employed. It allows estimating the states and the unknown parameters by using two distinct filters. By rewriting the state-space model of the system parametrized by the unknown parameter vector $\theta(k)$, the relations in (2.46) are obtained.

$$\begin{cases} x(k+1) = f(x(k), u(k), \theta(k)) + w(k) \\ y(k) = g(x(k), u(k), \theta(k)) + r(k) \end{cases} \tag{2.46}$$

where $f(\cdot)$ and $g(\cdot)$ are linear (or non-linear) functions that describe the dynamic system. $\theta(k)$ is the vector of the time-varying parameters, whose dynamic state-space model is described in (2.47).

$$\begin{cases} \theta(k+1) = \theta(k) + v(k) \\ d(k) = g(x(k), u(k), \theta(k)) + e(k) \end{cases} \tag{2.47}$$

The first equation indicates that the parameter dynamic is very slow with respect to the state dynamic. $v(k)$ is the parameter noise and it takes into account some driving process that could lead to a parameter changes over time. Some additive estimation errors are considered through the signal $e(k)$. Like the LKF, the DKF consists in two steps: prediction and update. In this case two filters are developed: one for the states and an other one for the parameters. Finally, four steps are presented in the DKF:

1. **State Prediction**: a priori state prediction $\widehat{x}(k|k-1)$ is evaluated by using the previous estimation $\widehat{x}(k-1|k-1)$ and the previous input data $u(k-1)$, as well as the previous estimation of the parameter vector $\widehat{\theta}(k-1|k-1)$. The error covariance matrix $P_{\widehat{x}}(k|k-1)$ is also evaluated.

2. **Parameter Prediction**: a priori parameter prediction $\widehat{\theta}(k|k-1)$ is calculated, as well as the parameter error covariance matrix $P_{\widehat{\theta}}(k|k-1)$.

3. **State Update**: the new measurements are used to calculate the new state Kalman Gain $L_x(k)$. Thus, the a priori state prediction is corrected and a new posteriori evaluation of the state vector $\widehat{x}(k|k)$ is made. The state error covariance matrix is also updated.

4. **Parameter Update**: using the new measurements, the parameter Kalman Gain $L_\theta(k)$ is evaluated. Thus, a new a posteriori evaluation of the parameter vector $\widehat{\theta}(k|k)$ is obtained correcting the a priori parameter prediction. The parameter error covariance matrix is also updated.

The main recursive equations of the DKF are summarized in Table 2.2.

Table 2.2 Dual Kalman Filter

State Prediction

$$\widehat{x}(k|k-1) = f(\widehat{x}(k-1|k-1), u(k-1), \widehat{\theta}(k|k-1))$$

$$P_{\widehat{x}}(k|k-1) = \widehat{A}(k-1)P_{\widehat{x}}(k-1|k-1)\widehat{A}^T(k-1) + W$$

Parameter Prediction

$$\widehat{\theta}(k|k-1) = \widehat{\theta}(k-1|k-1)$$

$$P_{\widehat{\theta}}(k|k-1) = P_{\widehat{\theta}}(k-1|k-1) + V$$

State Update

JM	$\widehat{A}(k) = \dfrac{\partial f(\widehat{x}(k), u(k), \widehat{\theta}(k))}{\partial \widehat{x}(k)} \quad \widehat{C}_x(k) = \dfrac{\partial g(\widehat{x}(k), u(k), \widehat{\theta}(k))}{\partial \widehat{x}(k)}$
KG	$L_x(k) = P_{\widehat{x}}(k|k-1)\widehat{C}_x^T(k)\left[C_x(k)P_{\widehat{x}}(k|k-1)\widehat{C}_x^T(k) + R\right]^{-1}$

$$\widehat{x}(k|k) = \widehat{x}(k|k-1) + L_x(k)\left[y(k) - g(\widehat{x}(k|k-1), u(k), \widehat{\theta}(k|k-1))\right]$$

$$P_{\widehat{x}}(k|k) = \left(I - L_x(k)\widehat{C}_x(k)\right)P_{\widehat{x}}(k|k-1)$$

Parameter Update

JM	$\widehat{C}_\theta(k) = \dfrac{dg(\widehat{x}(k), u(k), \widehat{\theta}(k))}{d\widehat{\theta}(k)}$
KG	$L_\theta(k) = P_{\widehat{x}}(k|k-1)\widehat{C}_\theta^T(k)\left[C_\theta(k)P_{\widehat{\theta}}(k|k-1)\widehat{C}_\theta^T(k) + E\right]^{-1}$

$$\widehat{\theta}(k|k) = \widehat{\theta}(k|k-1) + L_\theta(k)\left[y(k) - g(\widehat{x}(k|k-1), u(k), \widehat{\theta}(k|k-1))\right]$$

$$P_{\widehat{\theta}}(k|k) = \left(I - L_\theta(k)\widehat{C}_\theta(k)\right)P_{\widehat{\theta}}(k|k-1)$$

where $\widehat{A}(k)$, $\widehat{C}_x(k)$ and $\widehat{C}_\theta(k)$ are the Jacobian Matrix (JM). As for the LKF, the recursive DKF equations have to be performed before the data coming from the new measurements are available. Figure 2.18 summarizes the block diagram of the DKF.

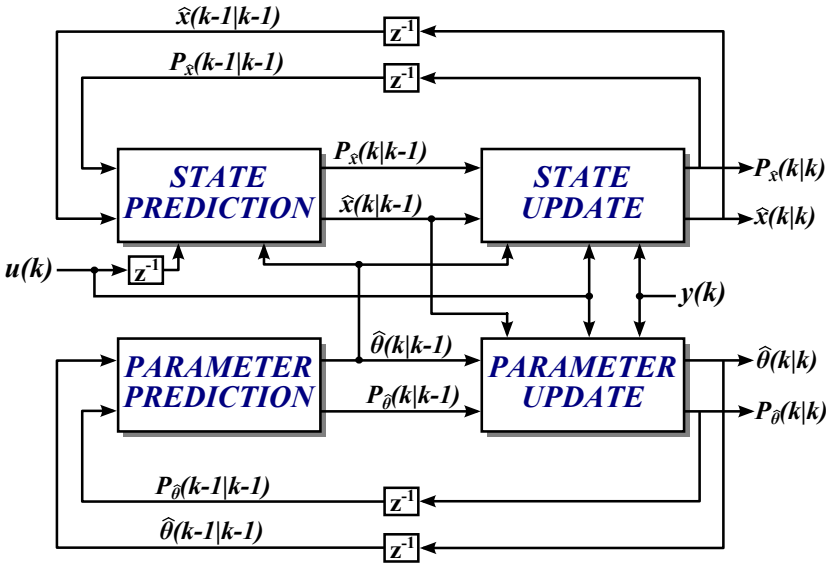


Figure 2.18 Block Diagram of Dual Kalman Filter.

Kalman Filter Initialization

Once the DKF has been developed, the initialization of the state vector and the parameter vector, as well as of the error covariance matrices, has to be made. The DKF is a recursive algorithm so that at the instant $k = 0$ it needs initial state and parameter vectors, $\widehat{x}(0|0)$ and $\widehat{\theta}(0|0)$, and initial error covariance matrices, $P_{\widehat{x}}(0|0)$ and $P_{\widehat{\theta}}(0|0)$, in order to begin the estimation. For linear filters, the effect of different initial vectors reduces with time and they do not influence the steady state performance of the filter [72]. This means that the filter converges to the same steady-state values regardless of the initial states (parameters) and the initial error covariance matrices. However, as far as the non-linear case is concerned, a proper choice of the initial conditions is mandatory. Indeed, in the non-linear case, the linearised model is

computed from the prediction of the states. The better the prediction is, the more accurate the linearised model describes the actual system. Hence, a suitable initialisation is needed for ensuring a valid linearised model from the beginning.

In order to achieve suitable initial conditions of the states $\widehat{x}(0|0)$ and of the parameters $\widehat{\theta}(0|0)$, experimental measurements on the real system or simulations can be adopted [72]. The error covariance matrices, $P_{\widehat{x}}(0|0)$ and $P_{\widehat{\theta}}(0|0)$ describe the confidence of these initial conditions. The lower the confidence is, the higher the initial covariance matrices are. The initial conditions for the application under study will be given in Chapter 3.

Filter Consistency

The quality of the parametric estimation is quantified by means of the filter consistency [72]. An estimator is consistent if it is unbiased:

$$\widehat{x}(k|k) = E[x(k)|y(k)] \tag{2.48}$$

and the equation (2.49) holds.

$$E[(x(k) - \widehat{x}(k|k))(x(k) - \widehat{x}(k|k))^T | y(k)] = P(k|k) \tag{2.49}$$

The filter consistency has to be verified by using the state estimation errors. In practice, the true state is not known and not measurable, so the state estimation errors can not be evaluated. Therefore the only practical way of checking filter performances is to compare the observation sequence with the prediction of the output of the filter. The difference between the current observation and the predicted one is called *innovation*:

$$\upsilon(k) = y(k) - \widehat{y}(k|k-1) = y(k) - C(k)\widehat{x}(k|k-1) \tag{2.50}$$

where C(k) is the output matrix of the I-S-O model. If the filter is consistent, the innovation will be zero mean and white with covariance **S(k)**:

$$S(k) = C(k)P(k|k-1)C^T(k) + R(k) \tag{2.51}$$

where R(k) is the measurement disturbance covariance. Then, in order to verify the filter consistency, these properties have to be checked. By observing the innovation sequence it is possible to understand if these properties hold. More rigorously, a number of statistical tests can be used.

Unbiased Test :
In the following an unbiased test is presented [72]. Its aim is to verify if the innovation is a zero mean sequence. Firstly, the sequence of normalized innovation squared is calculated as given in (2.52).

$$q(k) = \upsilon^T(k) S^{-1}(k) \upsilon(k) \tag{2.52}$$

After that, in order to verify the unbiased assumption, it is necessary to verify that the normalized innovation squared is a χ^2 distribution with m degrees of freedom, where m is the number of inputs, as shown in (2.53).

$$E[q(k)] = m \tag{2.53}$$

The mean of $q(k)$ can be calculated as a sample mean collecting N independent runs of the filter. However, the innovation is assumed uncorrelated and white, then, it is ergodic, therefore its statistic mean can be approximated by a time average for a suitable long time sequence as shown in equation (2.54).

$$E[q] = \frac{1}{N} \sum_{i=1}^{N} q(i) \tag{2.54}$$

In order to verify that $NE[q]$ is distributed as a χ^2 random variable in Nm degrees of freedom, the average $E[q]$ has to fall within a confidence interval $[b_1, b_2]$ so that the hypothesis H_0 that $NE[q]$ is distributed as χ^2_N with a probability $1-\alpha$ holds:

$$P(E[q] \in [b_1, b_2] | H_0) = 1 - \alpha \tag{2.55}$$

The confidence interval has to be constructed using the χ^2 table.

Whiteness Test :
The innovation sequence is white if the relation (2.56) holds.

$$E[\upsilon^T(i)\upsilon(i)] = S(i)\delta_{ij} \tag{2.56}$$

This is again a statistic test. Employing the stationarity and the ergodicity property of the innovation sequence, the (2.56) can be reduced to a time-averaged autocorrelation.

$$r(\tau) = \frac{1}{N-\tau}\sum_{i=0}^{N-\tau-1}\upsilon^T(i)\upsilon(i+\tau) \tag{2.57}$$

Normally, the autocorrelation is normalized, i.e. $r(0) = 1$. The autocorrelation of a white noise process is 1 at $t = 0$ and 0 otherwise. Therefore, if the whiteness condition is verified, the autocorrelation of the innovation sequence will be 1 in 0 and, within certain acceptable bounds, zero otherwise [72]. For suitable long sequence the test statistic can be approximated by a normally distribution with standard deviation $\frac{1}{\sqrt{N}}$, thus 95% confidence bounds in the normalized autocorrelation is approximated by $\pm\frac{2}{\sqrt{N}}$. Therefore, if at least the 95% of the autocorrelation samples will fall within the confidence bounds, the hypothesis will can be accepted, i.e. the innovation sequence is white.

Tuning Parameters

The filter consistency, the speed of the convergence and the steady-state performance depend on the choice of the error covariance matrices. Therefore, a particular attention has to be paid on the tuning of the filter parameters. It consists in choosing properly the covariance matrices of the process noise and parameter noise, W and R, as well as the covariance matrices of the observation noise for the states and for the parameters, V and E. High covariance matrices indicate a low confidence into the related quantity: high covariance matrix of the observation noise indicate low confidence on the measurements, high covariance matrix of the process noise indicate low confidence on the

model and so on. In literature, no methodology for choosing these error covariance matrices is given. Thus, a trial-and-error approach is usually adopted to perform the filter tuning. In [72] some guidelines are provided. They are based on the innovation sequence and on the two tests of un-biasing and whiteness.

Settling Time Estimation

The DKF is used for estimating the PV system settling frequency f_ε, i.e. the inverse of the settling time. Figure 2.19 shows a possible estimation, where the blue curve is the settling frequency estimation and the green curves represent the confidence interval of the estimate, i.e. the estimated parameter $\pm$ the corresponding standard deviation $\sigma_{\widehat{f_\varepsilon}}$. The latter corresponds to the square root of the variance given by the error covariance matrix $P_{\widehat{\theta}}$. The real value of the parameter, in this case f_ε, will fall within the estimated interval $\left[\widehat{f_\varepsilon} - \sigma_{\widehat{f_\varepsilon}}; \widehat{f_\varepsilon} + \sigma_{\widehat{f_\varepsilon}}\right]$. Therefore, in order to evaluate an overestimated settling time, the lower bound is adopted for the settling time evaluation, as shown in (2.58).

$$T_\varepsilon = \frac{1}{\widehat{f_\varepsilon} - \sigma_{\widehat{f_\varepsilon}}} \tag{2.58}$$

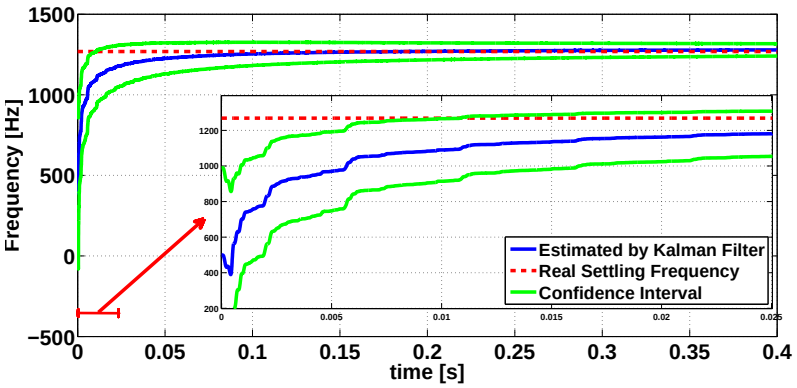


Figure 2.19 Estimated Settling Frequency by the DKF in Matlab/Simulink Simulation test with a parallel resistance equal to 2 Ω.

2.3 FPGA Design Methodology for On-line Identification Systems

In the control systems, digital solutions are commonly accepted as opponent to analog controllers. At the cost of lower efficiency and limited control bandwidth, digital platforms allow a better noise immunity, a high re-programmability, the ability to execute complex algorithms and a higher communication capability.

In Photovoltaic applications, digital software solutions, microcontrollers or Digital Signal Processors (DSPs), are largely adopted thanks to their low-cost and simple programming. However, their fixed internal architecture leads to serialize the treatment and then to increase the execution time. For this reason, in the real-time identification, other solutions have to be adopted. Many studies have confirmed that the hardware solutions, in particular the FPGA technology, are good candidates where high speed performances are required [73]. FPGAs exploit the intrinsic algorithm parallelism allowing a significant reduction of the execution time. Moreover, their increasing integration density allows to achieve more complex algorithms and to develop a distributed control within the same device.

In this section an overview of the FPGA technology is given. Then, its main structure is presented and its advantages in PV applications are focused on. Moreover, a design methodology for both parametric and non-parametric identification techniques is illustrated. Its aim is to give an efficient design methodology for fulfilling the implementation constraints and making the right adequation between the developed architecture and the chosen FPGA target.

2.3.1 FPGA Overview

The FPGA device has been introduced for the first time in 1985 by Xilinx Company. Such a platform belongs to the semi-custom Application Specific Integrated Circuit (ASIC). It gives a good compromise between the ICs and the fully custom ASIC. On one hand, the ICs can perform only a simple and specific function, then, a set of them have to be used in order to perform the desired complex function. Their disad-

vantages are already pointed out above (high time-to-market, reduced speed performances and so on). On the other hand, the fully custom ASIC is a singular IC used for a particular application. Such devices can guarantee high speed performances and low power consumption in the cost of a lower flexibility. Therefore, semi-custom ASICs, and in particular FPGAs, are more and more used in order to achieve proper compromises between flexibility, speed performances and power consumption. In the following section its structure is illustrated and the main elements are presented.

FPGA Structure

The general FPGA structure is deeply described in [75] and the main parts are now reviewed. It consists of a configurable matrix of preallocated logic blocks, as shown in Figure 2.20. Through a fully programmable interconnections the logic blocks can be completely configured by the end-user in order to develop the desired architecture [76]. Mainly two major vendors of FPGA exist in the market: *Xilinx* and *Altera*. Depending on applications the designer has to chose the better platforms for its aims. For low-cost applications *Spartan* (Xilinx) or *Cyclone* (Altera) devices can be used, whereas, for high performances applications *Virtex* (Xilinx) or *Stratix* (Altera) can be adopted.

The main elements, found in a FPGA device, are [75]:

- *Logic Block*

This is the main element in a FPGA device. It is able to perform combinatorial and/or sequential operations. Depending on the manufacturer and on the type of device (high-performances or low-cost), such an element consists of a certain number of *Look-Up-Tables* (LUTs), devoted to combinatorial operations and a set of *D Flip-Flops*, dedicated to sequential operations. Besides, others functions, as *Shift Registers* (SR), *multiplexers*, *adder* (*subtracter*) operations and *distributed RAM memories*, can be performed through this block [74]. Each vendor gives a different name to this logic block: *Slice* or *Configurable Logic Block* (CLB) for Xilinx and *Logic Element* (LE) or *Logic Array Block* (LAB) for Altera.

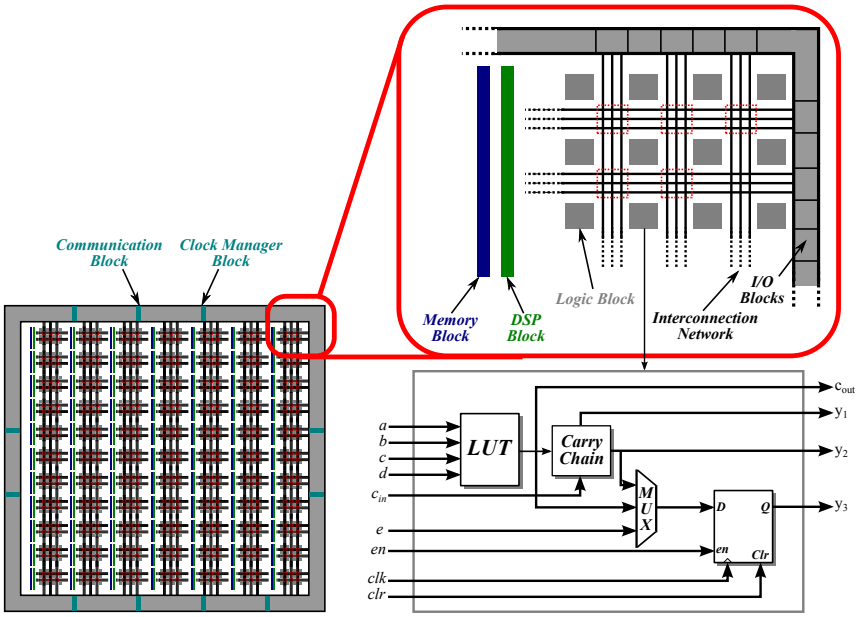


Figure 2.20 General Structure of FPGA device.

- Interconnection Network
It ensures the connection between the different elements for performing the desired architecture. Generally, three interconnection techniques can be found: the *nearest-neighbor technique*, the *segmented technique* and the *hierarchical technique* [77].

- I/O Block
I/O block ensures a bidirectional interface from the external signals to the internal FPGA signals. The I/O blocks can be configured in three ways: input, output or input/output.

- DSP Block
This is a complex arithmetic block constituted by an hardwired multipliers, one or more adders/subtracters and an advanced Multiply AC-Cumulator (MACC) operator. DSP blocks allow computing complex arithmetic functions at high sampling rate.

- Memory Block
The manufacturers provide also embedded memory blocks (ROM, RAM, Dual RAM). In general, a RAM can be implemented either as distributed RAM or as dedicated RAM. In the first case the logic blocks are used to store the data, whereas, in dedicated RAM the embedded memory blocks are used.

- Clock Manager Block
It manages the clocking resources within the FPGA device. Generally, it consists of Phase-Locked-Loops (PLLs) allowing frequency division or multiplication, propagation delay compensation, duty cycle correction and phase shifting. In Xilinx this block is called Mixed Mode Clock Manager (MMCM) and their number depends on the chosen device.

In some cases, inside the FPGA, one or more embedded processor cores can be found allowing the so-called System on Chip (SoC) solutions [78]-[79]. The aspect that makes this solutions always more adopted is the combination of software and hardware treatment.

Design Tools

In order to re-configure the logic block matrix and to validate the final architecture several tools are available. They permit to enhance the quality of the design process allowing a more and more diffusion of FPGA technologies. Hardware design tools for synthesizing, placing, routing and physical implementing are proposed by the vendors. In addition, verification tools, libraries (IP cores), simulation and debug tools are also supplied.

Several tools have been used during this work in order to implement, simulate and debug the developed architecture. Concerning the hardware implementation, the *Integrated Software Environment* (ISE) tool from Xilinx has been adopted. In order to simulate the different architectures, *ModelSim* tool has been used, whereas, for debugging during the experimental validation the *ChipScope* tool has been used [80].

2.3.2 FPGA Contributions in PV applications

In PV applications, software solutions are commonly used in order to implement MPPT algorithms [81]-[82]. However, with the increasing of high speed performance demands and the more and more integration density requests, the hardware solutions, and then the FPGAs, are more frequently employed [83]. Some recent power processing system for PV applications, including those ones dedicated to single modules, e.g. power optimizers and micro inverters, employ FPGAs in order to embed many advanced control functions or real-time PV emulators [84]. In addition, an FPGA platform can be adopted in order to carry out real-time identification of the PV system.

Concerning the speed performance, FPGA takes advantages of the inherent algorithm parallelism, reducing drastically the execution time. This characteristic makes the FPGA the most suitable platform for real-time identification.

As far as the high integration density is concerned, more complex algorithms and distributed control can be implemented within the same device. Moreover, the same hardware block implemented in the FPGA can be re-used and the whole architecture is quite easily scalable and portable in a dedicated ASIC device.

2.3.3 FPGA Design Methodology

In PV applications, the cost is one of the most important aspect that influences the end-user to choose one controller instead of another. Hence, the main issue will be to develop a low-cost controller preserving high time performances. To this aim, a suitable design implementation methodology is essential. It has to be able to meet a proper adequation between the control algorithm and the chosen low-cost FPGA platform using the inherent algorithm parallelism and, then, keeping low the execution time.

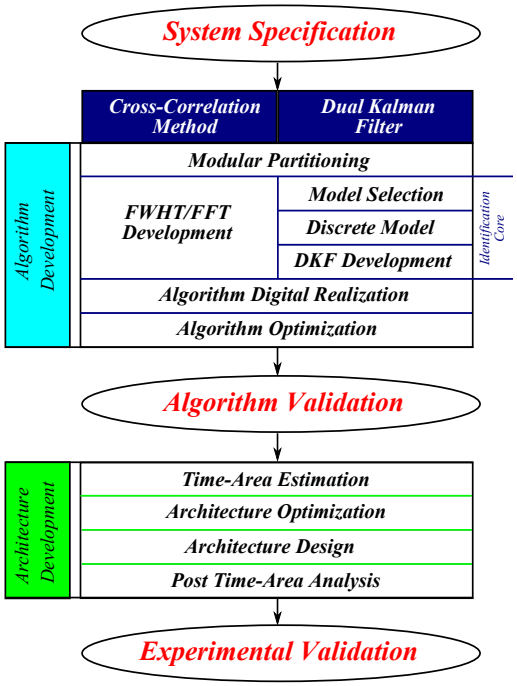


Figure 2.21 Adopted design methodology.

Figure 2.21 shows the adopted implementation procedure for real-time non-parametric and parametric identification procedures. It highlights the differences between the implementation procedure of the two identification techniques. The aim is to give to the designers a global point of view in terms of complexity, required resources and execution

time for these two different identification techniques. In the following subsections each step is investigated.

System Specification

First of all, a preliminary system specification has to be made. It consists in giving a physical specification about the PV system and in choosing the digital platform where the control algorithm, and then the identification procedures, will be implemented.

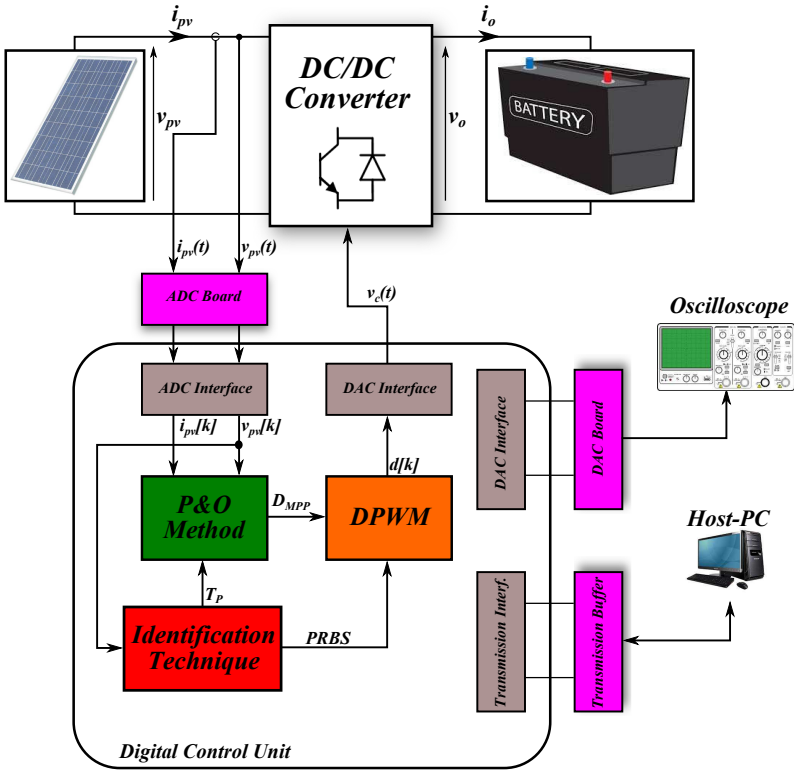


Figure 2.22 Synoptic of the developed adaptive MPPT controller.

The synoptic of the PV system controlled by the developed adaptive MPPT controller is overviewed in Figure 2.22. It underlines the main blocks in PV applications as the PV panels, the power DC/DC converter, the load, the ADC Board and the digital control unit. The latter is a FPGA platform where both the MPPT algorithm and the

identification techniques (CCM and DKF) have been implemented. Now each of these principal blocks are described in details.

1. PV panel
The photovoltaic panel is a Kyocera KC120-1 characterized by an open circuit voltage equal to 21.5 V and a short circuit current of 7.45 A. Its length is about 1.4 meters with a width of 65.2 centimetres. It is positioned on the roof of the University of Salerno.

2. Load
A 36 V battery pack has been connected to the converter output guaranteeing a quasi constant voltage. This assumption is evidently verified with a battery load but it can be considered true in the most PV practical case. Indeed, in case the PV system is connected to the grid, an other power stage (DC/AC converter) is placed in place of the battery and a bulk capacitance is typically adopted in order to keep quasi-constant the output voltage v_o [1].

3. Power DC/DC Converter
In the application under study a DC/DC boost converter has been adopted. The switching frequency f_{sw} is equal to 200 kHz and its main parameters are summarized in Table 2.3.

Table 2.3 DC/DC Boost Power Converter

Parameter	Absolute Value
Input Inductance L	115 μH
Equivalent loss Resistance R_L	100 $m\Omega$
Input Capacitance C	50 μF
Capacitor ESR R_C	10 $m\Omega$

4. ADC Board
The PV voltage and current are acquired by using two sensors: one for the voltage and one for the current. On one hand, a series sensing resistance of 5 $m\Omega$ has been adopted for the voltage sensing. On the other hand, regarding the PV current sensing, a ACS712 of Allegro has been used. Thanks to two operational amplifier LMH6551 of Texas Instru-

ment the voltages in output to the both sensors are amplified and sent to the ADC122S706 of Texas Instruments. This provides the converted results with a frequency up to 1 Mega Sample Per Second (MSPS) and on 12 bits resulting in a voltage and current resolution equal to 40.4 mV and 20 mA, respectively.

It is of prime importance to choose properly the sampling period T_s. Being the PV voltage affected by the ripple due to the input inductance, an easy way to filter the voltage measurement is sampling it at the good instant and one time per switching period T_{sw} as shown in Figure 2.23.

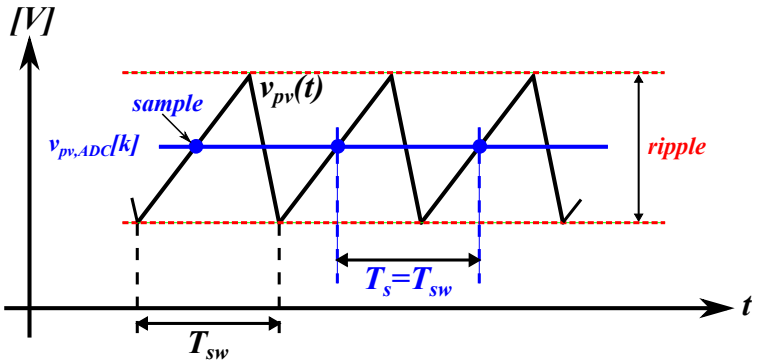


Figure 2.23 Sampling of the photovoltaic voltage.

Thus the sampling period is equal to the switching period, i.e. $T_s = 5\ \mu s$.

5. FPGA device

In this PV application, the digital control unit is based on a Xilinx FPGA board. The objective is to implement the proposed adaptive controller in a low-cost FPGA platform. Thus, among the Xilinx FPGAs, the low-cost XC6SLX45 Xilinx Spartan-6 FPGA has been chosen.

6. Host-PC Interface

Real-time transfer of data between the FPGA board and the Host-PC

is ensured by this interface. In this work the ChipScope tool of Xilinx has been adopted in order to acquire the internal signals of the FPGA device for debugging the final hardware architecture. In addition, such a tool allows to set the controller parameters in real-time, during normal operation of the system. The transmission is based on USB-JTAG protocol.

Given the system specifications, the adaptive MPPT controller algorithm has to be developed. However, the whole algorithm complexity is located in the identification technique. Thus, the main efforts are focused on implementing the identification technique and on the proper adequation between the required identification performances (high-speed and good identification accuracy) and the real system under analysis (high switching frequency, low-cost platform). Hence, the next step is to develop the identification algorithms.

Algorithm Development

After the preliminary system specification the algorithm development is afforded. During this step the designer has to be able to validate the functionality of the identification algorithms and to prepare them for the digital implementation.

1. Modular Partitioning
The aim is to minimize the development time and the algorithm complexity thanks to a hierarchical decomposition [74]-[80]-[85]. It consists in decomposing a large and complex algorithm into simpler and more manageable parts, called modules. An other goal is the regularity, that aims to maximize the re-usability of the modules and to organize the whole algorithm in different levels of granularity [86]-[87]. This approach lead to build a specific library for identification techniques.

2. Identification Core
During this step the biggest differences between the non-parametric and the model-based identification techniques are found. These differ-

ences are now pointed out.

On one hand, the non-parametric techniques require, after the modular partitioning, the development of the operator that allows to identify the system response. For example, in the EIS case, the module and phase have to be calculated, whereas, in the CCM case, the FWHT and FFT have to be performed.

On the other hand, regarding the model-based techniques, the approach is completely different. In this case the model of the system is required and then, as shown in Figure 2.21, the following steps have to be made:

- *Model Selection*
  This is the fundamental step in a model-based identification technique. It is essential to choose the best model structure for the system under analysis. An inappropriate choice could lead to higher complexity, larger identification time and, in some cases, the divergence of the identification technique. Therefore, it is simple to understand the importance of a good model selection.

- *Discretization of the Model*
  After choosing the right model the corresponding discrete model has to be achieved. It is very important to choose accurately the discretization method in order to achieve a suitable compromise between the model complexity and its accuracy. Once the discrete model is obtained, it has to be compared to the continuous one in order to check its accuracy.

- *DKF Development*
  Now the dual kalman filter can be developed using the equations in Table 2.2 and the discrete model just developed.

3. Digital Realization
To implement such modules, 2 steps need to be performed:

- *Normalization*
  The normalization consists in developing a per-unit algorithm in which the coefficients and the variables are replaced by their corresponding per-unit representation. For this purpose, a base

value for each variable has to be determined. This step is required for the DKF development in order to increase the accuracy of the model and to avoid variables with very different dynamical range.

- *Quantization*
  This step consists in determining the fixed-point data format of the whole discrete algorithm. This choice can be performed using a trial-and-error approach, i.e. several simulations are carried out with different fixed-point formats and compared with the floating-point algorithm [75]. The lower word length that satisfies the required accuracy will be chosen. However, also other methods can be adopted for choosing the fixed-point data format. For example Matlab/Simulink is proposing a fixed-point tool to this purpose. The latter, after having run different simulations, detects the maximum and minimum value for each variable. These values are after used by the designer for the fixed-point evaluation.

4. Algorithm Optimization
An optimization of the algorithm is usually required in the case of complex algorithms, as the real-time identification. Typically, this optimization can be reached by simplifying the algorithm and/or the system model, as well as by choosing properly the parameters to identify.

5. Algorithm Validation
Once the whole digital identification algorithm is developed, a final functional validation has to be made. The proposed identification techniques are then validated and their robustness verified.

Architecture Development

After having made the system specification and developed the identification algorithm, the designer starts with the development of the corresponding FPGA architecture. Different tools are available for generating automatically the VHDL (Very high speed integrated Hardware Description Language) code, as the HDL Coder toolbox of Matlab/Simulink or the Xilinx's SysGen toolbox [88]. This tool allows implementing quite easily the algorithm developed in Simulink. The main issue is the lack of flexibility to synchronize the different tasks of the algorithm to be implemented.

For this reason, in this work, the FPGA-based architecture has been hand-coded, respecting the following constraints:

- *Modularity constraint*: the algorithm modularity has to be preserved.
- *Area constraint*: as explained above, the developed identification techniques have to be implemented in a low-cost FPGA. Thus, the available resources are limited and, then, a proper adequation between the algorithm and the resources has to be reached.
- *Time constraint*: for real-time identification applications, it is required that the identification is carried out before a change in the system occurs.

For complex algorithms, the fulfilment of all these constraints is a difficult challenge for the designers.

The main steps for the FPGA-based architecture development are presented below and they are summarized in Figure 2.24.

1. Peformance Evaluation
Firstly, an evaluation of the time/area analysis of the developed identification algorithm preserving the whole parallelism is carried out. The aim is to verify if the corresponding FPGA architecture satisfies all the aforementioned constraints. If this is the case, it is possible to go

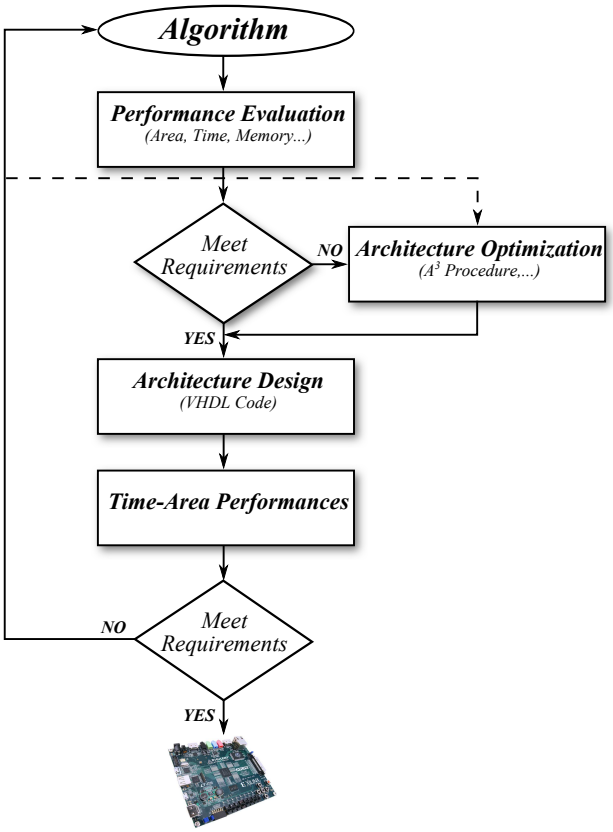


Figure 2.24 Architecture Development Procedure.

directly to the architecture design. Otherwise, if the fully parallel architecture does not meet the previous constraints, some architecture optimizations have to be envisaged.

2. Architecture Optimization
Firstly, in order to cope with the time constraint, a fully pipelined architecture is developed in order to increase the maximum possible clock frequency of the architecture. It consists in placing registers between operators in order to cut the maximum delay path and then to decrease the propagation delay. After that, the factorization of the architecture is carried out by using the *Algorithm Architecture Adequation* (A^3) [89]-[75]. The aim is to develop an optimized architec-

ture that satisfies the area and time constraints. It consists in finding the proper compromise between the number of operators and the time needed to perform all operations. Such an algorithm consists of three steps:

- *Data Flow Graph (DFG) Design*: the algorithm is represented through its corresponding graph.
- *Data Dependency Evaluation*: the data dependency is evaluated and the factorization rules are determined. The factorization reduces the number of operators, and, then, the hardware resources at the cost of a higher execution time. Therefore, a compromise between the hardware resources and the execution time has to be achieved. Generally, the factorization is used for the greediest operators, as multiplier and divider. It can be also applied to an entire module if necessary.
- *Factorized DFG (FDFG) Design*: the factorized algorithm is represented through its corresponding factorized graph. In order to delimit the factorized borders some nodes have to be introduced: Fork (F), Join (J) and Iterate (I) [80].

Figure 2.25 shows a simple example of A^3 procedure.

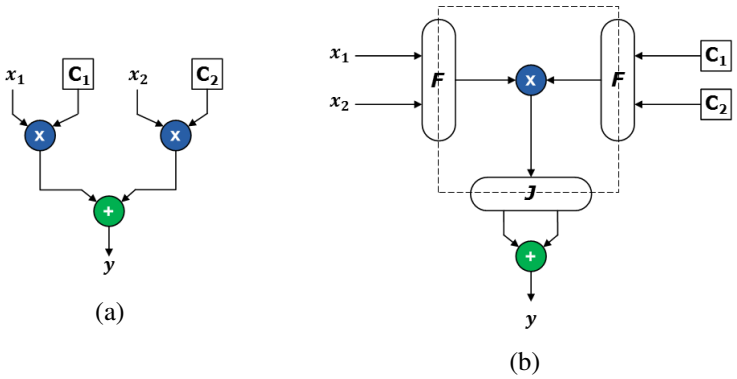


Figure 2.25 A^3 procedure example: a) Data Flow Graph ; b) Factorized Data Flow Graph.

3. Architecture Design

It consists in developing the hardware architecture of each modules. This architecture is constituted by a data path and a control unit. The aim of the first one is to treat the signal in input and to give the desired results. It is obtained from the factorized data flow graph substituting each nodes (F, J and I) by their corresponding operator. A multiplexer is used to implement a fork node and registers are used in place of join and iterative nodes. On the other hand, the control unit has to ensure the synchronization of all the tasks within the module. It is typically a simple Finite State Machine (FSM) that is activated by a *start* pulse signal and indicates the end of the treatment with an *end* signal. Figure 2.26 shows the hardware architecture of the FDFG of Figure 2.25. At this point, each hardware module is described in VHDL.

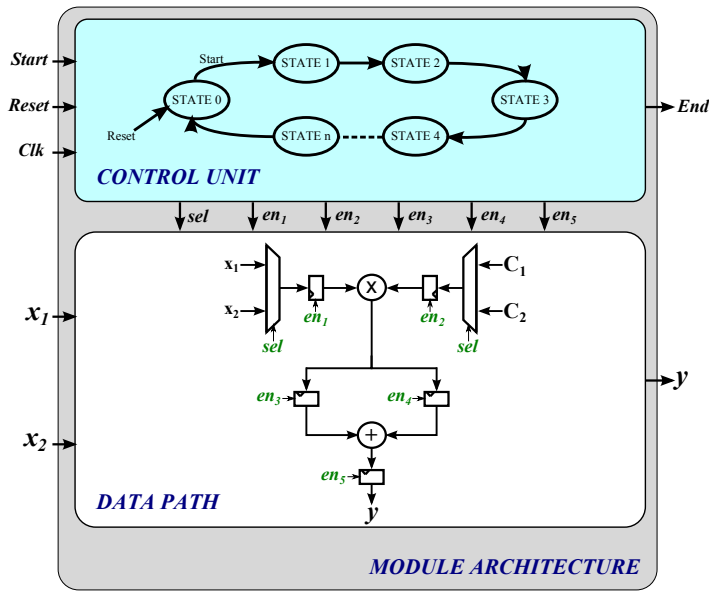


Figure 2.26 Module FPGA Architecture - General Structure.

4. Time-Area Performances

Once the FPGA architecture is validated and described in VHDL, it is synthesized via the map, place and route tools of the manufacturer.

An area analysis is then carried out. It gives a summary of the necessary hardware resources for implementing the developed architecture.

Four cases may occur [75]:

- the time-area constraints are not satisfied. In this case it could be necessary to change the platform if further algorithm optimizations are not possible.

- the time constraint is fulfilled and the area constraint is not satisfied. The designers can choose between different solutions depending on the application requirements. If the application makes possible an increase of cost, an another FPGA with more resources could easily lead to the fulfilment of the area constraint. In the case of a low-cost application, other solutions have to be performed. It is possible increase the level of factorization or re-optimize the algorithm, as well as reduce the word length at the cost of a lower accuracy.

- the area constraint is fulfilled but the time constraint is not satisfied. In this case the operating clock frequency can be increased or the factorization level can be reduced.

- both the constraints are satisfied. Only in this case the architecture is considered as appropriate and then the physical implementation can be achieved.

Experimental Validation

This is the last step of the design process. It consists in making the last validation of the developed controller. Before to make the final experimental validation an Hardware In the Loop (HIL) procedure is used [90]-[91]. It consists in developing both the digital controller and an emulator of the plant. This is an intermediate validation between a fully-software validation (Matlab/Simulink) and a fully experimental validation.

2.4 Conclusions

In this chapter the photovoltaic system has been presented. An in-depth discussion has been made regarding the relation between the perturbation time and the MPPT performances. Thus, it has been demonstrated that identification techniques can significantly increase the MPPT algorithm performances by identifying on-line the PV system parameters. Moreover, the identification results can be used for monitoring or diagnosis processes. Among the identification procedures, two identification techniques have been described: the cross-correlation method and the dual kalman filter. These two identification approaches are really different (being one non-parametric and the other one parametric) and then their main differences have been pointed out. The high speed performance and high integration density requirements have leaded to choose a digital platform based on FPGA devices, then an overview of the FPGA structure has been made. The design methodology has been illustrated as well as the PV system specification. In the next Chapter the identification algorithm development will be treated and its validation and robustness will be verified. In the fourth chapter the whole adaptive MPPT architecture, based on both identification techniques, will be developed. Finally, in the fifth chapter, experimental validations of the proposed adaptive controller in a real PV application will be carried out.

Chapter 3

Fully FPGA-based Implementation of Adaptive Digital Controller for PV Applications - Algorithm Development

3.1 Introduction

This chapter is aimed at affording the algorithm development step. Once the specifications for the PV system are given, the development and the validation of the adaptive MPPT controller algorithms must be done. As explained above, the majority of the complexity of the proposed adaptive control is located in the identification modules. As a consequence, the main task during this step will be to develop and validate the identification algorithms and to prepare them for the digital implementation.

In this chapter the common P&O MPPT algorithm is firstly reviewed. After that, the adaptive MPPT algorithm based on CCM is presented and, in particular, the CCM algorithm development is afforded. Then, the adaptive controller based on DKF is discussed. Finally, conclusions are drawn.

3.2 Basic MPPT Controller

The basic MPPT controller algorithm is now reviewed. This is the common P&O algorithm implemented for controlling the PV systems. At this stage, no adaptive process is included. Figure 3.1 is showing the synoptic of this controller.

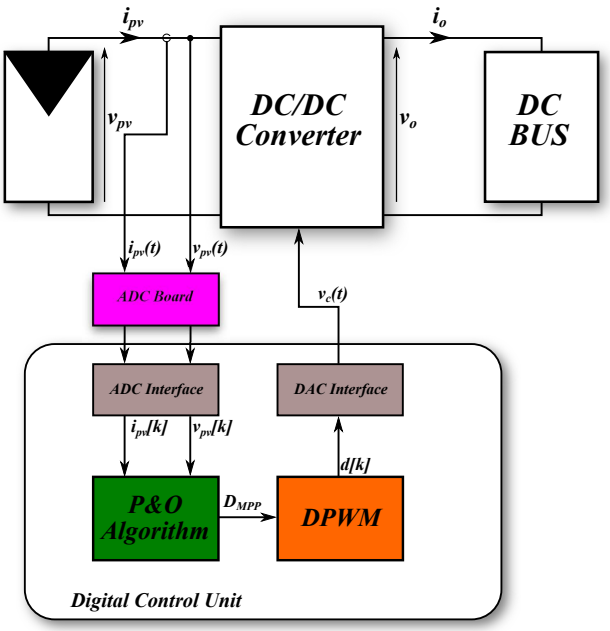


Figure 3.1 Synoptic of the basic MPPT Algorithm

3.2.1 P&O Algorithm

The P&O algorithm is based on a periodical perturbation of the PV source operating point by means of a perturbation of amplitude Δd applied to the duty cycle of the dc/dc converter control signal. After each perturbation, if the power extracted from the PV source, $P((k-1)T_p)$, has increased with respect to the PV power extracted before the perturbation, $P(kT_p)$, this means that the operating point has moved towards the MPP. Therefore, the next perturbation is applied with the same sign of the former one. Otherwise, the sign of the perturbation

is changed [92]. Thus, the control law describing the P&O MPPT algorithm is:

$$D((k+1)T_p) = D(kT_p) + \Delta D((k+1)T_p) \tag{3.1}$$

where

$$\begin{aligned} \Delta D((k+1)T_p) &= \Delta D(kT_p) \cdot \Delta S(kT_p) \\ \Delta D(kT_p) &= \Delta d \cdot sign\left(D(kT_p) - D((k-1)T_p)\right) \\ \Delta S(kT_p) &= sign\left(P(kT_p) - P((k-1)T_p)\right) \end{aligned} \tag{3.2}$$

In the equations 3.1 and 3.2, D and P represent the duty cycle and the PV power respectively, the latter being evaluated by means of the measured PV voltage $v_{pv}[k]$ and PV current $i_{pv}[k]$. The behaviour of the equation 3.1 can be described as shown in Table 3.1.

Table 3.1 Decision Table of the P&O algorithm

$\Delta S(kT_p)$	$\Delta D(kT_p)$	$\Delta D((k+1)T_p)$
+	+	+
+	-	-
-	+	-
-	-	+

By substituting "$+$" with "1" and "$-$" with "0" in Table 3.1, it is easy to note that the decision table of the P&O algorithm is equivalent to the truth table of the XNOR function then the algorithm is developed as shown in Figure 3.2.

D_{max} and D_{min} are the upper and lower limit of the duty-cycle, respectively. They are fixed by the designer in order to ensure a proper operating of both the DC/DC converter and the P&O MPPT technique. Δd is the step amplitude and the algorithm is repeated each T_p. These two parameters are usually fixed, based on the worst case analysis.

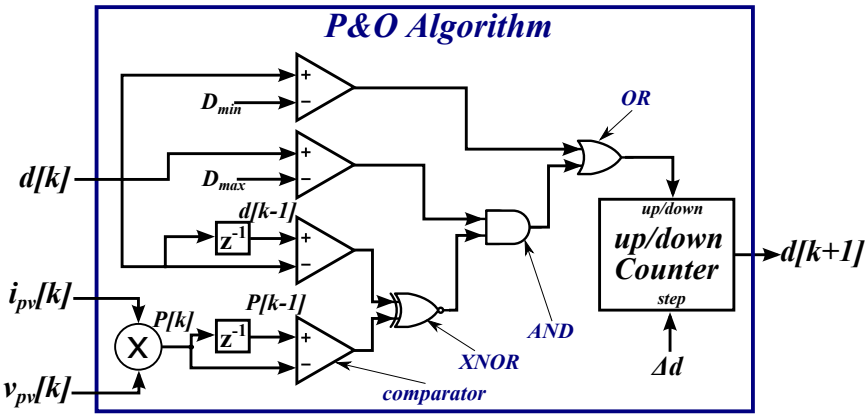


Figure 3.2 Synoptic of the implemented P&O algorithm

3.2.2 DPWM module

The DPWM module is in charge of generating the switching signals for the dc/dc converter. Different DPWM techniques have been developed in order to increase the carrier frequency or reducing the power consumption, without deteriorating the DPWM resolution. For instance, in [93] an hybrid DPWM is presented where a high switching frequency, up to 2 MHz, is guaranteed. Other techniques have also been proposed, such as the Delay-line, the Segmented Delay-Line, the Ring-Oscillators or the Delta-Sigma DPWM [94]-[95]-[96].

In this work, thanks to its simplicity, a common DPWM technique is adopted. It consists in generating the switching signals by comparing a sawtooth signal, called carrier signal, with the duty cycle imposed by the MPPT algorithm, as shown in Figure 3.3.

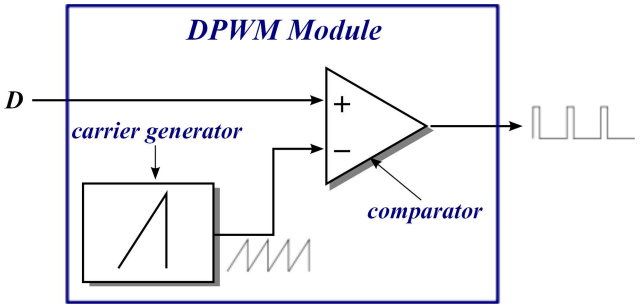


Figure 3.3 Block Diagram of the DPWM Module

Using a 9-bit counter with a clock frequency equal to 100 MHz, a switching frequency of 195 kHz is achieved.

3.3 Adaptive MPPT Controller based on the Cross Correlation Method

In this section the adaptive MPPT controller based on the CCM is presented. As shown in Figure 3.4, the PV system is controlled by the adaptive digital controller composed of the P&O algorithm, the Digital Pulse Width Modulator (DPWM), the T_p Calculation block and the CCM algorithm.

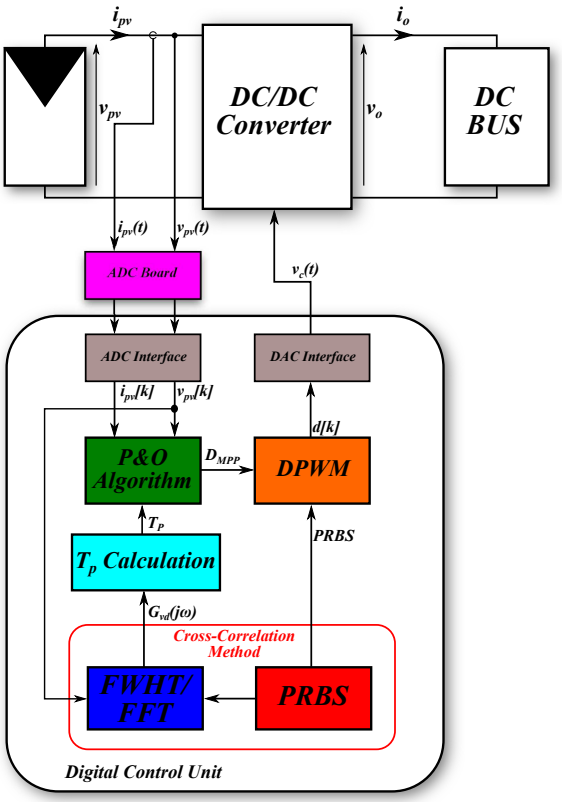


Figure 3.4 Synoptic of the Adaptive MPPT Algorithm based on Cross-Correlation Method

The P&O algorithm and the DPWM module have been explained in the previous sections. Thus, the development of the CCM algorithm is now made and the T_p Calculation block is presented in Section 3.3.5. It calculates the optimal perturbation period by using the parameters

extrapolated from the identified transfer function.

According to the proposed design methodology presented in Section 2.3.3, the algorithm development of the CCM consists of the following steps:

1. Modular Partitioning (Section 3.3.1)
The aim of this step is to divide the CCM algorithm into reusable and more manageable modules of different granularity levels.

2. FWHT/FFT Development (Section 3.3.2)
This is the identification core for the cross-correlation algorithm. It consists in developing the algorithm in charge of performing the FWHT and the FFT. Moreover an in-depth study concerning the choice of PRBS amplitude is carried out.

4. Algorithm Digital Realization (Section 3.3.3)
This step consists in quantizing the developed algorithm. Thus, the proper fixed-point data format for coefficients and variables has to be chosen.

5. Algorithm Optimization (Section 3.3.4)
In the CCM case, this step consists in increasing the quality of the identification results. Three types of optimizations can be performed: impulse response truncation, filtering and smoothing techniques.

6. Algorithm Validation (Section 3.3.6)
Once the identification algorithm has been developed, its functional validation has to be carried out.

3.3.1 Modular Partitioning

The modular partitioning consists in dividing complex algorithms into reusable, independent and simpler sub-algorithms, called modules. Therefore, in the developed adaptive controller, the CCM algorithm is divided into sub-algorithms leading to four hierarchy levels as shown in Figure 3.5. The lowest level is composed of both arithmetic and logic operators. They can be considered as fine-grain operators [75]-[85]. The second level contains modules included in the FWHT and the FFT transformations, as the ButterFly blocks (BFI, BFII), the smoothing technique (explained in Section 3.3.4), the ordering blocks and the complex multiplier. At the third level of hierarchy heavy-grain modules can be found, i.e. the FWHT/FFT module and the PRBS generator. Finally, the whole CCM algorithm is placed into the highest hierarchy level.

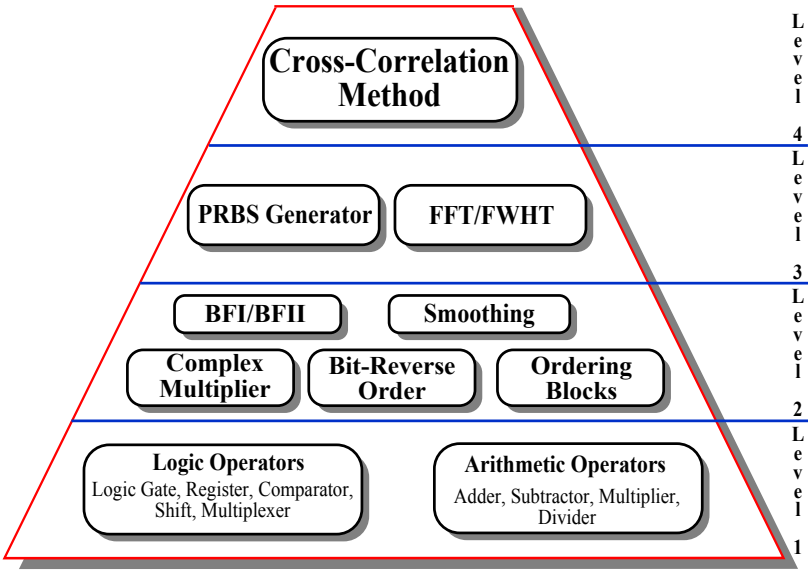


Figure 3.5 Modular Partitioning of the developed cross-correlation method.

3.3.2 FWHT/FFT Development

The FWHT/FFT module is the core of the identification technique. Figure 3.6 shows the block diagram of the developed FWHT/FFT module. An input *sel* selects which transformation has to be performed. When it is equal to 0, the FWHT is performed. According to the equation 2.33, the acquired panel voltage array $v_{pv}[n]$ is firstly re-ordered according to the matrix P_t. Then, the ordered data, as well as the selection signal, has sent to the transformation block, which executes the H-transform. The transformed data is then re-ordered, according to the matrix P_g^T, giving the identified impulse response $h[n]$. On the other hand, when the *sel* signal is equal to 1, the FFT is carried out. The just identified impulse response is the input of the transformation block. The FFT is then executed and the result is sent to the bit-reverse order block, resulting in the system frequency response, described by the real part $y_{re}[n]$ and the imaginary part $y_{im}[n]$ of the discrete transfer function $G_{vp,d}(j\omega_n)$.

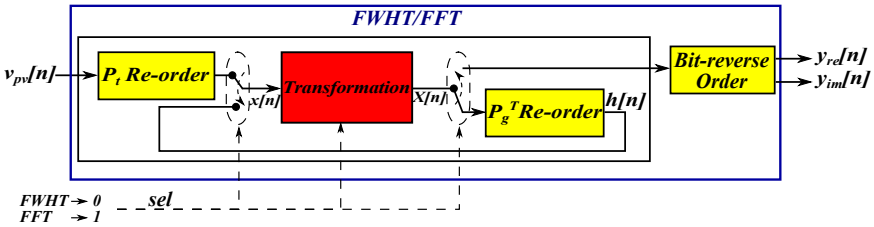


Figure 3.6 FWHT/FFT Block.

The transformation block can be completely developed in parallel involving a very short execution time. However, this solution leads to a very high consumption of the hardware resources, preventing from using a low-cost FPGA. As a consequence, a fully sequential algorithm is used. It allows reducing significantly the consumption of the FPGA resources at the cost of a larger execution time. Thus, in order to reach a compromise between the execution time and the use of FPGA resources, a Radix-2^2 FFT algorithm [97] has been developed and, then, the transformation block has been implemented in a pipeline way [98]-[99]-[100] (see Figure 3.7).

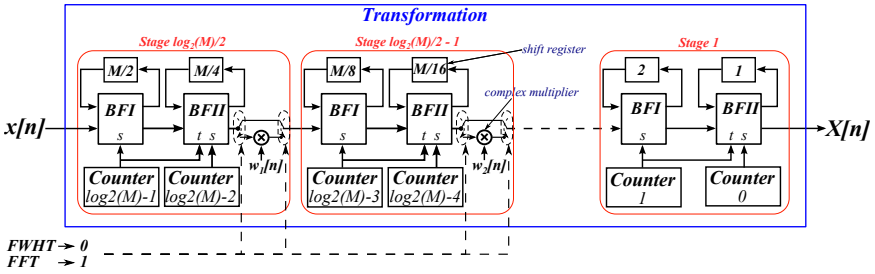


Figure 3.7 Transformation Block.

It consists of different stages composed of two shift registers, two counters to respect the synchronization, one complex multiplier used only for the FFT and two main blocks: the ButterFly I (BFI) and the ButterFly II (BFII), presented in Figure 3.8. In the FWHT case, the input signal $x[n]$ is the acquired panel voltage, being a real digital sequence. Any complex multiplier is present then the result is still real. On the other hand, when the FFT is executed, the input signal $x[n]$ is the impulse response. The latter is an array of real samples, but, in this case, complex multiplications are executed between the signal within the transformation block and the twiddle factor $w_k[n]$. Therefore, the result will be complex.

The number of stages in the transformation block depends on the PRBS length M. A longer sequence involves more stages and then a bigger complexity. It is worth to note that the PRBS length cannot be changed without changing the whole block. Then, the FWHT and the FFT are strongly related with the chosen PRBS sequence. For this reason, the PRBS length selection is now investigated, as well as the PRBS amplitude, in order to fix the PRBS sequence and to develop consequently the FWHT/FFT block. Moreover, the length and the amplitude of PRBS sequence influence the identification accuracy, as explained in Chapter 2. Therefore, a compromise has to be reached between the identification accuracy and the algorithm complexity.

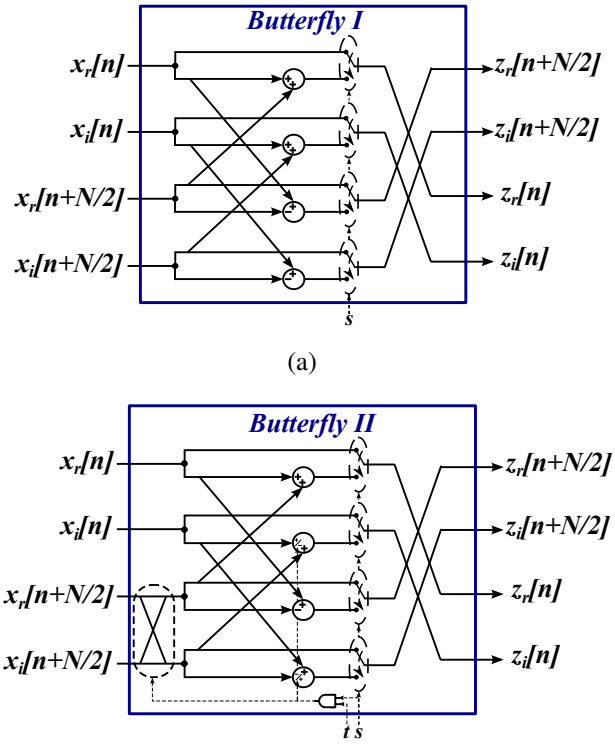


Figure 3.8 Butterfly Blocks: a) BFI b) BFII

PRBS length Selection

The PRBS length is the main important aspect. From a functional point of view, the PRBS length has to guarantee that the impulse response of the system decays toward zero within one period of the PRBS [101]. Moreover, a longer PRBS sequence is close to a white noise signal and then guarantees a better identification, as well shown also in equation 2.22. However, the algorithm complexity and the identification time increase strongly with the PRBS length.

A compromise is reached choosing a MLS PRBS length M equal to 1023. This choice allows a good accuracy with an acceptable algorithm complexity. Moreover, it is able to fulfil the time requirement of the application under analysis (real-time identification) by fixing the time needed to inject the PRBS sequence, as will be shown in Chapter

5. This choice means that the internal vectors ($v_{pv}[n]$, $x[n]$, $X[n]$, h[n]) in this block are of 1024 elements.

PRBS Amplitude Seletion

In order to evaluate the effect of the PV non-linearity, the system shown in Figure 3.4 has been previously simulated in PSIM, with the system specifications given in Chapter 2. The PSIM simulation allows predicting the effects of the PV array non linearity on the applicability of the identification technique proposed in this work. The possibility of implementing in PSIM some user-defined blocks, written in C code, allows the simulation of the whole identification procedure before implementing it in the FPGA device. Thus, the non-linear model of the PV array, the identification technique and the P&O MPPT algorithm have been simulated as a whole in PSIM environment.

The transfer function $G_{vp,d}$, evaluated via the CCM for different values of the PRBS amplitude, has been compared with the corresponding one obtained by means of the linearized model. The results shown in Figure 3.9 allow to draw the conclusion that a PRBS amplitude not larger than 0.06 is mandatory in order to obtain a good estimation of the $G_{vp,d}$ transfer function. Figure 3.9 shows the Bode plot obtained by letting the PV array operating at its MPP, thus where the non-linearity of the model is the most sensitive effect [1]. In the practical implementation of the CCM the PRBS amplitude must be designed also according to the signal-to-noise ratio of the acquisition system.

Table 3.2 summarizes the parameters of the PRBS signals used in the examples proposed in this thesis. Thus, the chosen PRBS length involves 5 stages in the transformation block.

Table 3.2 Adaptive P&O MPPT Controller Parameters

Parameter	Value
PRBS length M	1023
PRBS amplitude e	0.03125
Injection frequency f_{inj}	195 kHz
Frequency resolution $\frac{f_{inj}}{M}$	190 Hz
Number of PRBS injection	2

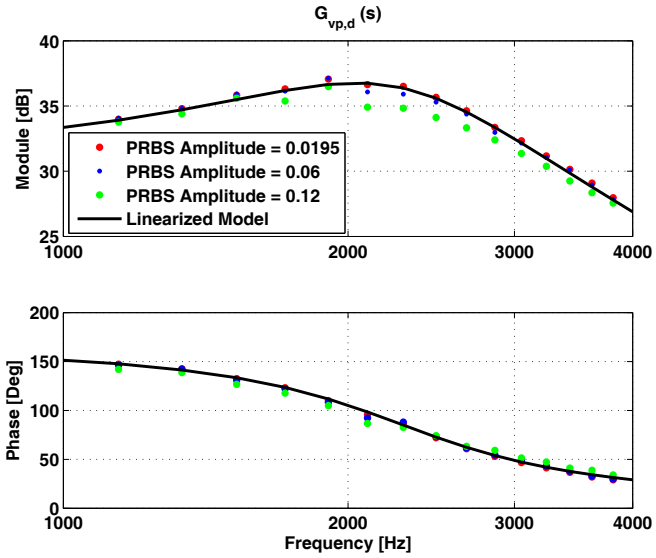


Figure 3.9 Evaluation of the transfer function by means of the CCM and the FFT algorithm by using different PRBS Amplitudes. The frequency range has been limited to [1 kHz,4 kHz] in order to highlight the discrepancies between the different obtained transfer functions.

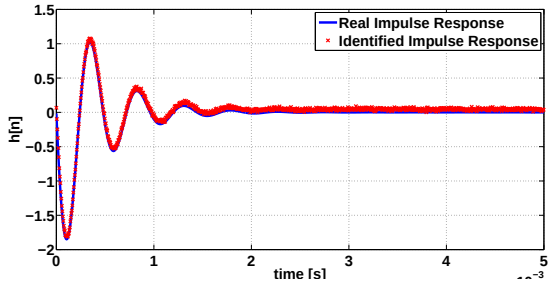
3.3.3 Digital Realization

The aim of this step is to prepare the developed algorithm for the implementation in the FPGA device. To this aim the fixed-data format has to be chosen accurately. The choice of the format is of prime importance because it influences directly the identification accuracy and the needed FPGA resources. On one hand, a high format involves very good accuracy at the cost of a heavier FPGA architecture. On the other hand, a smaller data length reduces the consumed resources causing a lower precision of the identification. Thus, a compromise has to be reached. Moreover, the choice of the format has to guarantee no overflow in the FFT/FWHT block.

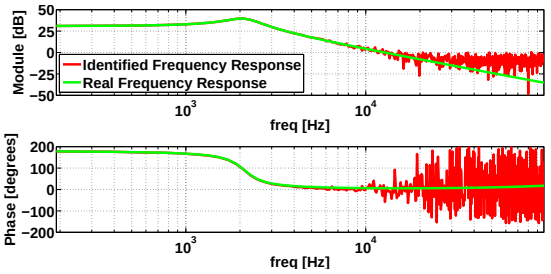
After several tests, a 20-bit fixed-point format for the CCM algorithm and a 12-bit fixed-point format for the MPPT algorithm are adopted.

3.3.4 Algorithm Optimization

The ADC effect and the noise in the system affect the measurement, causing the deterioration of the identified response. In order to compare the identified transfer function with the real one, the real and imaginary parts, obtained by using the CCM technique, are converted in module and phase. Figure 3.10 shows the identified impulse and frequency responses supposing in the simulated system an ADC with a voltage resolution equal to 40 mV and a PV differential resistance equal to 5 Ω. The error at high frequency is due to the ADC effect, which reduces the Signal-to-Noise Ratio (SNR).



(a)



(b)

Figure 3.10 Identified Response without any algorithm optimization: a) Impulse Response b) Frequency Response

In order to quantify the goodness of the identified impulse and frequency responses, the Normalized Root Mean Square Error (NRMSE), expressed as a percentage, can be adopted.

$$NRMSE_{\%} = \frac{1}{\Delta y}\sqrt{\frac{\sum_{k=1}^{K}(y[k]-\hat{y}[k])^2}{K}} \cdot 100 \tag{3.3}$$

where y, $\widehat{y}$, Δy and K are the actual response, the identified response, the range of y and the number of samples, respectively. A bigger NRMSE means poor identification results. It is proportional to the ADC resolution and to the inverse of the PRBS amplitude. In this simulation, the NRMSE for the impulse response is *1.65 %*, for the frequency response magnitude is *0.31 %* and for the frequency response phase is *49.6 %*. The large error on the phase is due to the low SNR at high frequency.

Therefore, in order to increase the quality of the identified responses without increasing the PRBS amplitude, several techniques can be adopted [101]:

- Pre-emphasis and De-emphasis filters
- Impulse response truncation
- Smoothing technique

Pre-emphasis and De-emphasis Filters

The ADC effect can be reduced by using digital filters: the idea is to pre-emphasize the PRBS before its injection into the system, in order to make it less susceptible to the noise introduced by the quantization effect. As a consequence, the high frequency SNR decreases. Later, a de-emphasis filter, i.e. the inverse of the pre-emphasis one, must be used to remove the shaping introduced by the latter. The transfer function of the pre-emphasis filter is:

$$F(z) = K(1 - z_1 \cdot z^{-1}) \tag{3.4}$$

where K is the filter gain and

$$z_1 = e^{\frac{-2\pi f_{z_1}}{f_{sw}}} \tag{3.5}$$

In (3.5), f_{z_1} represent the filter corner frequency, whereas f_{sw} is the converter switching frequency. An automatic procedure of evaluation of f_{z_1} and K has been carried out and confirmed by simulations: at the beginning the frequency response is identified without any filter; later f_{z_1} and K are evaluated according to the following criteria:

1. the output must be boosted above the noise floor introduced by the ADC;

2. the duty cycle must not saturate;

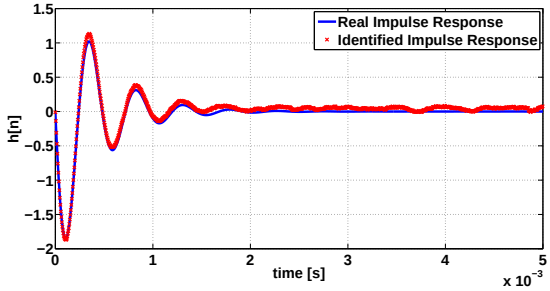
3. the PV voltage must not exceed a fixed regulation window.

Thus:

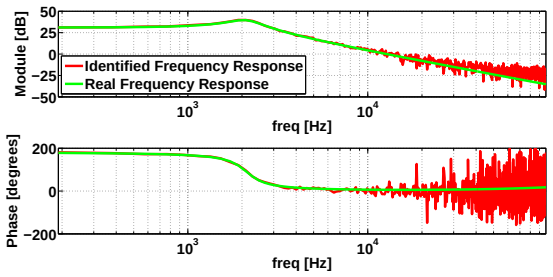
- f_{z_1} is frequency where the magnitude of the frequency response drops below q_{ADC}/e, where q_{ADC} and e are the ADC resolution and the PRBS amplitude, respectively;

- K can be calculated by using the following relation:

$$K = min\left(\frac{1}{1-z_1} \quad ; \quad \frac{\Delta d_{sat}}{||F_{K=1}(-1)||}\right) \tag{3.6}$$



(a)



(b)

Figure 3.11 Identified Response applying the pre-emphasis and de-emphasis filters: a) Impulse Response b) Frequency Response

Finally the identification is carried out by using also the pre-emphasis and de-emphasis filters. In Figure 3.11 the effects of the filters on the impulse and frequency response can be seen. Although SNR is increased, the response is still affected by random noise at high frequencies. The corresponding NRMSE is 0.546 % and 42.14 % for the module and phase of the transfer function, respectively.

Impulse Response Truncation

In order to reduce the random noise at high frequencies an impulse response truncation can be adopted. For a long PRBS injection period, the system dynamic is completely extinguished into the first part of the identified impulse response, while the second part just contains noise components, as shown in Figure 3.10(a). This noisy part of the response can be eliminated by truncating the impulse response, decreasing also the high frequency distortion of the frequency response. Several tests are carried out with different window and the correspondent NRMSE is calculated, as shown in Table 3.3.

Table 3.3 RMSE with different truncation window

Window	NRMSE	$NRMSE_{mag}$	$NRMSE_{phase}$
Rectangular	1.18 %	0.26 %	45.43 %
Hann	1.05 %	0.28 %	43.92 %
Hamming	1.08 %	0.27 %	44.33 %
Gauss σ=0.5	1.12 %	0.27 %	44.77 %
Gauss σ=0.3	0.98 %	0.42 %	43.27 %
Gauss σ=0.1	2.43 %	1.88 %	38.49 %
Bartlett	1.06 %	0.36 %	43.86 %
Blackman	0.99 %	0.35 %	43.37 %
Kaiser β=5	1.08 %	0.28 %	44.34 %
Kaiser β=10	0.98 %	0.38 %	43.32 %
Kaiser β=20	1.04 %	0.63 %	42.63 %

This analysis depends strongly on the damping factor. Indeed, a high damping factor reduces the settling time and then the truncation windows with a narrow main lobe become more effective. In general, the best results are achieved by using either Kaiser or Gauss windows; however, these solutions require complex algorithms and then more required resources. Therefore, a simple rectangular window is adopted

in order to keep a great simplicity. Figure 3.12 shows the identified results adopting the rectangular window.

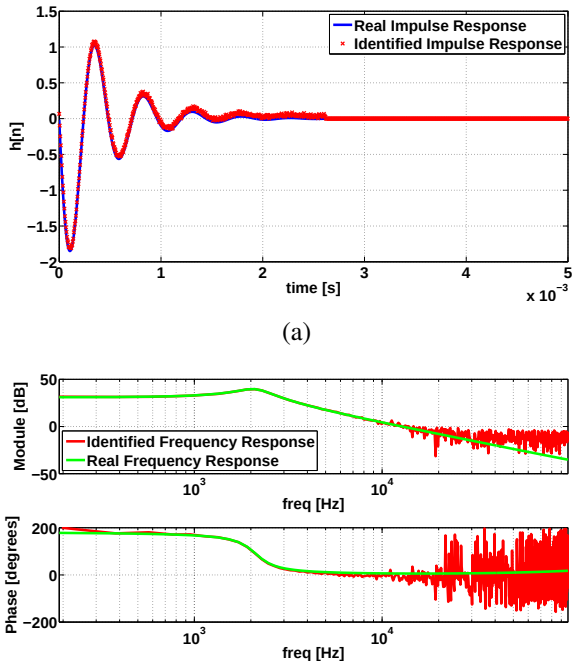


Figure 3.12 Identified Response applying a rectangular window: a) Impulse Response b) Frequency Response

Smoothing Technique

Another improvement of the identified frequency response can be achieved using a smoothing technique, i.e. an adaptive moving average of the linearly spaced frequency response data. The averaging window must be narrow at low frequencies in order to avoid a smoothing of the true resonant peak and large at high frequencies, to smooth the random noise effect. In audio applications [102], an usual choice of the window size $W(f)$, expressed in terms of sample number, can

be obtained by solving the following system:

$$\begin{aligned} b &= S \cdot \frac{\Delta f}{f_{-90}} \\ W(f) &= floor\left(b \cdot \frac{f}{\Delta f}\right) \end{aligned} \tag{3.7}$$

where S is a scaling factor, Δf is the frequency resolution and f_{-90} is the frequency where the phase drops of -90. Several simulations are carried out with different values of scaling factor; the results are reported in Table 3.4 in terms of NRMSE. Figure 3.13 shows the identified frequency response using a smoothing technique with scaling factor equal to 3, 6 and 12. As can be seen, a good compromise can be reach by choosing a scaling factor equal to 6.

Table 3.4 RMSE with different scaling factors

S	$NRMSE_{mag}$	$NRMSE_{phase}$
3	2 %	9.9 %
6	2.3 %	9.7 %
12	5.2 %	9.5 %

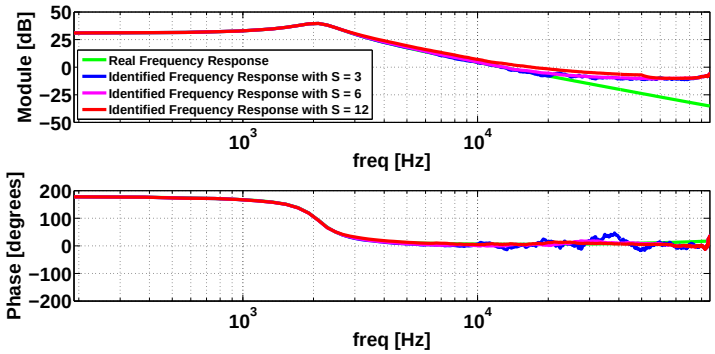


Figure 3.13 Identified Frequency Response applying a smoothing technique with scaling factor equal to 3, 6 and 12.

These three techniques are introduced to improve the identified responses. Using all of them, the identified frequency response is shown in Figure 3.14. However, the filters need a prior identified frequency

response in order to calculate their coefficient. This means that the CCM has to be performed twice: one time without filters and another time applying the filters. Moreover, the obtained improvement thanks to the filters does not affect the part of the frequency response necessary to calculate the system settling time. Therefore, in order to reduce the identification time, it has been chosen to adopt only the truncation and the smoothing technique obtaining a NRMSE for the magnitude and for the phase equal to 2.3 % and 7.9 %, respectively. Indeed, these two techniques do not require any additional injection of PRBS and they are able to improve the identified responses by enhancing the estimated accuracy of the parameters. Figure 3.15 shows the identification result.

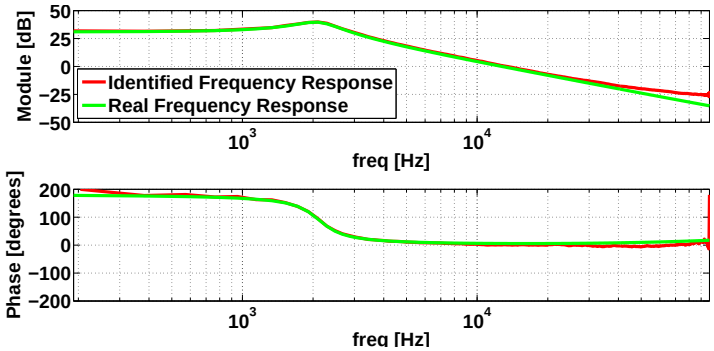


Figure 3.14 Identified Frequency Response applying the filters, the impulse response truncation and the smoothing technique with scaling factor equal to 6.

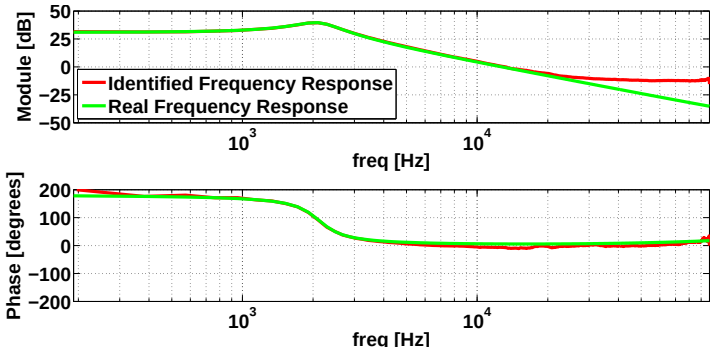


Figure 3.15 Identified Frequency Response applying the impulse response truncation and the smoothing technique with scaling factor equal to 6.

3.3.5 T_p Calculation block

This block allows calculating the PV system settling time T_ε from the identified real and imaginary parts of the identified system frequency response, $y_{re,s}[n]$ and $y_{im,s}[n]$ respectively. Thus, the optimal perturbation period is evaluated and imposed to the P&O algorithm.

According to (2.9) and (2.43), T_ε can be rewritten as:

$$T_\varepsilon = -\frac{2 \cdot \left|G_{vp,d}(j\omega_n)\right|}{\left|G_{vp,d}(j0)\right| \cdot \omega_n} \cdot ln\left(\frac{\varepsilon}{2}\right) \tag{3.8}$$

Thus, the natural pulsation ω_n, the DC gain $|G_{vp,d}(j0)|$ and the modulus at ω_n $|G_{vp,d}(j\omega_n)|$ of the system frequency response are evaluated from $y_{re,s}[n]$ and $y_{im,s}[n]$. Indeed, ω_n is the pulsation at which the real part is close to 0:

$$\omega_n = i_{\omega_n} \cdot \frac{2 \cdot \pi \cdot f_{sw}}{M} \tag{3.9}$$

where i_{ω_n} is the position where the real part vector crosses 0 and $\frac{f_{sw}}{M}$ is the frequency resolution. The corresponding value of the imaginary vector in i_{ω_n} is corresponding to $|G_{vp,d}(j\omega_n)|$. $|G_{vp,d}(j0)|$ is approximated as the first value of the real part, i.e $y_{re,s}[0]$. Subsequently, the system settling time, and thus the optimal perturbation period, can be derived as shown in (3.10).

$$T_p = T_\varepsilon = \left\{ -\frac{M}{f_{sw} \cdot \pi} \cdot \ln\left(\frac{\varepsilon}{2}\right) \right\} \cdot \frac{|G_{vp,d}(j\omega_n)|}{|G_{vp,d}(j0)| \cdot i_{\omega_n}} \tag{3.10}$$

where f_{sw} is the switching frequency of the power converter, M is the PRBS length and ε is the relative band within which the PV power has to be confined in order to consider the system in steady-state [103]. In equation 3.10, the value between the brackets is known, thus it can be considered as a constant coefficient, named C. Figure 3.16 shows the block diagram of the T_p Calculation Module.

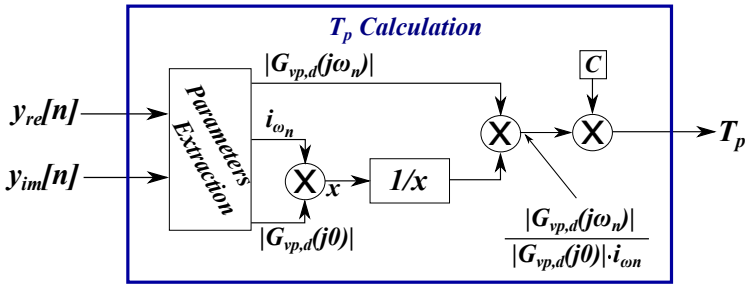


Figure 3.16 T_p Calculation Module

3.3.6 Algorithm Validation

At this stage the proposed Cross-Correlation method has to be validated. Figure 2.16 shows the model simulated in Matlab/Simulink with the nominal cell parameters and nominal boost parameters described in Chapter 2. The nominal differential resistance r_d is equal to 5 Ω and the duty cycle is fixed to 0.5.

Validation of Parameter Estimation

The impulse response truncation and the smoothing technique with scaling factor equal to 6 are adopted in order to improve the identification results as shown in Section 3.3.4. In the same section the identified frequency response has been compared with the real one and then the CCM has been validated (see Figure 3.15). The needed parameters for the settling time calculation are extrapolated to the identified real and imaginary parts of the system transfer function as explained in Section 3.3.5. Table 3.5 compares the estimate of the DC gain, the natural frequency, the damping factor and the settling time with the real values giving the corresponding percentage error.

Table 3.5 Adaptive P&O MPPT Controller Parameters

Parameter	Real Value	Estimated Value	Percentage Error
$G(j0)$	-35.29 V	-35.10 V	0.5 %
ω_n	13.3 krad/s	13.1 krad/s	1 %
ζ	0.186	0.186	0.06 %
T_ε	1.49 *ms*	1.51 *ms*	1 %

The estimated parameters are very close to the real ones then the adaptive procedure should work properly.

Robustness of Identification Procedure

Up to now the CCM and the parameter estimation have been validated with the nominal system parameters. The robustness of the proposed method is investigated in this section. The CCM does not require the previous knowledge of the system model, so that the algorithm can be applied to a large class of DC/DC converters, even whenever the second order approximation does not hold. Instead, the settling time estimation method is based on the use of (2.9), requiring a second order model. In the application under study, the second order behaviour of the PV system has been demonstrated in Section 2.2.2 and then the T_ε estimation remains valid.

First of all, a variation of the differential resistance r_d is supposed keeping constant the other converter parameters. r_d is the quantity that is subject to potentially high variations. Three cases are considered: $r_d = 2\ \Omega$, $r_d = 50\ \Omega$ and $r_d = 200\ \Omega$.

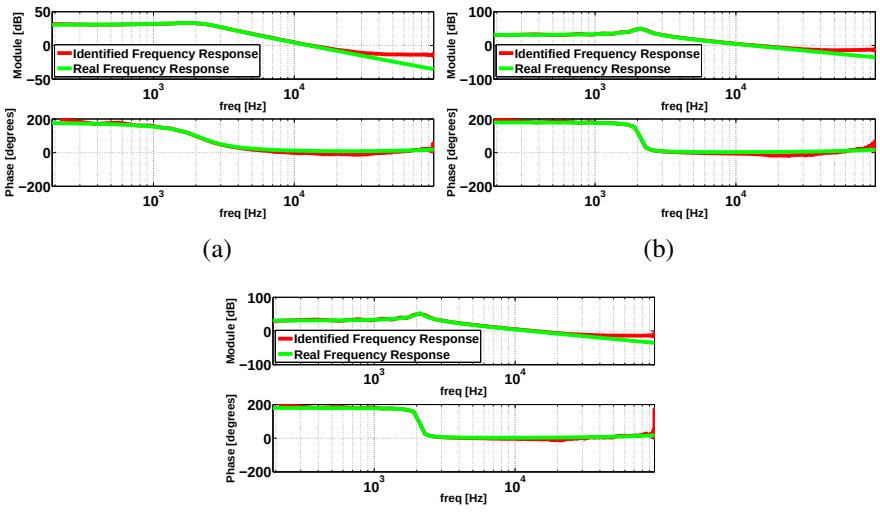


Figure 3.17 Identified Response with different differential resistance value: a) $r_d = 2\ \Omega$ b) $r_d = 50\ \Omega$ c) $r_d = 200\ \Omega$

Figure 3.17 shows the identified frequency response and the real one, whereas, Table 3.6 shows the estimated parameters and the corresponding percentage error.

Table 3.6 Estimated Parameters with different r_d values

	$r_d = 2\ \Omega$		$r_d = 50\ \Omega$		$r_d = 200\ \Omega$	
	Error	Error (%)	Error	Error (%)	Error	Error (%)
$G(j0)$	0.3 V	0.71 %	5.0 V	13.7 %	7.4 V	20 %
ω_n	300 rad/s	2.2 %	100 rad/s	0.12 %	0 rad/s	0.06 %
ζ	0.016	2.8 %	0.001	1.3 %	0	0.4 %
T_ε	4 μs	0.6 %	64 μs	1.2 %	42 μs	0.6 %

Several simulations are now made varying both the converter parameters and the differential resistance according to Table 3.7.

Table 3.7 Cases presented in this thesis

	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8
C [μF]	20	20	20	20	100	100	100	100
r_d [Ω]	2	2	40	40	2	2	40	40
L [μH]	50	160	50	160	50	160	50	160

The results are shown in Figure 3.18 where the identified frequency responses are compared with the real responses. The estimated parameters and their percentage errors are shown in Table 3.8.

Table 3.8 Estimation Results for the CCM robustness

		case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8
$\lvert G(j0)\rvert$ [V]	Real	34.3	34.3	35.9	35.9	34.3	34.3	35.9	35.9
	Est.	34.4	34.4	36.1	34.0	34.2	34.5	37.5	30.3
	Err %	0.29	0.42	0.51	5.20	0.32	0.63	4.57	15.5
ω_n [krad/s]	Real	32.3	18.0	31.7	17.7	14.4	8.1	14.2	7.9
	Est.	31.1	18.0	31.2	18.0	14.4	8.4	14.4	8.4
	Err %	3.6	0.5	1.6	1.6	0.5	3.8	1.6	6.0
ζ	Real	0.420	0.707	0.054	0.055	0.248	0.350	0.086	0.059
	Est.	0.407	0.740	0.058	0.060	0.246	0.396	0.097	0.104
	Err %	2.85	4.65	6.10	9.30	0.77	13.05	12.32	74.87
T_ε [ms]	Real	0.27	0.29	2.10	3.80	1.00	1.30	3.00	7.90
	Est.	0.29	0.27	2.00	3.40	1.01	1.10	2.60	4.20
	Err %	6.78	3.95	4.28	9.93	1.29	14.80	12.35	46.05

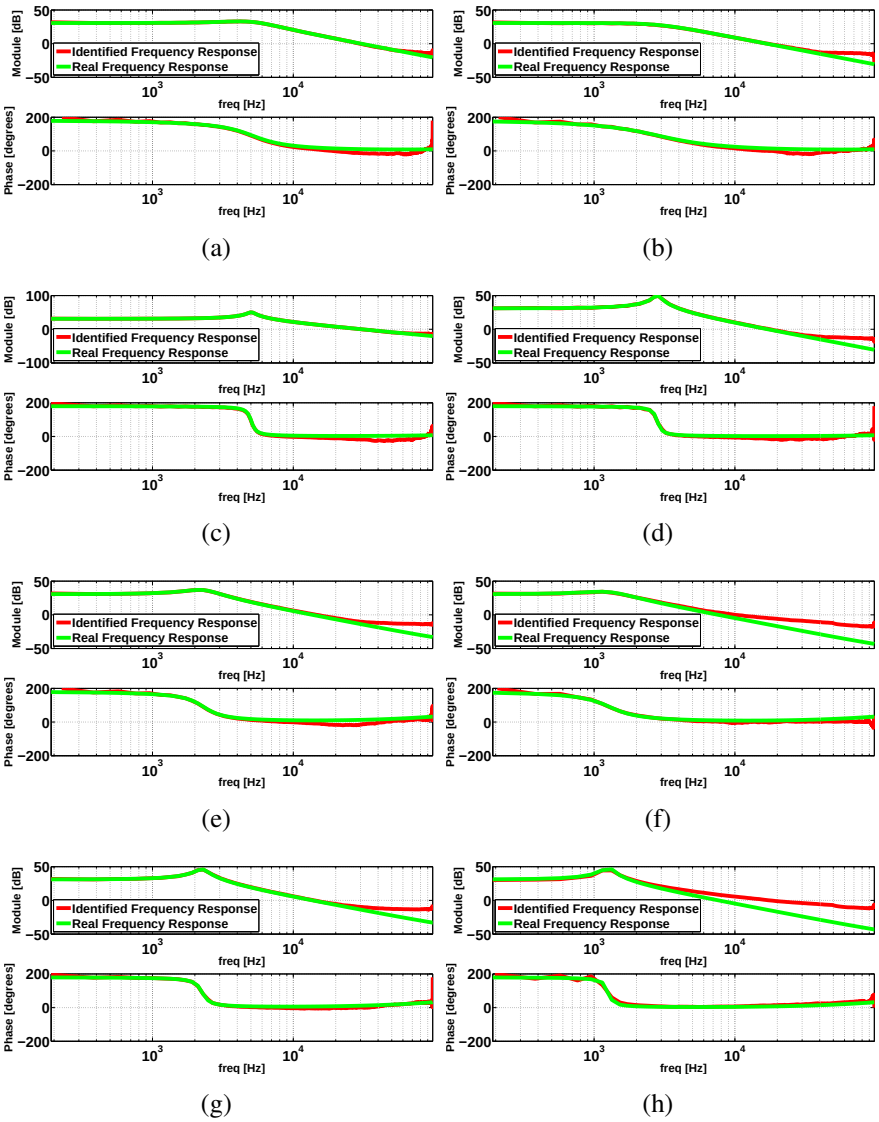


Figure 3.18 Identified Response with different parameters: a) case 1 b) case 2 c) case 3 d) case 4 e) case 5 f) case 6 g) case 7 h) case 8

In order to appreciate the robustness of the proposed method, the system has been simulated also in presence of an irradiance transient with a rate of $100W/m^2/s$, this value being the highest possible in the typical range reported in the EN50530 standard. The $G_{vp,d}(s)$ transfer function evaluated in steady-state and also in transient environmental conditions has been plotted in Figure 3.19. The two tests have been done around the same nominal operating point and with the same PRBS signal amplitude. The overlapping is almost perfect, demonstrating the intrinsic CCM immunity to irradiance variations. Moreover, because of the shortness of the time interval during which the PRBS is injected, the operating point does not change significantly even in presence of an irradiance variation, thus the PV non-linearity does not affect the CCM by assuring a good estimation of $G_{vp,d}(s)$.

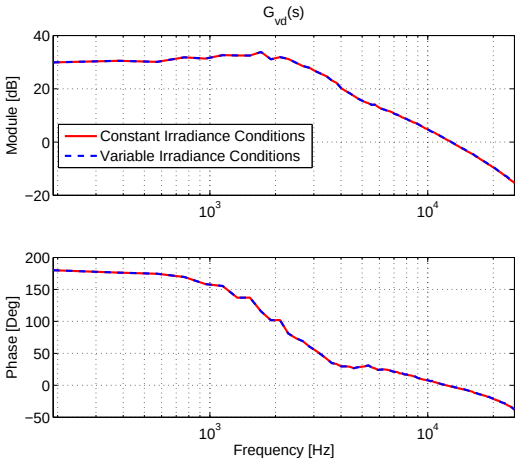


Figure 3.19 Transfer function evaluated by means of the CCM and the FFT algorithm in constant (red) and variable (dotted blue) irradiance conditions.

3.3.7 Conclusion

The CCM method and the proposed parameter estimation procedure have been validated. Their robustness has been verified by performing several simulations with different differential resistances and converter parameters. In all cases the percentage error of the parameter estimation is lower than 20 %, expect in the 8^{th} case where all the parameters are very far from their nominal values. The main issue in this last case is that one of the prerequisites of the CCM does not hold. Indeed, the impulse response is not decayed toward zero within half period of PRBS. Thus, in this case a different length of the PRBS is required.

The identification time of the CCM is fixed and known at the beginning, being equal almost to the time needed to inject two PRBS sequences (see Chapter 5). The identified responses are very close to the real one in each case. The advantage of the CCM lies in the fact that no knowledge about the system to be identified is required. The main drawback of this identification procedure is that it does not give any information about the confidence of the estimation. Moreover, the identified parameters can be underestimated or overestimated (see Table 3.8) without any knowledge about it. Thus, a safety margin, evaluated in the worst case, has to be introduced.

3.4 Adaptive MPPT Controller based on the Dual Kalman Filter

In this section the adaptive MPPT technique based on the DKF is presented. Figure 3.20 shows a synoptic of the whole adaptive controller composed of the P&O algorithm, the DPWM module and the DKF aimed at identifying the states and the parameters of the PV system. The optimal perturbation time is calculated by the T_p calculation block from the identified parameters.

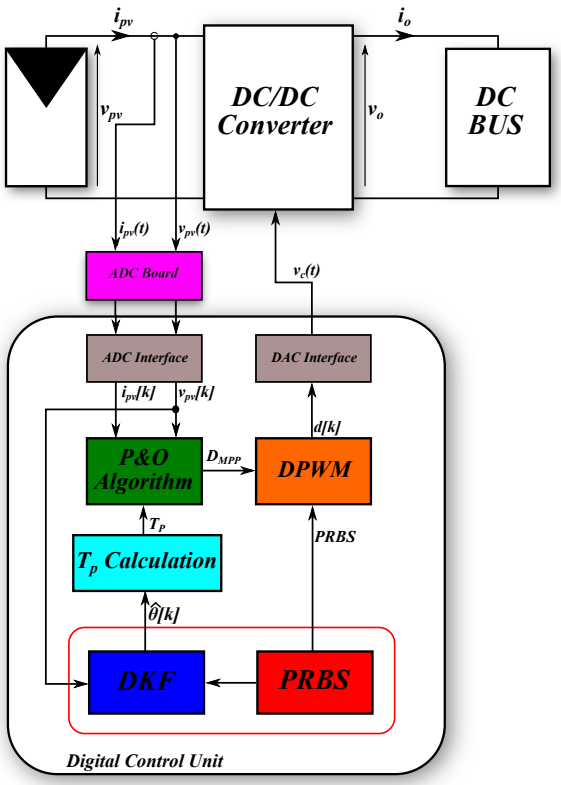


Figure 3.20 Synoptic of the Adaptive MPPT Algorithm based on Dual Kalman Filter

The P&O algorithm, as well the DPWM module, have been already presented in Section 3.2. The DKF algorithm is developed according to the following steps (see Section 2.3.3).

1. Modular Partitioning (Section 3.4.1)
The aim of this step is to divide the whole DKF algorithm into reusable and manageable modules of different granularity levels.

2. Model Selection (Section 3.4.2)
In this section the choice of the PV model to be adopted in the DKF is deeply discussed. This is the main step to achieve a high quality identification.

3. Model Discretization (Section 3.4.3)
The aim of this step is to discretize the previous continuous-time model in view of its digital implementation.

4. Algorithm Digital Realization (Section 3.4.4)
This step consists in normalizing and quantizing the developed DKF algorithm.

5. Algorithm Optimization (Section 3.4.5)
Some algorithm optimizations are used in order to reduce the algorithm complexity.

6. Algorithm Validation (Section 3.4.9)
Once the DKF algorithm has been developed, its last functional validation has to be carried out.

The input selection and the convergence criterion of the proposed DKF are discussed in Section 3.4.6 and 3.4.7, respectively. Moreover, in Section 3.4.8 the T_p estimation is discussed.

3.4.1 Modular Partitioning

The more complex algorithm in this second adaptive MPPT controller is surely the DKF process. It is divided into different re-usable and independent modules divided into distinct levels of granularity as shown in Figure 3.21. The basic level consists of the logic (Register, Shift, Multiplexer and so on) and arithmetic operators (Adder, Subtractor, Multiplier and Divider). The second level contains the indispensable modules for implementing the DKF (KG calculation, JM calculation, error covariance matrix calculation). The third level of hierarchy is constituted of the following modules: state prediction, state update and parameter update modules as well as the PRBS generator. Finally, the last level consists of the whole DKF algorithm.

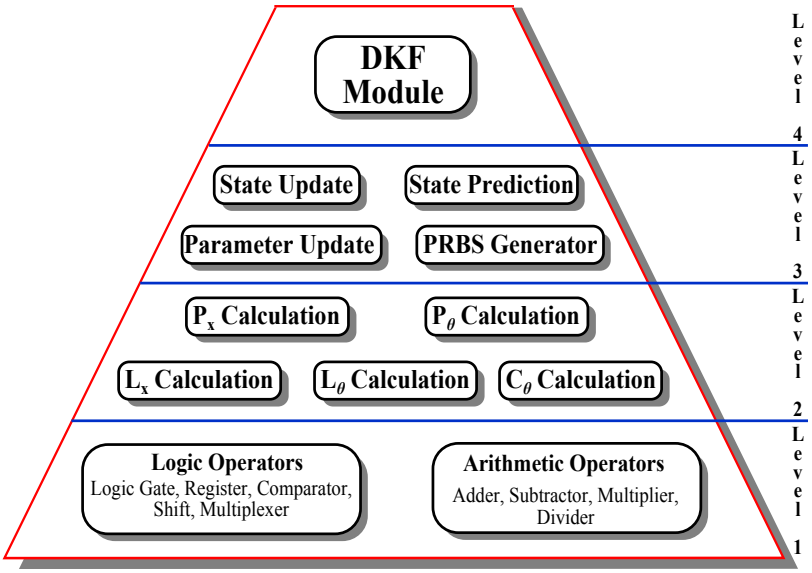


Figure 3.21 Modular Partitioning of the developed dual Kalman filter.

3.4.2 Model Selection

This is the corner-stone in any model-based identification technique. Indeed, it is essential to choose the best model structure of the system under analysis, since an inappropriate choice could lead to higher complexity, larger identification time and, in some cases, to the divergence of the identification technique.

First of all, the dynamical model of the sole PV generator has been considered for estimating the main PV source parameters. In this case, the output and the input of the model will be the PV voltage and the PV current, respectively. The main issue of this approach is related to the low value of the PV source capacitance, which involves a very fast dynamic of the source itself [34]. Thus, the PV states could be not observable in discrete-time by using the commercial low-cost ADCs with limited acquisition frequencies. Moreover, also if high-performance ADCs are adopted, the increase of the sampling frequency causes the reduction of the time during which the DKF algorithm must be able to predict the states and the parameters. Therefore, a high-performance FPGA must be adopted which impacts the cost.

Another approach that has been attempted is based on the identification of the whole PV system, i.e. the PV source and the dc/dc power stage. The single-diode model (see Figure 2.5) has been considered for modelling the PV source. The panel voltage and the converter duty-cycle will be the output and the input of the model, respectively. In this case the parameter vector θ should include too many unknown and variable parameters (photocurrent, series resistance, parallel resistance, converter inductance and capacitance with their parasitic resistances), thus involving a very complex DKF algorithm. Moreover, the single-diode model parameters vary too rapidly and too strongly with the irradiance, the temperature and the operating point, so that the hypothesis of quasi-constant parameters 2.47 is not verified and mistakes in the identification are likely to occur.

For these reasons, the small signal model of the whole PV system around a given operating point has been considered. It allows reducing the number of unknown parameters leading to a simpler DKF algorithm. In addition, since the identification is performed by means

of a small perturbation of the system around a steady-state operating point neither PV parameter variation nor converter parameter change occurs during the DKF estimation process. Due to these advantages this approach has been adopted for performing the DKF-based optimization of the MPPT perturbation period. As shown in the following sections, the identification procedure is executed only when needed and the identification time is adapted by using the convergence criterion proposed in Section 3.4.7. It will be demonstrated that this time is short enough with respect to the typical environmental variations, thus the irradiance and the temperature can be considered constant during the identification process.

Small Signal PV Observable Canonical Form (OCF) State Space (SS) Model

The small signal PV Observable Canonical Form (OCF) State-Space (SS) model is now derived. Figure 2.16 shows the small signal model in the neighbourhood of an operating point. As demonstrated in the previous section, the PV system is well described by a regular second-order transfer function (see equation (2.41)), that can be rewritten as:

$$G_{vp,d}(s) = \frac{V_{pv}(s)}{D(s)} = \mu \cdot \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{b_0}{s^2 + a_1 s + a_2} \tag{3.11}$$

where

$$\begin{aligned} b_0 &= \mu\omega_n^2 \\ a_1 &= 2\zeta\omega_n \\ a_2 &= \omega_n^2 \end{aligned} \tag{3.12}$$

From (3.11), it can be obtained that:

$$\begin{aligned} (s^2 + a_1 s + a_2)V_{pv}(s) &= b_0 D(s) \\ s^2 V_{pv}(s) &= b_0 s D(s) - a_1 s V_{pv}(s) - a_2 V_{pv}(s) \\ \frac{d^2}{dt^2}\tilde{v}_{pv}(t) &= b_0\tilde{d}(t) - a_1\frac{d}{dt}\tilde{v}_{pv}(t) - a_2\tilde{v}_{pv}(t) \end{aligned} \tag{3.13}$$

Integrating the last equation:

$$\begin{aligned}\frac{d}{dt}\tilde{v}_{pv}(t) &= b_0\int \tilde{d}(t)dt - a_1\tilde{v}_{pv}(t) - a_2\int \tilde{v}_{pv}(t)dt \\ \frac{d}{dt}\tilde{v}_{pv}(t) &= -a_1\tilde{v}_{pv}(t) + \int (b_0\tilde{d}(t) - a_2\tilde{v}_{pv}(t))dt\end{aligned} \tag{3.14}$$

Choosing $\tilde{v}_{pv}(t)$ as the first state variable and the integral of $(b_0\tilde{d}(t) - a_2\tilde{v}_{pv}(t))$ as the second state, it yields:

$$\begin{aligned}x_2(t) &= \int (b_0\tilde{d}(t) - a_2\tilde{v}_{pv}(t))dt \\ x_1(t) &= \tilde{v}_{pv}(t) \\ \frac{d}{dt}x_2(t) &= b_0\tilde{d}(t) - a_2\tilde{v}_{pv}(t) = b_0\tilde{d}(t) - a_2x_1(t) \\ \frac{d}{dt}x_1(t) &= \frac{d}{dt}\tilde{v}_{pv}(t) = -a_1\tilde{v}_{pv}(t) + x_2(t)\end{aligned} \tag{3.15}$$

Hence, the OCF SS Model is derived:

$$\begin{aligned}\dot{x} &= \begin{bmatrix}-a_1 & 1\\ -a_2 & 0\end{bmatrix}x + \begin{bmatrix}0\\ b_0\end{bmatrix}\tilde{d} \\ \tilde{v}_{pv} &= \begin{bmatrix}1 & 0\end{bmatrix}x\end{aligned} \tag{3.16}$$

Then, by substituting (3.12) in (3.16):

$$\begin{aligned}\dot{x} &= \begin{bmatrix}-2\zeta\omega_n & 1\\ -\omega_n^2 & 0\end{bmatrix}x + \begin{bmatrix}0\\ \mu\omega_n^2\end{bmatrix}\tilde{d} \\ \tilde{v}_{pv} &= \begin{bmatrix}1 & 0\end{bmatrix}x\end{aligned} \tag{3.17}$$

According to the time-domain expression of the power step response of a second-order transfer function, the settling time can be expressed as in (2.9), then, the OCF SS Model becomes:

$$\begin{aligned}\dot{x} &= \begin{bmatrix}2\dfrac{ln(\frac{\varepsilon}{2})}{T_\varepsilon} & 1\\ -\omega_n^2 & 0\end{bmatrix}x + \begin{bmatrix}0\\ \mu\omega_n^2\end{bmatrix}\tilde{d} \\ \tilde{v}_{pv} &= \begin{bmatrix}1 & 0\end{bmatrix}x\end{aligned} \tag{3.18}$$

The achieved model presents one input ($\tilde{d}$) and one output ($\tilde{v}_{pv}$) signals. The parameters to be estimated are the settling time T_ε, the natural frequency ω_n and the DC gain μ.

3.4.3 Model Discretization

The aim of this step is to discretize the previous continuous time model. A good accuracy and a low complexity of the discrete model are necessary in order to achieve good identification results without increasing the execution time. For this purpose, the *Runge-Kutta* Methods (RKMs) [104] are taken into account. They are iterative methods for the approximation of ordinary differential equation solutions, usually used to discretize continuous time models. Excellent accuracy is obtained by using implicit RKMs (*Tustin* or *backward Euler method*). However, complex discrete models are achieved causing a significant increase of both the demanded resources and the execution time. For this reason, the attention has been focused on the explicit methods, which are not complex and then, very attractive for hardware implementations. Here two explicit methods are compared: the *forward Euler method* and the *explicit midpoint method*.

Forward Euler Method

Among the discretization methods, the first order forward Euler, described in equation (3.19), is probably the most common.

$$x(t+T_s) = x(t) + T_s \cdot \frac{dx}{dt} \tag{3.19}$$

where T_s is the sampling period. According to this method, the discrete SS matrices are:

$$\begin{array}{cc} A_d = I + T_s \cdot A & B_d = T_s \cdot B \\ C_d = C & D_d = D \end{array} \tag{3.20}$$

where I is the identity matrix. By applying this method to (3.18), the following discrete SS model is obtained:

$$x(k+1) = \begin{bmatrix} 1+T_s\left(\dfrac{2\cdot ln\left(\frac{\varepsilon}{2}\right)}{T_\varepsilon}\right) & T_s \\ T_s\left(-\omega_n^2\right) & 1 \end{bmatrix} \cdot x(k) + \begin{bmatrix} 0 \\ T_s\left(\mu\cdot\omega_n^2\right) \end{bmatrix} \cdot u(k) \qquad (3.21)$$
$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot x(k)$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$
$$u(k) = \tilde{d}(k)$$
$$y(k) = x_1(k) = \tilde{v}_{pv}(k)$$

$\tilde{d}(k)$ and $\tilde{v}_{pv}(k)$ are the small variations of duty cycle and the panel voltage, respectively. This method generates a simple discretized model to be implemented in the Kalman filter. This way, the final architecture is not expensive in terms of used resources. However, it is of prime importance to verify that this method is able to achieve an acceptable accuracy between the discretized model and the continuous time one. The quality of the model depends mainly on the chosen sampling period T_s. Therefore, several simulations are carried out and the step responses of the Euler model with different T_s are depicted in Figure 3.22. It can be easily deduced from these curves that the best accuracy of the discretized model is obtained for the shortest sampling period.

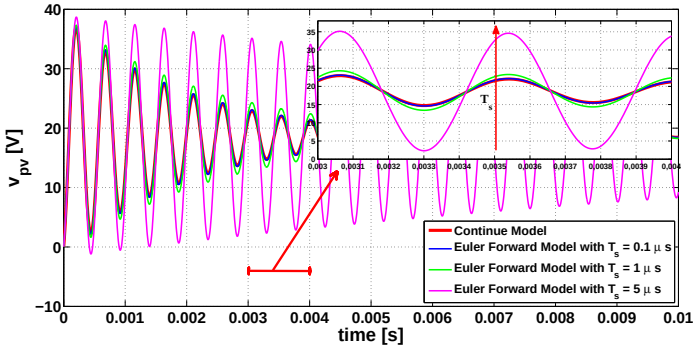


Figure 3.22 Comparison of Euler forward model step response with different sampling periods.

An additional validation is given plotting the poles and zeros of the discrete model with different T_s in the z-plane (see Figure 3.23). It is obvious that by increasing the sampling period the discrete poles move toward the unit circle. Beyond a particular value of T_s the poles are located outside of the unit circle, leading to instability. This particular value depends on both the system parameters and the operating conditions and then, it needs to verify the goodness of the discrete model case by case.

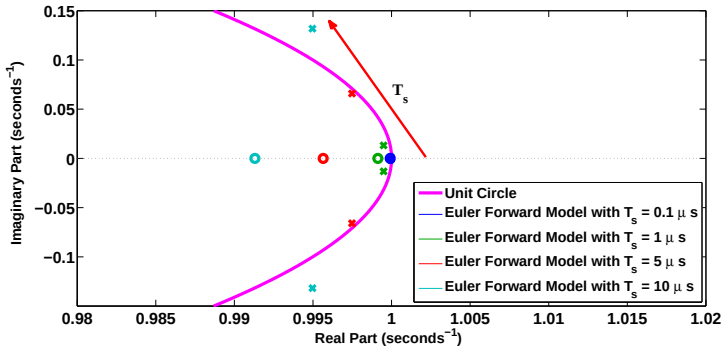


Figure 3.23 Locations of the discrete poles and zeros by using the Euler forward discretization method with different sampling periods.

In the system under analysis, the sampling period is fixed equal to the switching time (5 μs) as described in Chapter 2. Therefore, the forward Euler method involves higher error in the model causing wrong identification results. Moreover, even if the sampling period is reduced for achieving the required accuracy, the time constraint becomes too strict making necessary the use of high performance FPGA platform. Therefore, the low-cost constraint becomes unfulfilled. As a consequence, another discretization method has to be used to achieve a good accuracy at a reasonable hardware cost.

Explicit Midpoint Method

To this purpose, explicit midpoint method is analysed. It gives more precision than the forward Euler method, at only a slight additional hardware cost. The next sample of a variable $x((k+1)T_s)$ is derived from the sum of the current sample $x(kT_s)$ and the product between the sampling period and the slope of the variable at $(k+1/2)T_s$, as shown in Figure 3.24. In this figure, this method is also compared to the Euler one and its better accuracy is shown.

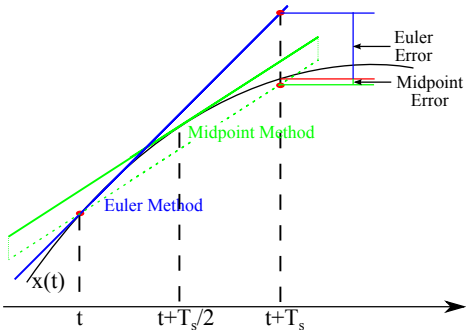


Figure 3.24 Illustration of the explicit midpoint method (green line) and the forward Euler method (blue line).

The explicit midpoint method can be described by the following expression:

$$x(t+T_s) = x(t) + T_s \cdot \left.\frac{dx}{dt}\right|_{t=t+\frac{T_s}{2}} \tag{3.22}$$

The new discrete matrices of the SS model are shown in (3.23).

$$\begin{array}{ll} A_d = I + T_s \cdot A + \dfrac{T_s^2}{2} \cdot A^2 & B_d = T_s \cdot B + \dfrac{T_s^2}{2} \cdot A \cdot B \\ C_d = C & D_d = D \end{array} \tag{3.23}$$

This discretization method gives the following SS model:

$$x(k+1) = \begin{bmatrix} 1+T_s\left(\frac{2\cdot ln\left(\frac{\varepsilon}{2}\right)}{T_\varepsilon}\right)+\frac{T_s^2}{2}\left(\frac{4\cdot ln\left(\frac{\varepsilon}{2}\right)^2}{T_\varepsilon^2}-\omega_n^2\right) & Ts+\frac{T_s^2}{2}\left(\frac{2\cdot ln\left(\frac{\varepsilon}{2}\right)}{T_\varepsilon}\right) \\ T_s\left(-\omega_n^2\right)+\frac{T_s^2}{2}\left(\frac{-2\cdot ln\left(\frac{\varepsilon}{2}\right)\cdot\omega_n^2}{T_\varepsilon}\right) & 1+\frac{T_s^2}{2}\left(-\omega_n^2\right) \end{bmatrix}\cdot x(k)+ \\ +\begin{bmatrix} \frac{T_s^2}{2}\left(\mu\cdot\omega_n^2\right) \\ T_s\left(\mu\cdot\omega_n^2\right) \end{bmatrix}\cdot u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix}\cdot x(k) \tag{3.24}$$

Several simulations are carried out in order to verify the accuracy of this method. Figure 3.25 shows the step response obtained with different settling time. It is clear that a higher accuracy is reached in comparison to the previous discretization method for each value of the sampling period.

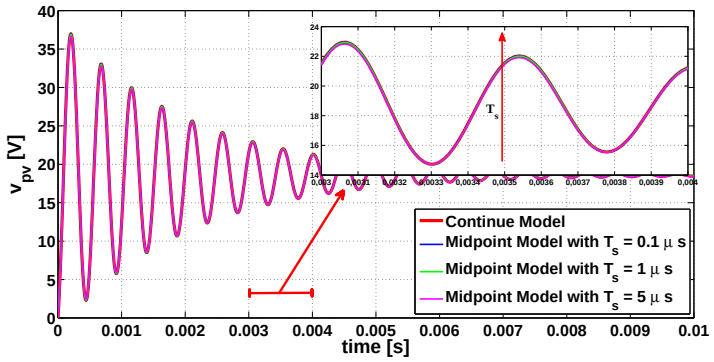


Figure 3.25 Comparison of midpoint model step response with different sampling periods.

The corresponding zeros and poles are displayed in Figure 3.26. It can be seen that in this case the poles are far from the stability region limit.

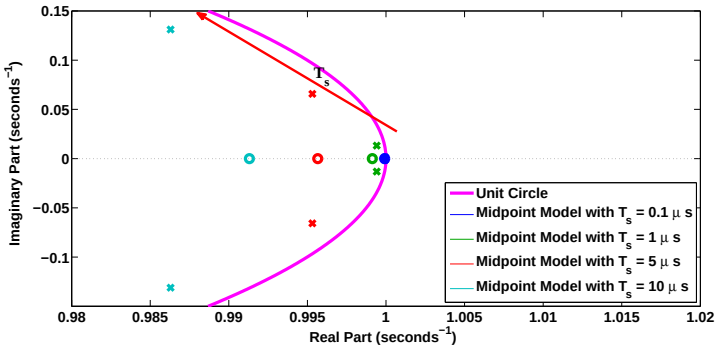


Figure 3.26 Locations of the discrete poles and zeros by using the midpoint discretization method with different sampling periods.

With a sampling period of 5 μs, the midpoint method achieves a very good approximation of the continuous-time model. Figure 3.27 compares the step responses of the two analysed discrete models.

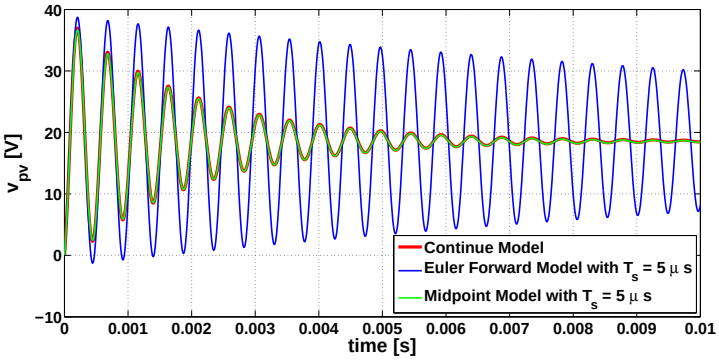


Figure 3.27 Comparison of the step responses obtained by using the midpoint and forward Euler discretization methods at $T_s = 5\ \mu s$.

The locations of the discrete poles and zeros of the two models, with the same sampling period (5 μs), are depicted in Figure 3.28. The poles of the Euler model is located closer to the unit circle, which explains the oscillations of the model in Figure 3.27.

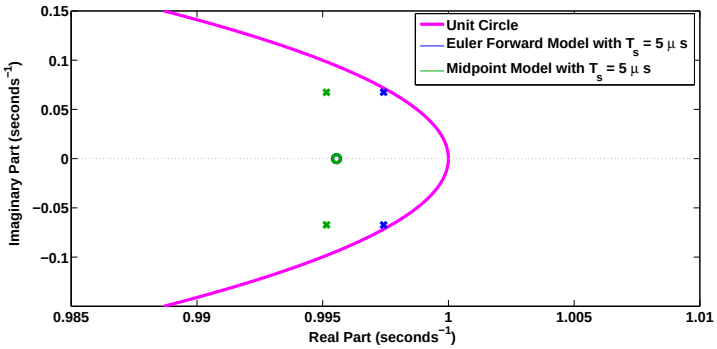


Figure 3.28 Locations of the discrete poles and zeros adopting the midpoint and forward Euler discretization methods at $T_s = 5\ \mu s$.

Another simulation has been performed to compare the PRBS response of the forward Euler model and the midpoint model, shown in Figure 3.29. Also in this case the higher accuracy of the midpoint model is clearly demonstrated.

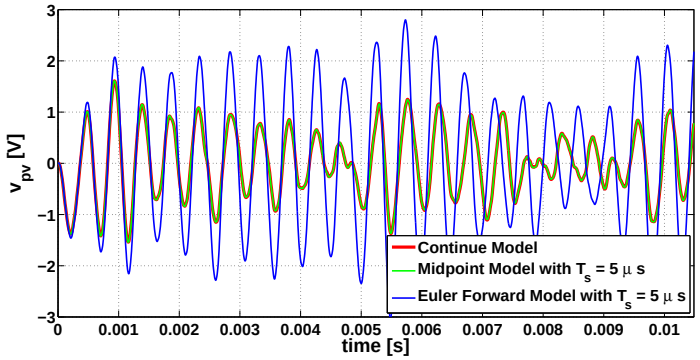


Figure 3.29 Comparison of the midpoint and Euler model output by superimposing the PRBS sequence on the duty-cycle.

The aim of these simulations has been to choose properly the discretization method. A guideline has been given and a compromise, between the complexity of the models and their accuracy has been proposed. Based on this analysis, the midpoint discretization method has been adopted.

3.4.4 Algorithm Digital Realization

Once the DKF based on the previous discrete model is developed, the identification algorithm has to be prepared for the hardware implementation. Unlike the CCM, this identification technique is based on a model of the system, so the identification accuracy is strongly related to the model precision. For this reason, it is very important to achieve a proper quantized algorithm. In order to use the whole dynamic of the words and to simplify the treatment in VHDL code ensuring the same range for each variable, a normalization is carried out. After that, a fixed-point format has to be chosen making a compromise between the algorithm precision and the algorithm complexity.

Normalization of the DKF

The aim is to develop a per-unit algorithm where each variable φ is divided by its base-value φ_B obtaining the corresponding per-unit counterpart φ_n, as shown in (3.25).

$$\varphi_n = \frac{\varphi}{\varphi_B} \tag{3.25}$$

The base-values depend on the nominal value of the variables and the ADC resolution. The defined main base-values are: $v_{pv,B}$ for the panel voltage, e_B for the PRBS sequence, $x_{k,B}$ for the estimated states, $\theta_{k,B}$ for the estimated parameters and the internal variable of the DKF. Base-values have to be chosen carefully in order to avoid either overflow or poor accuracy. They are selected according the maximum possible value that can take each variable, estimating them through several simulations or, when possible, calculating them by the system equations. In this work the following base-values are used:

$$v_{pv,B} = 21.5V \; ; \; e_B = 0.03125$$
$$x_{1,B} = 2.5V \; ; \; x_{2,B} = 35000$$
$$T_{\varepsilon,B} = 20\,ms \; ; \; \omega_{n,B} = 20000\,rad/s \; ; \; \mu_B = 50\,V$$

The equation system (3.26) presents the normalized midpoint model.

$$\begin{aligned} x_N(k+1) = & \begin{bmatrix} 1+\frac{\alpha}{T_{\varepsilon,N}}+\frac{1}{2}\cdot\left(\frac{\alpha^2}{T_{\varepsilon,N}^2}-\beta^2\cdot\omega_{n,N}^2\right) & T_s\left(1+\frac{\alpha}{2\cdot T_{\varepsilon,N}}\right) \\ \frac{\beta^2}{T_s}\cdot\omega_{n,N}^2-\frac{\alpha\cdot\beta^2}{2\cdot T_s\cdot T_{\varepsilon,N}}\cdot\omega_{n,N}^2 & 1-\frac{\beta^2}{2}\cdot\omega_{n,N}^2 \end{bmatrix}\cdot x_N(k)+ \\ & +\begin{bmatrix} \frac{\gamma\cdot T_s}{2}\cdot\mu_N\cdot\omega_{n,N}^2 \\ \gamma\cdot\mu_N\cdot\omega_{n,N}^2 \end{bmatrix}\cdot u_N(k) \end{aligned}$$

$$y_N(k) = \begin{bmatrix} 1 & 0 \end{bmatrix}\cdot x_N(k) \tag{3.26}$$

where

$$\alpha = \frac{2\cdot T_s\cdot ln\left(\frac{\varepsilon}{2}\right)}{T_{\varepsilon,B}}$$
$$\beta = T_s\cdot\omega_{n,B}$$
$$\gamma = T_s\cdot\mu_B\cdot\omega_{n,B}^2$$

Quantization of the DKF

For digital implementations the choice of a suitable data format is of prime importance. Indeed, the identification accuracy, the filter convergence and the identification time are strongly related to this choice. Therefore, the chosen fixed-point data format has to guarantee a proper data precision so as to preserve the algorithm accuracy as close as possible to the continuous-time version. However, a too high word length could cause a too heavy final architecture. Thus, a compromise is mandatory between the algorithm precision and the consumed FPGA resources. Generally, the fixed-point representation can be expressed by using the label $S(wQf)$ for signed data and $U(wQf)$ for unsigned data, where w is the number of bits of the word and f is the number of bits of the fractional part. If the variable is signed, it will be mandatory to assign one bit for the sign. Moreover, after the normalization, it could be necessary represent 1 in the normalized algorithm, so another bit has to be allocated. Then, two bits are used for the integer part:

one for the sign and the other one to be able to represent up to 1. Figure 3.30 shows the estimated settling time in the case of full precision and quantized algorithm with different fixed-point formats. Finally the $S[42Q40]$ fixed-point format has been chosen in the whole DKF algorithm in order to respect the required accuracy of the estimated settling time.

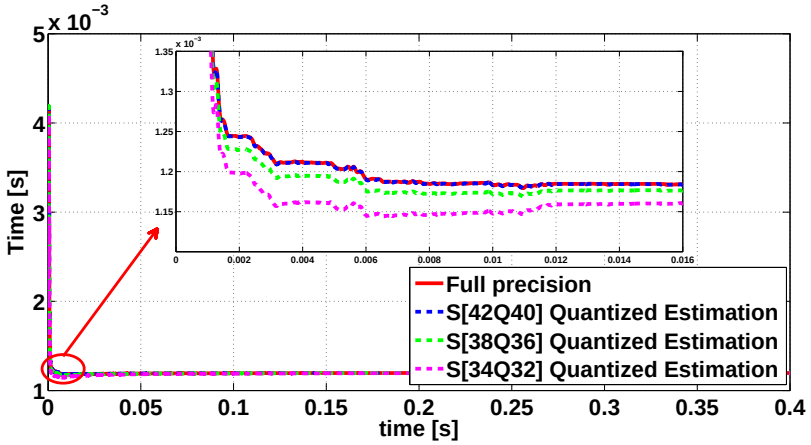


Figure 3.30 Comparison between the Estimated Settling Time with the Normalized DKF and different fixed-point format.

3.4.5 Algorithm Optimization

In the case of FPGA implementation with limited hardware resources, the need to reduce the computational cost of the algorithm makes necessary some optimizations. To this aim, the designer can operate either at the model selection stage or during the digital realization step.

In literature many KF algorithm optimizations have been presented. The objective is always the same: to achieve a reduction of the KF computational cost. For instance, in [75] the infinite Kalman gain K(∞) optimization procedure has been adopted. It consists in assuming that the optimal Kalman gain is constant and then, it can be precalculated off-line avoiding the whole matrix treatment. However, in PV system identification, this assumption does not hold. Indeed, the jacobians matrices are typically not constant, being functions of both states and parameters. Thus, the covariances and gain matrices must be computed on-line [72]. In this work, two optimizations have been used. One concerns the good choice of the parameters to be estimated, whereas, the other one refers to a proper choice of the base-values during the normalization step.

DKF complexity pre-evaluation

First of all, a pre-evaluation of the DKF complexity is made. Table 3.9 shows the needed arithmetic operations of the developed DKF module.

Table 3.9 Number of operation in the presented DKF Algorithm

Kalman Filter Modules	**+**	**-**	$*$	$\div$
STATE PREDICTION	10	0	20	4
STATE INNOVATION	25	5	90	6
PARAMETER INNOVATION	56	29	215	10
TOTAL	**91**	**34**	**325**	**20**

It shows the huge treatment that the DKF needs. Indeed, a large amount of multipliers are required. However, the major issue is given by the number of dividers. Indeed, these operators are the most complex to implement and the most consuming in terms of resources. The reason is that they can only be synthesized in the FPGA fabric con-

versely to the multipliers that can be implemented within the pre-wired DSP units.

DKF Optimization

On one hand, during the model selection stage, an accurate choice of the parameters to be identified has to be made in order to obtain the required information by using a less complex system model. For the application under study, a complexity reduction can be reached estimating instead of the settling time T_ε its inverse f_ε that will be called "*settling frequency*". This allows avoiding divisions in the SS model 3.26. The new SS model in terms of f_ε is obtained:

$$\begin{aligned} x_N(k+1) = &\begin{bmatrix} 1+\delta\cdot f_{\varepsilon,N}+\frac{1}{2}\left(\delta^2\cdot f_{\varepsilon,N}^2-\beta^2\cdot\omega_{n,N}^2\right) & T_s\left(1+\frac{1}{2}\delta f_{\varepsilon,N}\right) \\ \frac{\beta^2}{T_s}\omega_{n,N}^2-\frac{\delta\cdot\beta^2}{2T_s}f_{\varepsilon,N}\cdot\omega_{n,N}^2 & 1-\frac{\beta^2}{2}\omega_{n,N}^2 \end{bmatrix}\cdot x_N(k)+ \\ &\begin{bmatrix} \frac{\delta\cdot T_s}{2}\mu_N\cdot\omega_{n,N}^2 \\ \delta\cdot\mu_N\omega_{n,N}^2 \end{bmatrix}\cdot u_N(k) \\ y_N(k) = &\begin{bmatrix} 1 & 0 \end{bmatrix}\cdot x_N(k) \end{aligned} \tag{3.27}$$

where

$$\begin{aligned} \delta &= 2\cdot T_s\cdot ln\left(\frac{\varepsilon}{2}\right)\cdot f_{\varepsilon,B} \\ \beta &= T_s\cdot\omega_{n,B} \\ \gamma &= T_s\cdot\mu_B\cdot\omega_{n,B}^2 \end{aligned}$$

the index B and N indicates the base-values and the normalized counterpart of the variables, respectively. The base-value of the settling frequency is fixed equal to 5000 Hz.

On the other hand during the algorithm realization step, a good choice of the base-values allows an important reduction of the number of multiplications. In the DKF, it is possible to choose the base-values

of the internal variable as power of two. Thus, when a product due to the normalization has to be carried out, this becomes a simple shift operator.

DKF complexity post-evaluation

After having made these optimizations the DKF complexity is re-evaluated (see Table 3.10). It can be seen that a reduction of 75% of the dividers and 35% of the multipliers is obtained. However, relying on the chosen FPGA device, other efforts have to be made in order to adapt the developed algorithm to the available FPGA resources. This can be made during the development of the FPGA architecture (presented in Chapter 4).

Table 3.10 Number of operation in the presented DKF Algorithm

Kalman Filter Modules	**+**	**-**	$*$	$\div$
STATE PREDICTION	10	0	24	0
STATE INNOVATION	25	5	51	2
PARAMETER INNOVATION	56	29	137	3
TOTAL	**91**	**34**	**212**	**5**

3.4.6 Input Selection

The PV parameter estimation is first carried out without using additional inputs. Unfortunately, the MPPT stepwise perturbation signal does not excite properly the PV system, causing the divergence of the developed Kalman filter. The main cause is the sudden change of the operating point that leads to a variation of the differential resistance. For this reason, in order to persistently excite the PV system, the PRBS sequence is superimposed to the converter duty-cycle (see Section 2.2.1). Its parameters are chosen according to Section 3.3.2.

3.4.7 Convergence Criterion

In order to not perturb continuously the system, the DKF works from time to time. Therefore, it is of prime importance to define a procedure in order to deduce when the estimate is acceptable for stopping the identification process. To this aim, a convergence criterion has been introduced. The Kalman filter reaches the convergence when the estimation error falls within a given boundary error, as shown in (3.28).

$$\begin{aligned} \left|\theta_n - \widehat{\theta}_n\right| &\leq \theta_{n,err} \quad \forall\, n \\ \theta_{n,err} &= \theta_n \cdot err_{\theta_n} \end{aligned} \tag{3.28}$$

where θ_n and $\widehat{\theta}_n$ are the real value and the estimated value of the n-th parameter. $\theta_{n,err}$ defines an error threshold, which depends on the parameter value θ_n and on an imposed relative error err_{θ_n}. The error covariance matrix $P_{\widehat{\theta}}$, given by the filter, provides the variance of the estimates and, then, the corresponding standard deviation σ can be derived. Therefore, the DKF reaches the convergence when σ falls below the imposed error threshold:

$$\sigma_n < \theta_{n,err} \quad \forall\, n \tag{3.29}$$

In (3.29), the $\theta_{n,err}$ depends on the real parameter values, which are unknown. However, if the consistency is verified, it is possible to replace the real value with the identified one. Then, the following criterion is obtained.

$$\begin{aligned} \sigma_n &< \widehat{\theta}_{n,err} \quad \forall\, n \\ \widehat{\theta}_{n,err} &= \widehat{\theta}_n \cdot err_{\theta_n} \end{aligned} \tag{3.30}$$

Hence, the DKF reaches the convergence when the standard deviation is less than the chosen error bound. It is worth to note that the latter depends on the identified parameters and the relative error that is fixed by the designer. This introduces another degree of freedom in comparison with the CCM identification procedure. Indeed, a compromise between the identification time and the identification accuracy can be reached by tuning the error bound.

3.4.8 T_p Estimation

With the optimized DKF the f_ε is estimated. Thus, in order to evaluate an overestimated perturbation period, the equation in (3.31) is adopted.

$$T_p = \frac{1}{\widehat{f_\varepsilon} - \sigma_{\widehat{f_\varepsilon}}} \tag{3.31}$$

where $\sigma_{\widehat{f_\varepsilon}}$ is the settling frequency standard deviation. In order to reduce the computational burden, the $\sigma_{\widehat{f_\varepsilon}}$ has been replaced by an overestimation of it given by the imposed error in the convergence criterion (3.29). This guarantees a safety perturbation period unlike the CCM method proposed previously. The ratio in (3.31) can be carried out by using an internal divisor of the DKF, being at this instant disabled.

3.4.9 Algorithm Validation

In this section the proposed quantized DKF algorithm has to be validated. For this aim, as for the CCM, the model in Figure 2.16 has been simulated in Matlab/Simulink with the PV parameters described in Chapter 2. The differential resistance is imposed equal to 5 Ω, whereas the duty-cycle is fixed to 0.5. First of all, the consistency of the filter is demonstrated. After that, the estimation process is validated with the nominal parameters. Finally, the robustness of the proposed DKF is investigated.

Validation of the DKF consistency

In order to demonstrate the reliability of the estimation, the consistency of the developed DKF has to be verified. As explained in Section 2.2.3, the filter consistency is made of two tests: the unbiased test and the whiteness test. For this aim the PV system and the developed DKF are simulated by using Matlab/Simulink tool.

Concerning the unbiased test, Figure 3.31 shows the normalized innovation sequence q and its moving average $E[q]$, evaluated as in (2.54). The length N of the innovation sequence has been chosen equal to 100. It seems an acceptable value since, as shown below, it represents the degrees of freedom of the χ^2 Test.

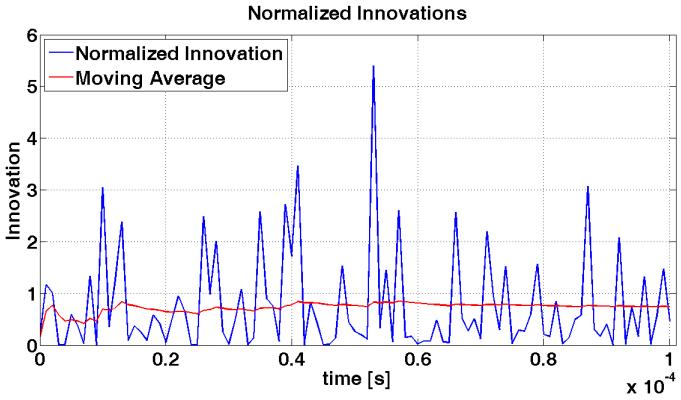


Figure 3.31 Normalized Innovation and moving average

Therefore, as shown in Section 2.2.3, the innovation is unbiased if $NE[q]$ is distributed as a χ^2 random variable in Nm degrees of freedom. Thus, the average of q has to fall within a confidence interval $[b_1, b_2]$, defined as in 3.32, in order to verify the hypothesis that $NE[q]$ is distributed as χ^2_N with a probability $(1-\alpha)$.

$$E[q] \in [b_1, b_2] = \left[\frac{\chi^2_{Nm}\left(\frac{\alpha}{2}\right)}{N}, \frac{\chi^2_{Nm}\left(1-\frac{\alpha}{2}\right)}{N} \right] \tag{3.32}$$

Choosing $(1-\alpha) = 0.95$, it is obtained:

$$E[q] \in \left[\frac{\chi^2_{100}(0.025)}{100}, \frac{\chi^2_{100}(0.975)}{100} \right] = \left[\frac{74}{100}, \frac{130}{100} \right] = [0.74, 1.3] \tag{3.33}$$

The average of the normalized innovation is approximately 0.8 (see Figure 3.31), then it falls within the given confidence interval. Thus, the innovation sequence can be considered unbiased.

On the other hand, concerning the whiteness test, the normalized autocorrelation of the innovation sequence and its 95 % confidence bounds, approximated by $\pm\frac{2}{\sqrt{N}}$, are plotted in Figure 3.32. It is obvious that more than 95% of the samples fall within the confidence bounds. Therefore, the innovation can be considered white.

In conclusion, the unbiasedness and the whiteness of the innovation sequence are demonstrated. Therefore, the estimations achieved

by the DKF are reliable and they can be adopted for adaptive control, as well as for diagnosis and monitoring purposes.

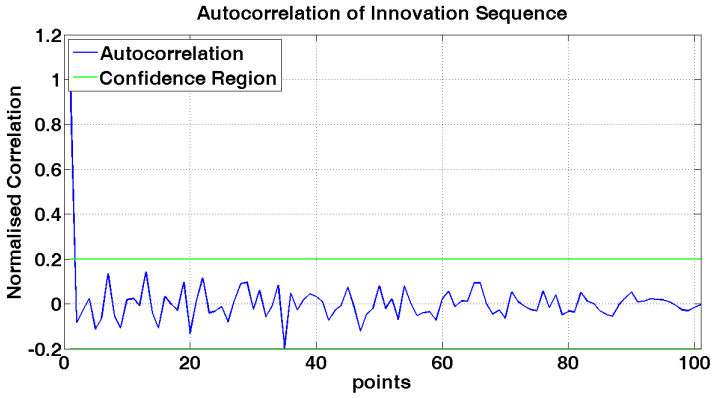


Figure 3.32 Time-Averaged Normalized Autocorrelation of the Innovation Sequence

Validation of the Estimation Process

In this section the estimation process based on the DKF is validated. According to the guideline in Chapter (2), the initialization of the filter is necessary due to the recursive formulation of the Kalman filter. It consists in choosing reasonable initial conditions for $\widehat{x}(0|0)$, $\widehat{\theta}(0|0)$, $P_{\widehat{x}}(0|0)$ and $P_{\widehat{\theta}}(0|0)$. In (3.34) the initial values, adopted in the simulations and experimental tests, are presented.

$$\begin{aligned} \widehat{x}(0|0) &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T \\ P_{\widehat{x}}(0|0) &= \begin{bmatrix} 1 & 1 \\ 0.01 & 10000 \end{bmatrix} \\ \widehat{\theta}(0|0) &= \begin{bmatrix} 500 & 10000 & -10 \end{bmatrix}^T \\ P_{\widehat{\theta}}(0|0) &= \begin{bmatrix} 24 \cdot 10^4 & 0 & 0 \\ 0 & 25 \cdot 10^6 & 0 \\ 0 & 0 & 9 \cdot 10^2 \end{bmatrix} \end{aligned} \tag{3.34}$$

After having made the initialization of the Kalman filter, its tuning has to be performed. It consists in choosing the noise covariance matrices W, R, V and E. They influence the filter consistency, the

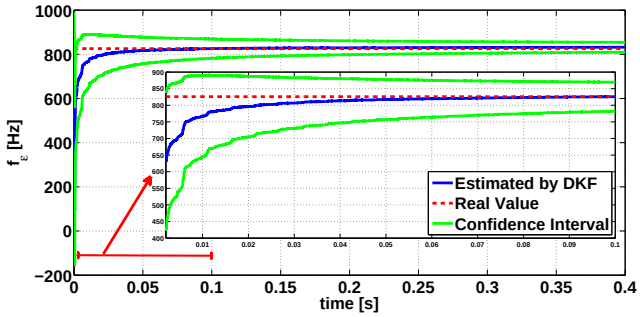
identification time and the filter steady-state error. A trial-and-error approach is used for tuning the filter under analysis and the following noise covariance matrices are used in both the simulations and the experimental tests:

$$W = \begin{bmatrix} 3.145 \cdot 10^{-7} & 3.125 \cdot 10^{-8} \\ 3.125 \cdot 10^{-8} & 3.125 \cdot 10^{-11} \end{bmatrix}$$
$$R = 0.00025$$
$$V = \begin{bmatrix} 6.25 \cdot 10^{-7} & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 3.125 \cdot 10^{-3} \end{bmatrix} \tag{3.35}$$
$$E = 0.8$$

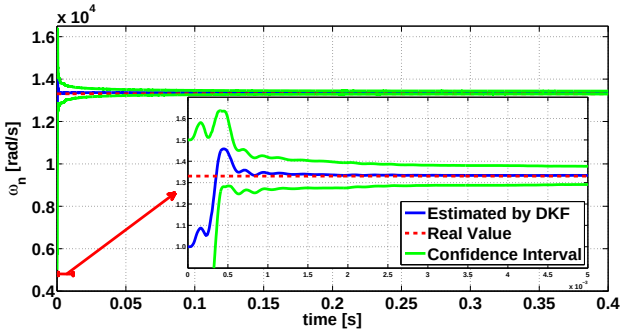
The simulation tests are performed using the PRBS parameters shown in Table 3.2. The choice of the PRBS amplitude has been made according to the discussion in Section (3.3.2) about the non-linearity of the PV system. Hence, Figure 3.33 shows the identified parameters and their confidence intervals, being the estimated parameter $\pm$ the corresponding standard deviation $\sigma_{\hat{\theta}}$. The latter is the square root of the variance given by the error covariance matrix $P_{\hat{\theta}}$. The dotted red lines in the figure are the reference values evaluated from the small-signal model as shown in equation (3.36).

$$\mu = -\frac{V_{dc} \cdot r_d}{R_L + r_d}$$
$$\omega_n = \sqrt{\frac{R_L + r_d}{L \cdot C \cdot (R_C + r_d)}} \tag{3.36}$$
$$f_\varepsilon = \frac{1}{T_\varepsilon} = -\frac{1}{2 \cdot ln\left(\frac{\varepsilon}{2}\right)} \left[\frac{1}{C(R_C + r_d)} + \frac{1}{L}\left(R_L + \frac{R_C \cdot r_d}{R_C + r_d} \right) \right]$$

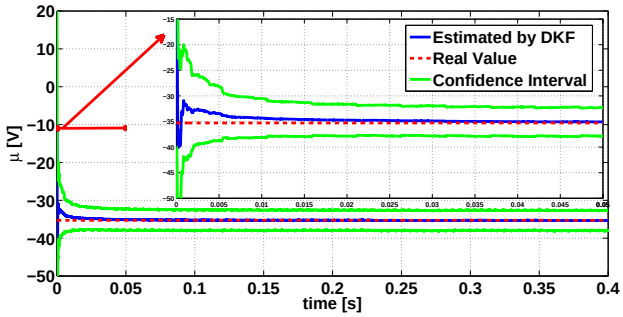
It is clear that the identification results are reliable. Indeed, the errors between the estimated parameters and the real values tend to small values. Moreover, the real value is within the confidence interval of the filter. Therefore, the DKF can be considered validated.



(a)



(b)



(c)

Figure 3.33 Estimated parameters by the DKF by using Matlab/Simulink tool with a differential resistance equal to 5 Ω: a) Estimated settling frequency b) Estimated natural frequency c) Estimated DC gain.

It has been seen that the estimated parameters tends to the corresponding real values. However, in the developed algorithm the esti-

mation is stopped according to the convergence criterion presented in Section 3.4.7. Then, it is interesting to evaluate the estimation error, as well as the identification time when the proposed stopping procedure is used. The relative error on the settling frequency is fixed equal to 17.6%. This choice guarantees an error of 15% on the settling time. It is a reasonable value for achieving a proper compromise between the precision and the identification time, as shown below. Figure 3.34 shows the estimated settling frequency (blue line), the absolute error (green line) and the standard deviation given by the DKF (red line). The DKF is stopped when $\sigma_{\hat{f_\varepsilon}} < f_{\varepsilon,err}$. The corresponding instant (red dotted line) is the identification time T_{id}.

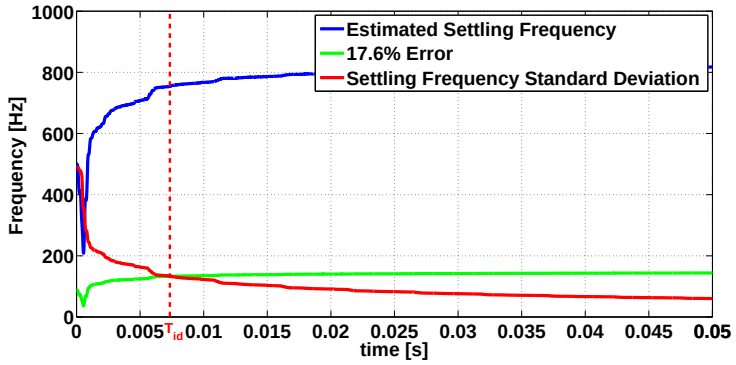


Figure 3.34 Convergence of the DKF.

In this simulation T_{id} is equal to 7.4 *ms* and the estimated parameters with their corresponding real value are shown in Table 3.11. It is worth to note that the achieved estimation error (8.6%) is lower than the error fixed in the convergence criterion (17.6%).

Table 3.11 Estimated Parameters by means the developed DKF

Parameter	Real Value	Estimated Value @ T_{id}	Percentage Error
$f_\varepsilon\ [Hz]$	826	755	8.6 %
$\omega_n\ [k\ rad/s]$	13.30	13.35	0.4 %
$\mu\ [V]$	-35.3	-34.2	3.2 %
$T_\varepsilon = (1/f_\varepsilon)\ [ms]$	1.21	1.32	9.1 %

Validation of the DKF Robustness

In this section the robustness of the developed DKF algorithm is demonstrated by supposing variations in the parameters of the system. Indeed, the differential resistance r_d is strongly related on the irradiance level and the cell temperature. Moreover, the ageing of the power stage mainly affects the input capacitance and the inductance, as well their parasitic resistance. Therefore, as for the CCM algorithm, different cases, shown in Table 3.12, are considered for proving the DKF robustness.

Table 3.12 Simulation cases presented in this thesis for demonstrating the DKF robustness

	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8
C $[\mu F]$	50	50	20	50	100	100	50	50
r_d $[\Omega]$	200	2	50	5	50	5	50	5
L $[\mu H]$	115	115	80	115	115	115	140	140
R_C $[m\Omega]$	10	10	50	50	10	50	10	50
R_L $[m\Omega]$	100	100	200	200	100	200	100	200

Figure 3.35 shows that the estimated parameters (blue line) tend to the corresponding real value (red dotted line). Moreover, the real value is always within the confidence interval (green line) defined by the standard deviation given by the DKF.

Table 3.13 Identification results for the DKF robustness

		case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8
$T_{id}[ms]$		15.6	7.4	3.5	7.4	22	17	17	10
f_ε $[Hz]$	Real	176	1270	688	1020	193	692	198	958
	Est.	176	1210	680	990	194	702	195	942
	Err %	0.2	4.7	1.3	2.8	0.6	1.4	1.6	1.7
ω_n $[krad/s]$	Real	13.2	13.4	25.0	13.4	9.3	9.4	11.9	12.1
	Est.	13.1	13.5	25.1	13.5	9.3	9.5	11.9	12.2
	Err %	0.4	0.6	0.2	0.6	0.01	0.3	0.001	0.01
$\lvert\mu\rvert$ $[V]$	Real	36.0	34.8	35.9	34.6	35.9	34.6	35.9	34.6
	Est.	35.0	34.0	35.8	34.1	36.0	34.8	35.9	34.3
	Err %	2.5	2.3	0.3	1.4	0.01	0.5	0.001	0.01
T_ε $[ms]$	Real	5.7	0.8	1.5	0.9	5.2	1.4	5.1	1.0
	Est.	5.7	0.8	1.5	1.0	5.2	1.4	5.1	1.1
	Err %	0.2	4.9	1.3	2.8	0.6	1.4	1.6	1.7

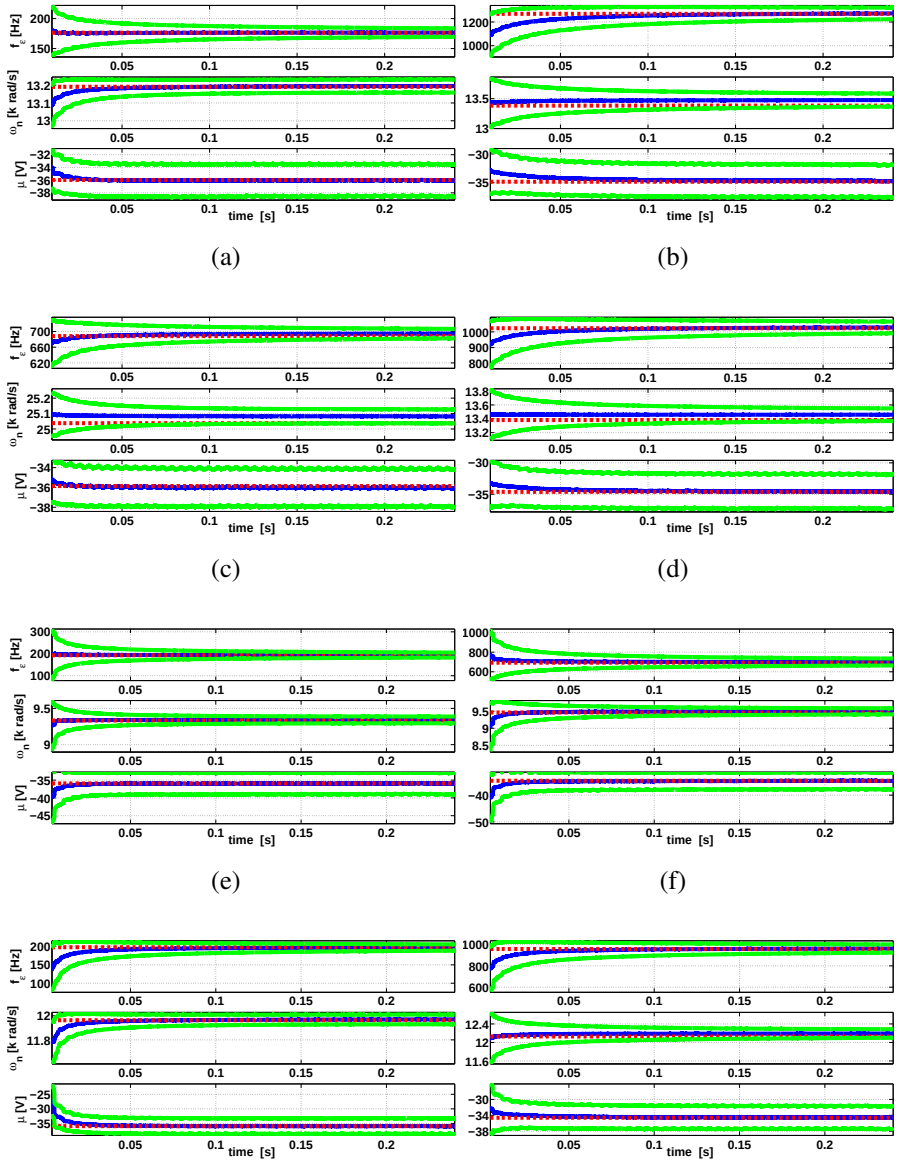


Figure 3.35 Parameter Estimation performed by the DKF with different PV parameters: a) case 1 b) case 2 c) case 3 d) case 4 e) case 5 f) case 6 g) case 7 h) case 8 (Estimated Parameter: blue line; Real Value: red dotted line; Confidence Interval: green line).

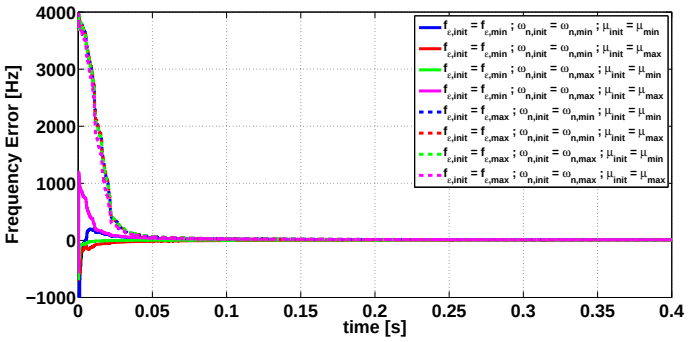
Table 3.13 shows, for every case, the identification time, the estimated parameters at T_{id}, the corresponding real value and the estimation error. It is worth to note that the error remains limited, whereas the identification time changes due to the convergence criterion.

The robustness to the variations of system parameters has been demonstrated. However it is not sufficient for the proposed adaptive MPPT technique. Indeed, as explained previously, the developed DKF works discontinuously and each time it starts the initial conditions are settled equal to the previous estimation. Therefore, another necessary robustness test is carried out by varying the initial values of both states and parameters. For this purpose, the initial values are varied between the 5% and 95% of the corresponding maximum value, as shown in Table 3.14.

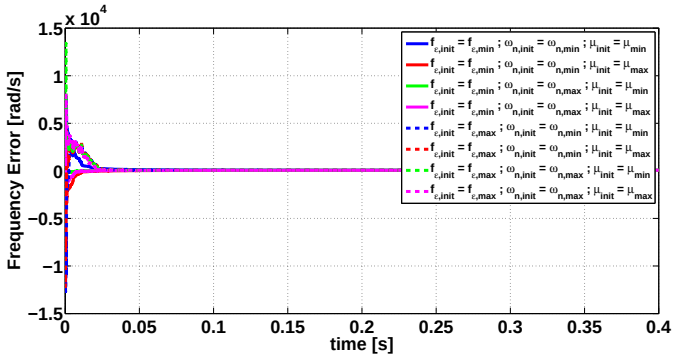
Table 3.14 Variations on the initial conditions

Estimation Parameters	Range of the parameter
f_ε	$[250 \div 4750]$ Hz
ω_n	$[1000 \div 19000]$ Hz
μ	$[-2.5 \div -47.5]$

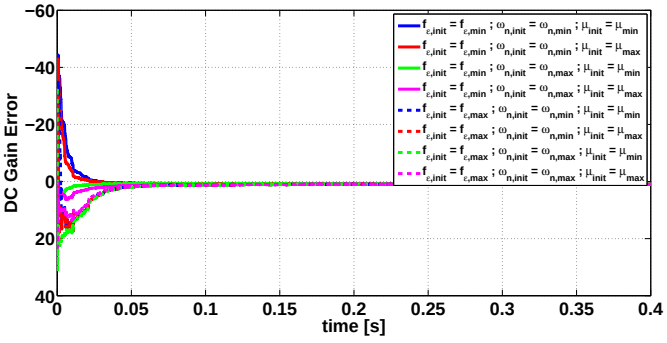
Figure 3.36 proves the goodness of the estimation process. Indeed, the developed DKF ensures a zero steady state error for each parameter. It is obvious that the identification time changes for each identification process and it is adapted thanks to the proposed convergence criterion.



(a)



(b)



(c)

Figure 3.36 Estimation error for different initial conditions: a) Settling frequency estimation error b) Natural frequency estimation error c) DC gain estimation error.

3.5 Conclusions

In this chapter, the development of both the CCM and the DKF algorithm for the identification of PV systems and for the real-time optimization of the MPPT P&O perturbation period has been presented. In the first part of the chapter, the basics of the P&O MPPT controller have been shown and its main blocks (MPPT algorithm and DPWM module) have been described. It is worth to note that, in general, the P&O parameters are not adapted to the actual operating conditions since they are usually chosen, once for all, based on the most critical operating condition. After that, in the second part of the chapter, the two proposed adaptive MPPT controllers have been discussed and the validations of the identification and optimization methods as well as their preparation for the digital implementation have been presented. The chapter has been organized according to the design methodology proposed in Chapter 2. Moreover, a comparison in terms of identification time, complexity and flexibility has been performed. The most important aspects are now re-called:

- The CCM identifies the frequency response in a wide range of frequencies, thus giving a large amount of information that can be useful for different objectives. In the case under study, some system parameters are estimated for achieving an adaptive controller. However, the estimated parameters are punctual values, i.e. without any indication about the reliability of such estimations. On the other hand, the DKF is able to give a confidence interval for each estimate. This guarantees an additional security margin to the adaptive controller, since it avoids underestimation of the settling time.

- The CCM is able to perform a good estimation even when the system presents complex dynamic, i.e. when the second-order dominant approximation does not hold. Whereas, the DKF allows obtaining reliable results only if the system is correctly modelled and the consistence is kept.

- The identification time of the CCM is fixed and it depends on the length of the PRBS. Therefore a compromise has been presented

in order to achieve good identification results and, at the same time, preserving the time performances. On the other hand, in the case of the DKF algorithm the identification time is adapted in real-time thanks to the proposed stopping rule. Moreover, a compromise between the time needed to identify the system and the estimation accuracy can be obtained by tuning the error bound in the convergence criterion.

- The estimation percentage error of the DKF may be kept low thanks to the evaluation of both the parameter vector and its error covariance matrix estimates. Indeed, the convergence criterion used for stopping the DKF algorithm allows defining the maximum allowed percentage error. On the other hand, the CCM estimation error cannot be reduced by performing a longer identification, being the CCM a batch method, but some improvements can be obtained by means of digital filters or smoothing techniques. The errors given by these two methods are summarized in Table 3.8 and in Table 3.13.

Chapter 4

Fully FPGA-based Implementation of Adaptive Digital Controller for PV Application - Architecture Development

4.1 Introduction

This chapter deals with the FPGA architecture development of the proposed adaptive MPPT controllers. First of all, the FPGA-based implementation of the classical MPPT is presented. The architectures of both the P&O algorithm and the DPWM module are illustrated. After that, according to the presented design methodology (see Figure 2.21), the architecture development of the adaptive MPPT controller based on CCM is illustrated, as well the one based on the DKF.

In according to the architecture design presented in Section 2.3.3, the first task consists in making a pre-evaluation of the time/area analysis of the developed identification algorithm preserving the whole parallelism. The aim is to verify if the corresponding FPGA architecture satisfies all the following constraints:

- *Modularity preservation constraint*: it consists in preserving the whole modular partitioning presented during the algorithm development (Section 3.3.1-3.4.1).

- *Area constraint*: the developed FPGA architectures must be implemented in the low-cost Spartan-6 XC6SLX45 FPGA device with the following resources.

Table 4.1 Available Resources in Spartan-6 XC6SLX45 FPGA device

FPGA Hardware Resources	Available Number
Slices	**6822**
FFs	54576
LUTs	27288
RAM Bloks	
16 k bytes Blocks	116
8 k bytes Blocks	232
DSP Blocks	**58**

- *Timing constraint*: it consists in defining an execution time limit for the developed architecture. In this work, the required timing performance depends on the chosen identification algorithm. The required performances for the CCM and for the DKF will be discussed in the corresponding section.

If the fully parallel architecture fulfils these constraints, it is possible to go directly to the architecture design. Otherwise, some architecture optimizations have to be adopted.

After having made the required optimizations the architecture is hand-coded in VHDL and a time/area performances analysis is carried out. Once all constraints are fulfilled, the physically implementation of the developed adaptive MPPT architecture is made.

Finally some conclusions and a time/area comparison between the two proposed identification methods are given.

4.2 FPGA-Based Implementation of the Classical MPPT

In this section the standard MPPT module implementation is presented. It is composed of two main blocks: the DPWM module, to control the DC/DC converter, and the P&O algorithm, to follow the PV source MPP. In this case no particular constraints are noted. The dynamic of the MPPT algorithm is normally slow and its corresponding computational burden is limited. Therefore, no optimization is required and then, the architecture development can be directly made.

4.2.1 Performance Pre-Evaluation

The objective is to estimate the used FPGA resources when the whole parallelism is preserved. Table 4.2 shows the required resources for implementing the classical MPPT controller, presented in Section 3.2.

Table 4.2 Estimation of the FPGA resources keeping the whole parallelism - Classical MPPT Controller (LUT: 6-bit Look-Up-Table; FF: Flip-Flop; DSP: DSP48E Slice).

FPGA Module	Operator	Number	Hardware Resources
DPWM Module	**10-bit Comparator**	**1**	9 LUTs
	Flip Flop	**9**	9 FFs
	9-bit Register	**1**	9 FFs
P&O Algorithm	**12-bit Multiplier**	**1**	1 DSP
	10-bit Comparator	**3**	27 LUTs
	24-bit Comparator	**1**	23 LUTs
	24-bit Register	**2**	48 FFs
	10-bit Register	**2**	20 FFs
	Flip Flop	**7**	7 FFs
	Logic Gate	**3**	3 LUTs
	Up-Down Counter	**1**	51 LUTs 30 FFs
TOTAL			**113 LUTs** **123 FFs** **1 DSP**

As can be easily deduced from Table 4.1, the classical MPPT controller can be easily implemented in the chosen FPGA device.

4.2.2 FPGA Architecture Design

As explained in Section 2.3.3, this step consists in developing the data path and the control unit of each module.

P&O Algorithm

Figure 4.1 shows the developed FPGA architecture of the P&O algorithm presented in Section 3.2.1. This block is activated via the signal *start* each perturbation period T_p. The latter, as well as the step amplitude $\triangle d$, is evaluated taking into account the worst case condition.

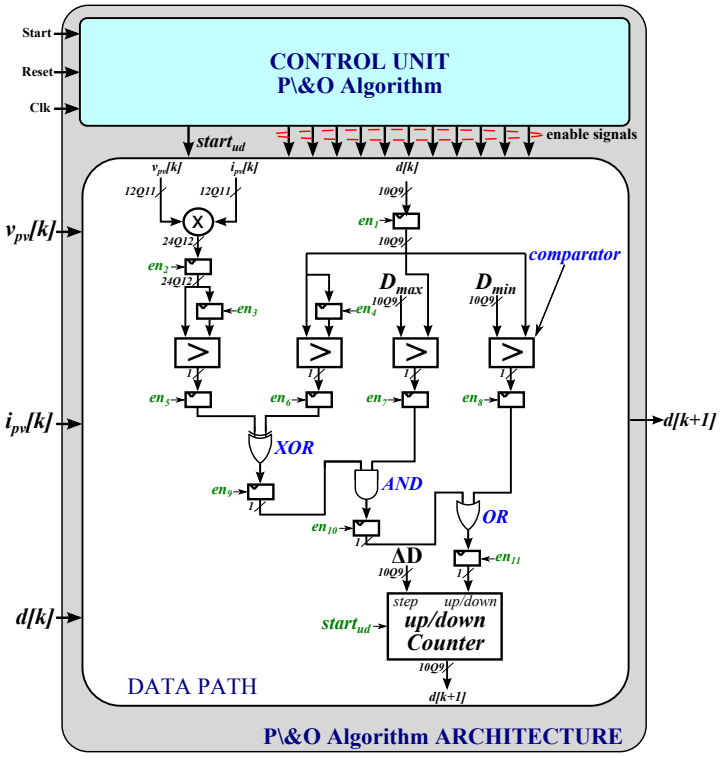


Figure 4.1 FPGA architecture of the P&O algorithm.

DPWM Module

Figure 4.2 shows the FPGA architecture of the developed DPWM module. It is used for driving the power stage. The duty-cycle d, imposed by the P&O algorithm, is compared with the carrier signal generated by a 9-bits counter.

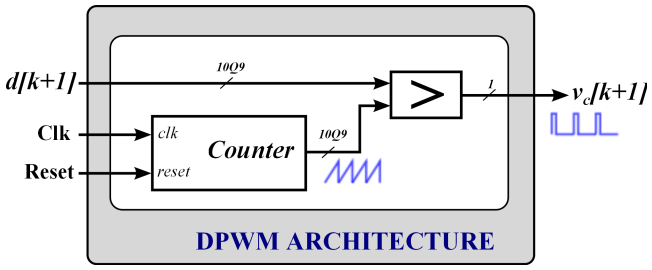


Figure 4.2 FPGA architecture of the DPWM Module.

MPPT Controller

The MPPT Controller uses the DPWM Module and the P&O Module to control the DC/DC Converter in order to follow the time-varying MPP. Figure 4.3 shows the final MPPT Controller Architecture. The *ADC interface* is in charge of reconstructing the digital data, $v_{pv}[k]$ and $i_{pv}[k]$, from the serial ADC signals.

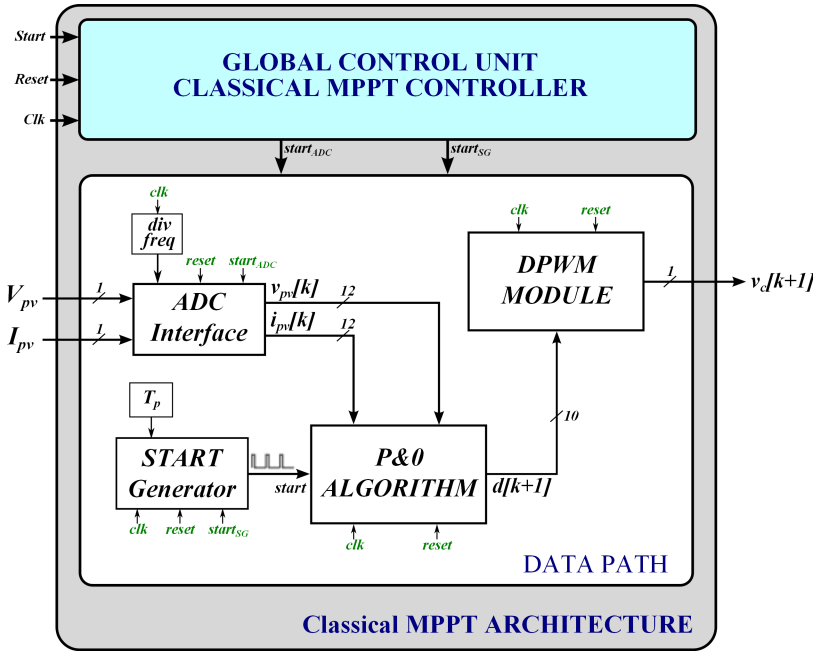


Figure 4.3 FPGA architecture of the classical MPPT controller.

4.2.3 Time/Area Performances Analysis

After having synthesized the developed MPPT controller, the time area analysis is given in this section. Table 4.3 shows the required hardware resources for implementing the whole module.

Table 4.3 Consumed Hardware Resources of the classical MPPT Controller

Resources	Available	Consumed	Percentage
Slice Registers	54576	258	1 %
Slice LUTs	27288	228	1 %
Occuped Slices	6822	172	2 %
16-KByte RAM Blocks	116	0	0 %
DSP48A1s Slices	58	1	1 %

It is clear that the area constraint is satisfied. Concerning the time constraint, no requirements are given. The P&O module gives the new duty-cycle value in only 8 clock cycles.

4.3 FPGA-Based Implementation of the Adaptive MPPT based on CCM

In this section the development of the adaptive MPPT controller based on CCM is made. Firstly, a time-area performance of the whole controller is carried out. It will be demonstrated that in this case no optimizations are required and then, the FPGA architecture can be built. After having hand-coded the architecture by means the VHDL code, a time/area analysis is carried out. The P&O algorithm and the DPWM module have been already presented above.

4.3.1 Performance Pre-Evaluation

A pre-estimation of FPGA resources of the CCM-based adaptive MPPT controller is carried out. Table 4.4 shows that the required resources are lower than the available ones (see Table 4.1).

Therefore, it is possible to build the FPGA architecture without any additional optimizations. Concerning the timing constraint, the identification process has to be as fast as possible in order to achieve a real-time optimization. It will be demonstrated that the developed architecture reduces the time needed for performing the CCM and for evaluating the settling time. Hence, the time needed to optimize the perturbation period is almost equal to the time necessary for injecting the PRBS twice (Section 2.2.2).

4.3.2 FPGA Architecture Design

According to the proposed modular partitioning (see Figure 3.5), the hardware FPGA architecture of each module is designed starting from the lower hierarchical level. The global data path is designed by combining properly the appropriate modules whereas the whole synchronization is guaranteed by the global control unit. In the following only some modules are presented, the other ones being made in a similar way.

Table 4.4 Estimation of FPGA Resources keeping the whole parallelism - adaptive MPPT controller based on CCM (LUT: 6-bit Look-Up-Table; FF: Flip-Flop; DSP: DSP48E Slice).

Modules	Operators	Number	Hardware Resources
CCM Module	20-bit Adder	22	440 LUTs 440 FFs
	20-bit Subtractor	21	420 LUTs 420 FFs
	20-bit complex Mult.	4	16 DSPs 590 LUTs 640 FFs
	1024x14-bit RAM	3	2 BRAM18k 1 BRAM9k
	20-bit ROM	8	3 BRAM18k 1 BRAM9k
	20-bit register	47	940 FFs
	20-bit shift-register	40	2046 FFs
T_p Calculation	20-bit Multiplier	4	4 DSPs 550 LUTs
	512x20-bit RAM	2	1 BRAM18k 1 BRAM9k
	20-bit ROM	1	1 BRAM18k 1 BRAM9k
	20-bit register	12	240 FFs
	9-bit register	2	18 FFs
	Flip Flop	4	4 FFs
	Comparator	4	29 LUTs
	Counter	1	11 LUTs 10 FFs
PRBS Generator	10-bit Register	1	10 FFs
	Flip Flop	10	10 FFs
	XOR	1	1 LUT
	10-bit Multiplexer	1	10 LUTS
DPWM Module	10-bit Comparator	1	9 LUTs
	Flip Flop	9	9 FFs
	9-bit Register	1	9 FFs
MPPT Module	12-bit Multiplier	1	1 DSP
	10-bit Comparator	3	27 LUTs
	24-bit Comparator	1	23 LUTs
	24-bit Register	2	48 FFs
	10-bit Register	2	20 FFs
	Flip Flop	7	7 FFs
	Logic Gate	3	3 LUTs
	Up-Down Counter	1	51 LUTs 30 FFs
TOTAL			**2164 LUTs** **4901 FFs** **21 DSPs** **7 BRAM18k** **4 BRAM9k**

Butterfly Block

Figure 4.4 shows the hardware architecture of the BFI presented in Section 3.3.2. It can be seen that four registers have been added in or-

der to guarantee a pipelined processing. The BFII has been developed according to the same principle.

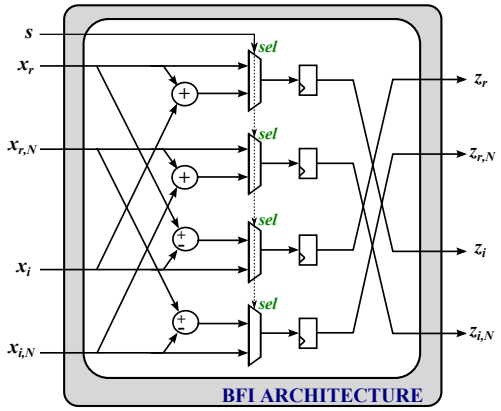


Figure 4.4 FPGA architecture of the BFI Module.

Complex Multiplier

During the FFT evaluation complex multipliers are needed. A complex multiplication is calculated as in (4.1).

$$(a_r + ja_i) \cdot (b_r + jb_i) = (a_r \cdot b_r - a_i \cdot b_i) + j(a_r \cdot b_i + a_i \cdot b_r) \qquad (4.1)$$

It requires 4 multipliers and 2 adder/subtracter. Figure 4.5 shows its hardware implementation.

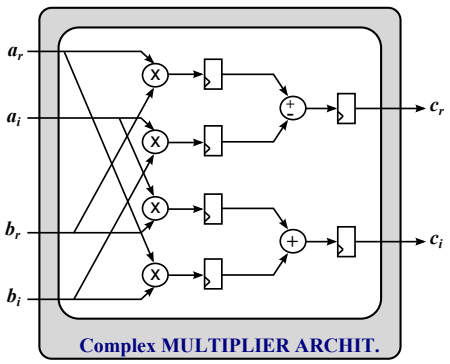


Figure 4.5 FPGA architecture of the complex multiplier.

FWHT/FFT Block

Figure 4.6 presents the FPGA architecture of the FWHT/FFT module, located in the 3^{rd} level of hierarchy. The Transformation block includes the butterfly structures, as well the complex multipliers and the different stages (Figure 3.7). Some memory blocks are adopted for storing the internal vectors (RAM blocks) and the twiddle factors (ROM blocks), which are used for performing the FFT (Section 3.3.2).

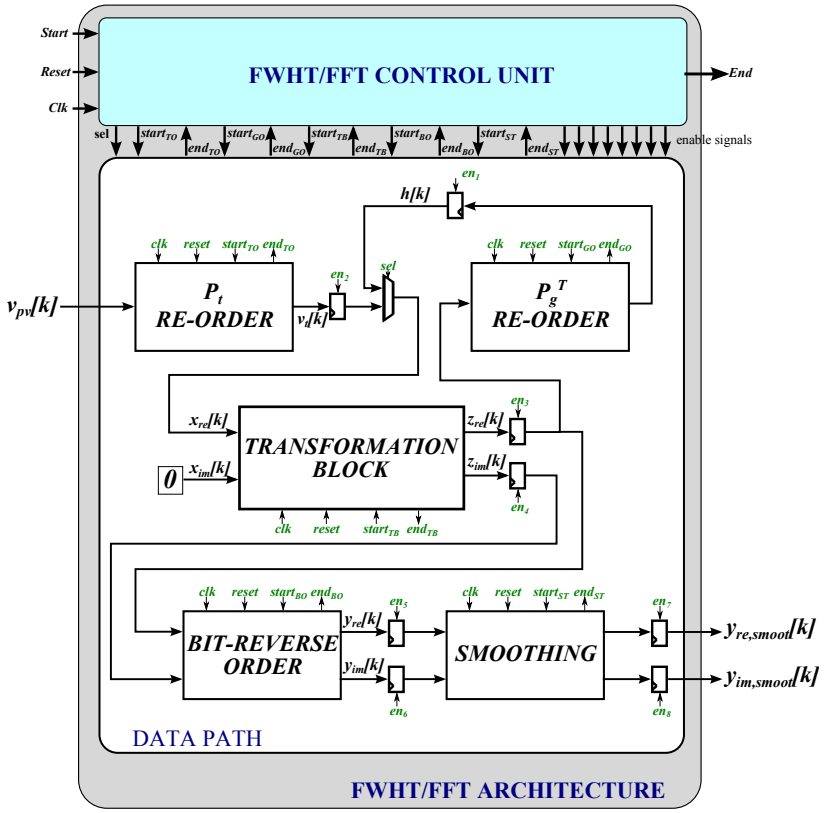


Figure 4.6 FPGA architecture of the FWHT/FFT Module.

Adaptive MPPT controller

Figure 4.7 depicts the whole adaptive MPPT controller. It includes the main modules (P&O module, DPWM module, FWHT/FFT module, T_p Calculation block and PRBS generator), as well the *ADC interface*. The aim of the *Voltage Interface* is to evaluate the small signal panel voltage $\tilde{v}_{pv}$. It removes the DC part from the measurement of the panel voltage v_{pv}. The global control unit has the task of synchronizing all the treatment.

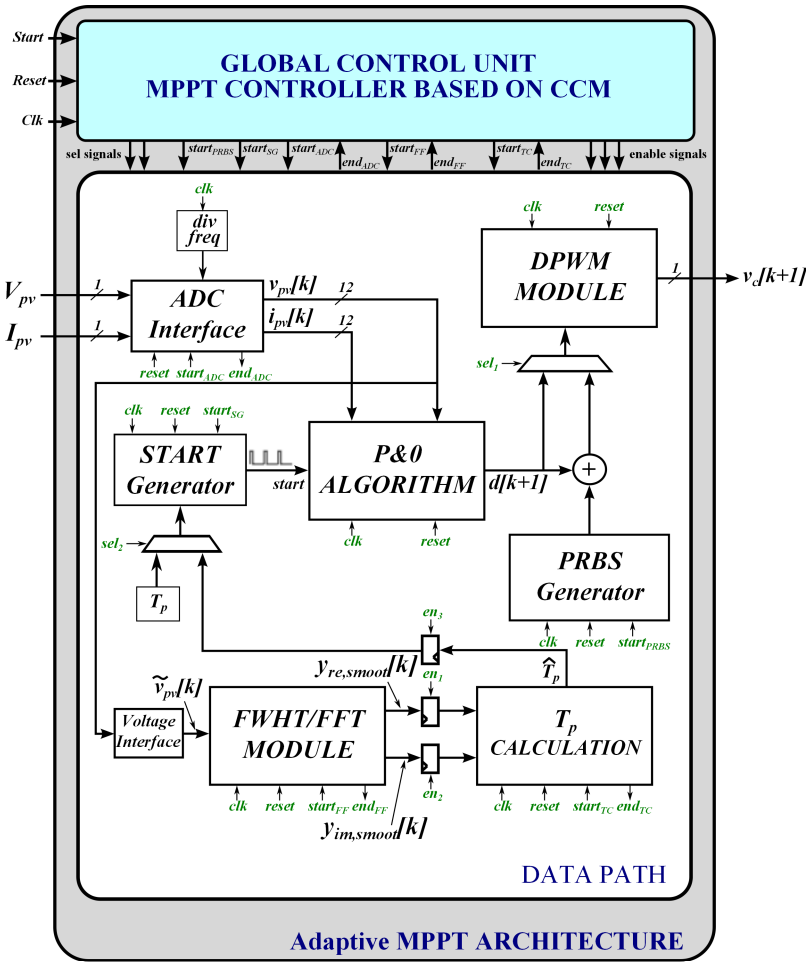


Figure 4.7 FPGA architecture of the adaptive MPPT controller.

4.3.3 VHDL Coding

After having designed the FPGA architecture of each module, the whole controller is ready to be programmed in VHDL using Xilinx ISE design tool. Each module is described by the corresponding VHDL file for preserving the modularity and achieving a hierarchical program.

4.3.4 Time/Area Performances Analysis

At this stage a time/area analysis is made. Table 4.5 shows the required hardware resources for implementing the whole adaptive MPPT controller.

Table 4.5 Consumed Hardware Resources of the adaptive MPPT Controller based on CCM.

Resources	Available	Consumed	Percentage
Slice Registers	54576	4113	7 %
Slice LUTs	27288	7538	27 %
Occuped Slices	6822	2489	36 %
16-KByte RAM Blocks	116	18	15 %
8-KByte RAM Blocks	232	11	4 %
DSP48A1s Slices	58	23	39 %

It can be seen that the developed architecture meets the area constraint and then, can be implemented in the chosen low-cost FPGA device.

Concerning the time constraint, Figure 4.8 shows the timing diagram of the developed MPPT controller.

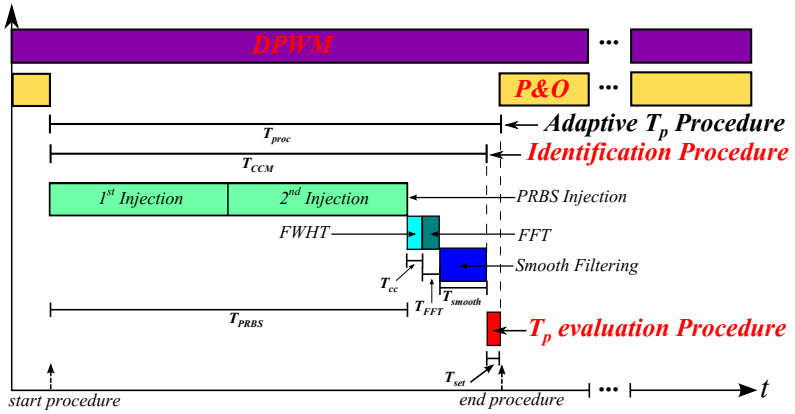


Figure 4.8 Timing Diagram of the adaptive P&O MPPT controller based on CCM.

The P&O algorithm is normally enabled, except when the adaptive procedure is launched. The latter consists in two steps: the identification technique and the T_p evaluation procedure. As expected, the largest part of the whole identification time, which is about 10 *ms*, is due to the injection of the PRBS signal. Indeed, the value T_{PRBS} depends on both the dc/dc converter switching frequency f_{sw} and the PRBS length M. Moreover the PRBS sequence must be injected two times in order to evaluate the impulse response by means of the CCM (see Section 2.2.2). Thus, by injecting one sample per switching period, it results that

$$T_{PRBS} = 2\frac{M}{f_{sw}} \tag{4.2}$$

During T_{PRBS} the typical irradiance variation rates (Section 2.2) give rise to a negligible variation of the PV power, so that the proposed method allows implementing a T_p optimization procedure that really works in real time. The time needed to perform the optimization procedure is the sum of 5 contributions: the time T_{PRBS}, the time T_{cc} needed to evaluate the cross-correlation, the time T_{FFT} needed to evaluate the Fourier transform, the time T_{smooth} necessary for performing the smoothing technique and the time T_{set} needed to calculate the settling time. Their values are summarized in Table 4.6.

Table 4.6 Elaboration time of the FPGA computational block.

Parameter	Absolute Value	% of T_{Tot}
T_{PRBS}=PRBS injection	10.5 ms	**83.51%**
T_{cc}=cross-correlation evaluation	61.7 μs	**0.49%**
T_{FFT}=FFT evaluation	41.3 μs	**0.33%**
T_{smooth}=smoothing technique	1.97 ms	**15.34%**
T_{set}=Settling time evaluation	41.1 μs	**0.33%**
T_{proc} Total time	12.572 ms	100%

The total identification time is equal to 12.5 *ms* and then shorter than the imposed maximum identification time limit (see Section 2.2).

4.4 FPGA-Based Implementation of the Adaptive MPPT based on DKF

This section presents the FPGA implementation of the adaptive MPPT controller based on DKF, illustrated in Chapter 3. Firstly, a pre-evaluation of the time/area performance is made. Its aim is to verify if the architecture, preserving the whole parallelism, fulfils the given constraints. If it is not the case, some architecture optimizations have to be applied. Then, pipelined multipliers are used for increasing the maximum allowable clock frequency. Furthermore, the Algorithm Architecture Adequation (A^3) is adopted in order to reduce the required resources. Finally, the design of the FPGA architecture is carried out and an evaluation of the time/area performances is achieved. Once the constraints are satisfied, the designed FPGA architecture can be physically implemented.

4.4.1 Performance Pre-Evaluation

Concerning the area constraint, Table 4.7 shows the required resources to implement the adaptive MPPT controller preserving the whole parallelism.

In this case, the required number of DSP units is larger than the available resources, shown in Table 4.1. Therefore, an optimization of the architecture is mandatory in order to implement the proposed

Table 4.7 Estimation of FPGA Resources keeping the whole parallelism. LUT: 6-bit Look-Up-Table; FF: Flip-Flop; DSP: DSP48E Slice.

Modules	Operators	Number	Hardware Resources
DKF Module	**42-bit Adder**	**91**	3822 LUTs 3822 FFs
	42-bit Subtractor	**34**	1428 LUTs 1428 FFs
	42-bit Multiplier	**212**	1908 DSPs 22472 LUTs
	42-bit Divider	**5**	80 DSPs 7720 LUTs 8740 FFs
	42-bit Register	**564**	23688 FFs
PRBS Generator	**10-bit Register**	**1**	10 FFs
	Flip Flop	**10**	10 FFs
	XOR	**1**	1 LUT
	10-bit Multiplexer	**1**	10 LUTS
PWM Module	**10-bit Comparator**	**1**	9 LUTs
	Flip Flop	**9**	9 FFs
	9-bit Register	**1**	9 FFs
P&O Algorithm	**12-bit Multiplier**	**1**	1 DSP
	10-bit Comparator	**3**	27 LUTs
	24-bit Comparator	**1**	23 LUTs
	24-bit Register	**2**	48 FFs
	10-bit Register	**2**	20 FFs
	Flip Flop	**7**	7 FFs
	Logic Gate	**3**	3 LUTs
	Up-Down Counter	**1**	51 LUTs 30 FFs
TOTAL			**35566 LUTs** **37821 FFs** **1986 DSPs**

algorithm in the chosen low-cost FPGA device.

With regards to the timing constraint, Figure 4.9 shows the timing diagram of the whole adaptive MPPT controller. The treatment is synchronized as shown in Figure 4.9 and, as described in Section 2.3.3, the sampling period is equal to the switching time (5 μs). If the timing constraint is fulfilled, the treatment can be launched each time a new measurement is available leading to a faster identification process. As a consequence, an execution time lower than the sampling period is desired.

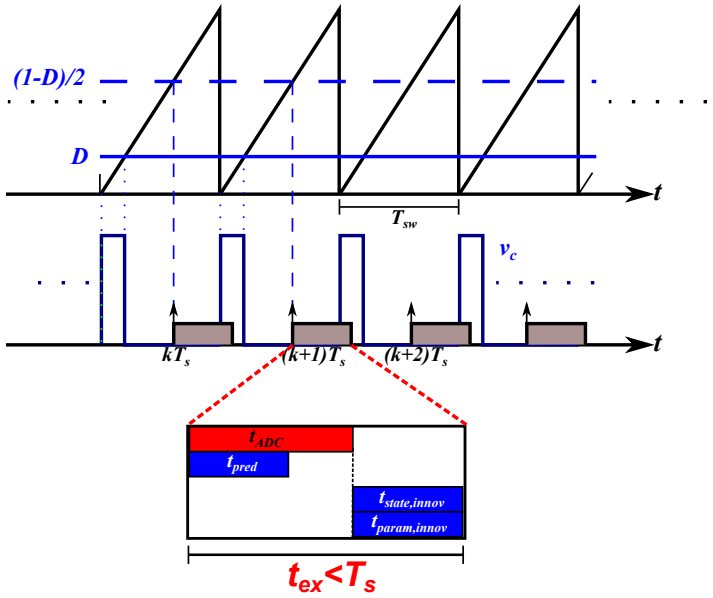


Figure 4.9 Timing Diagram of the Adaptive MPPT Algorithm.

In a pipelined architecture, the execution time can be expressed in terms of latency N (number of clock cycles) and clock period t_{clk} (Throughput).

$$t_{ex} = t_{ADC} + t_{inn} = t_{ADC} + N_{inn}\, t_{clk} \tag{4.3}$$

where N_{inn} represents the maximum latency of the innovation modules and t_{ADC} is the time the ADC needs for the analogue-digital conversion. A first attempt of the developed architecture presents a maximum allowable clock frequency equal to 53 MHz. This poor result comes from the adopted multipliers.

4.4.2 Architecture Optimization

In order to cope with the area mismatching and the limited clock frequency, pipelined multipliers and the A^3 methodology [89] are adopted. In the following, both optimizations are deeply discussed.

Pipelined Multiplier

The Xilinx pipelined multiplier IP with area optimization strategy can be used in order to increase the clock frequency and decrease the number of necessary DSP units. However, the pipelining leads to increase the latency. Therefore, a compromise has to be reached. Several multipliers with different pipeline levels are tested.

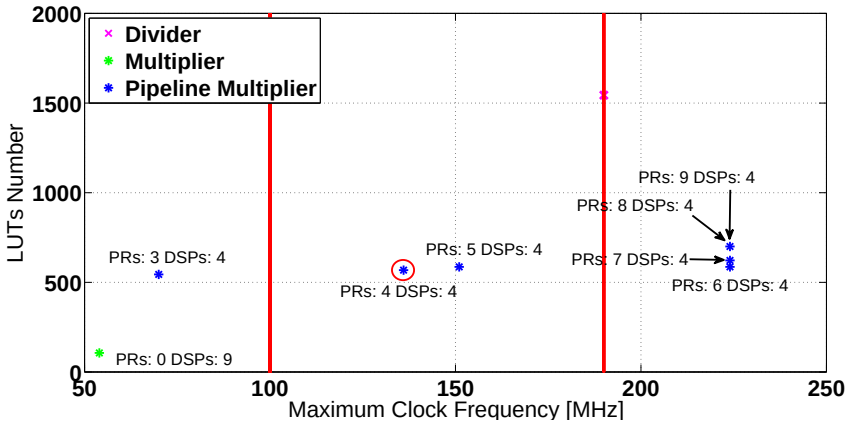


Figure 4.10 Comparison between multipliers with different pipeline level and the divider in terms of maximum clock frequency, used DSP units, number of Look Up Table (LUT) and Pipeline Registers (PRs).

Figure 4.10 shows the comparison between different multipliers with different number of Pipeline Registers (PRs). Their maximum clock frequency, number of LUTs and number of DSP units are compared. The maximum clock frequency equal to 190 MHz is imposed by the divider. Therefore, supposing to work at least with a clock frequency equal to 100 MHz, the multiplier with 4 pipeline registers is chosen. Then, a significant reduction of the required DSP blocks (see Table 4.8) and an increase of the clock frequency is achieved.

However the hardware consumed resources exceed again the available ones. Then, another optimization has been performed.

Table 4.8 Estimation of FPGA resources with and without the pipelined multiplier IP (LUT: 6-bit Look-Up-Table; FF: Flip-Flop; DSP: DSP48E Slice).

Operator	Before		After	
	Number	Resources	Number	Resources
42-bit Adder	**91**	3822 LUTs 3822 FFs	**91**	3822 LUTs 3822 FFs
42-bit Subtractor	**34**	1428 LUTs 1428 FFs	**34**	1428 LUTs 1428 FFs
42-bit Multiplier	**212**	1908 DSPs 22472 LUTs	**212**	848 DSPs 120416 LUTs
42-bit Divider	**5**	80 DSPs 7720 LUTs 8740 FFs	**5**	80 DSPs 7720 LUTs 8740 FFs
42-bit Register	**564**	23688 FFs	**564**	23688 FFs
TOTAL	**35566 LUTs 37678 FFs 1986 DSPs**		**133386 LUTs 37678 FFs 928 DSPs**	

Algorithm Architecture Adequation (A^3)

The aim of this step is to deal with the area mismatching between the available FPGA resources and the required ones. It consists in serializing the treatment in order to share the resources at the cost of an increase of the latency. Thus, it is generally applied to the heaviest operators like dividers and multipliers. According to Section 2.3.3, the A^3 approach consists in three steps. In the following, it is illustrated via the state prediction module.

1. Data Flow Graph

The state prediction module evaluates the prediction of the state vector as described in Table 2.2. Expressions (4.4)-(4.5) represent the implemented discrete-time normalized equations. They use the parameters prediction, the previous state innovation and the perturbation PRBS signal for calculating the predicted state vector.

$$\begin{aligned}\widehat{x}_{1,N}(k|k-1) = &\left[1 + C_1 \cdot \widehat{f}_{\varepsilon,N}(k) + C_2 \cdot \widehat{f}^{\,2}_{\varepsilon,N}(k) + C_3 \cdot \widehat{\omega}^{2}_{n,N}(k)\right] \widehat{x}_{1,N}(k-1) + \\ &+ \left[C_4 \cdot \widehat{f}_{\varepsilon,N}(k) + C_5\right] \widehat{x}_{2,N}(k-1) + \left[C_6 \cdot \widehat{\mu}_N(k) \cdot \widehat{\omega}^{2}_{n,N}(k)\right] u_N(k-1)\end{aligned} \tag{4.4}$$

$$\begin{aligned}\widehat{x}_{2,N}(k|k-1) = &\left[C_7 \cdot \widehat{\omega}_{n,N}^2(k) + C_8 \cdot \widehat{f}_{\varepsilon,N}(k) \cdot \widehat{\omega}_{n,N}^2(k)\right] \widehat{x}_{1,N}(k-1) + \\ &+ \left[1 + C_9 \cdot \widehat{\omega}_{n,N}^2(k)\right] \widehat{x}_{2,N}(k-1) + \left[C_{10} \cdot \widehat{\mu}_N(k) \cdot \widehat{x}_{1,N}(k-1)\right] u_N(k-1)\end{aligned} \tag{4.5}$$

where

$$C_1 = 2log\left(\frac{\varepsilon}{2}\right) T_s f_{\varepsilon,B} \qquad C_2 = 2\left[log\left(\frac{\varepsilon}{2}\right) T_s f_{\varepsilon,B}\right]^2 \qquad C_3 = -\frac{1}{2} T_s \omega_{n,B}$$

$$C_4 = \frac{log\left(\frac{\varepsilon}{2}\right) x_{2,B} T_s^2 f_{\varepsilon,B}}{x_{1,B}} \qquad C_5 = \frac{T_s x_{2,B}}{x_{1,B}} \qquad C_6 = \frac{T_s^2 \mu_B \omega_{n,B}^2 u_b}{2 x_{1,B}}$$

$$C_7 = -\frac{T_s x_{1,B} \omega n, B^2}{x_{2,B}} \qquad C_8 = \frac{log\left(\frac{\varepsilon}{2}\right) x_{1,B} T_s^2 f_{\varepsilon,B} \omega n, B^2}{x_{2,B}} \qquad C_9 = -\frac{1}{2} T_s^2 \omega n, B^2$$

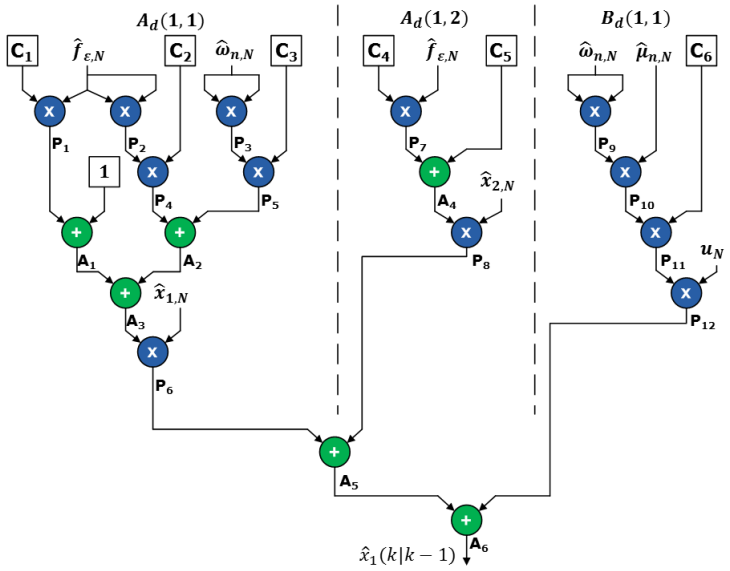
$$C_{10} = \frac{T_s \mu_B \omega n, B^2 u_B}{x_{2,B}}$$

The DFGs depicted in Figure 4.11(a) and Figure 4.11(b) are obtained replacing each operation by the corresponding operator.

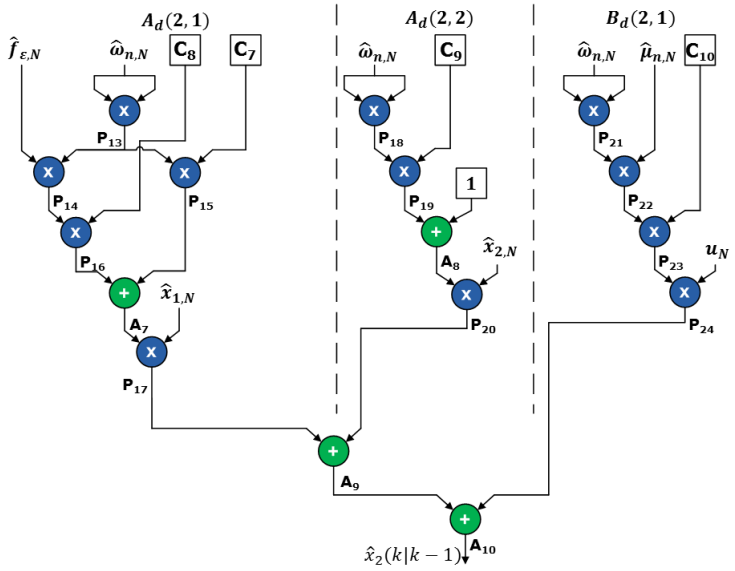
2. Data Dependency

At this stage the best factorization level is determined and the data dependency is evaluated.

The factorization level of the whole DKF architecture is now investigated. In a XC6SLX45 Xilinx Spartan-6 FPGA 58 DSPs slices are available. Of the 5 dividers necessary to this design, 3 are used in the state Kalman gain calculation and 2 in the parameter Kalman gain calculation. Using only one divider for the both Kalman filters will be too confusing. So, in order to preserve the architecture modularity, two dividers have to be used in the architecture, one for each KF. Knowing that each divider is consuming 16 DSP units, only 26 DSP units remain to implement all the requested multipliers. As each multiplier is requiring 4 DSP units, at most 6 multipliers can be implemented. They are located in the following way: one in the state prediction module, two in the state innovation module and three in the parameter innovation module, accordingly to the modularity constraint.



(a)



(b)

Figure 4.11 Data Flow Graph of State Prediction Module: a) First State ; b) Second State.

Since each multiplication must be executed only when its inputs are available, the synchronization of the data treatment is of primary importance. Thus, the data dependency must be evaluated for each module keeping in mind that each multiplier takes 4 clock cycles. First of all, the DFG is divided in levels. A level is a group of independent multiplications depending only on the previous level multiplication results. For instance, the state prediction DFG subdivided in levels is shown in Figure 4.12 where the name and the order of multiplications is preserved.

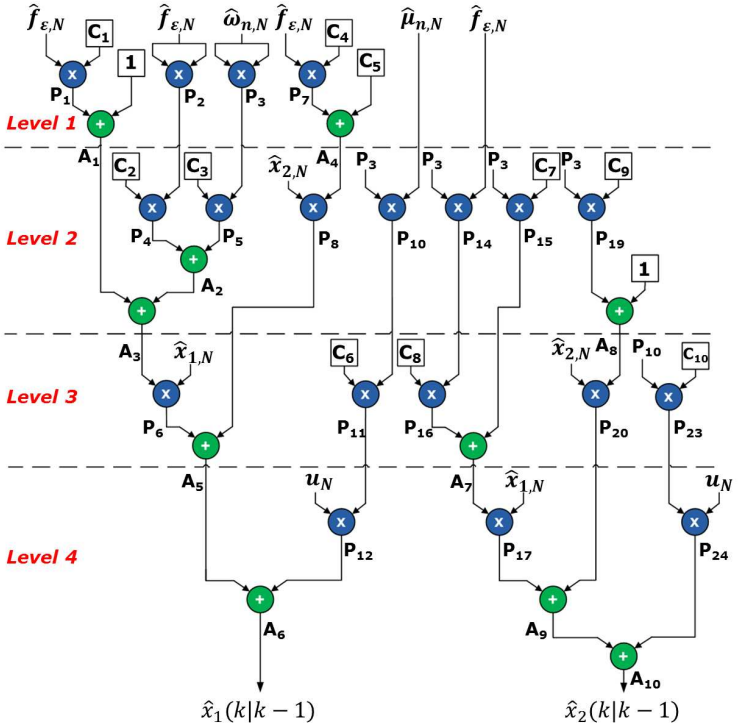


Figure 4.12 Data Flow Graph of State Prediction Module subdivided into Levels.

It is made of 4 levels so that, performing the multiplications from the first level up to the last level, 36 clock cycles are necessary to evaluate the predicted state vector. However, the latency can be reduced by changing the order of the multiplications. The change can be made within a level (inside-level ordering) or from a level to another one (outside-level ordering). For instance, in the state prediction DFG,

the multiplication P_3 can be performed before the multiplication P_1 (inside-level ordering) or the product P_8 can be moved from level 2 to level 3 (outside-level ordering). The purpose is to reduce the useless clock cycles. The re-ordering of the multiplications have leaded to the DFG depicted in Figure 4.13 where the latency is reduced to 29.

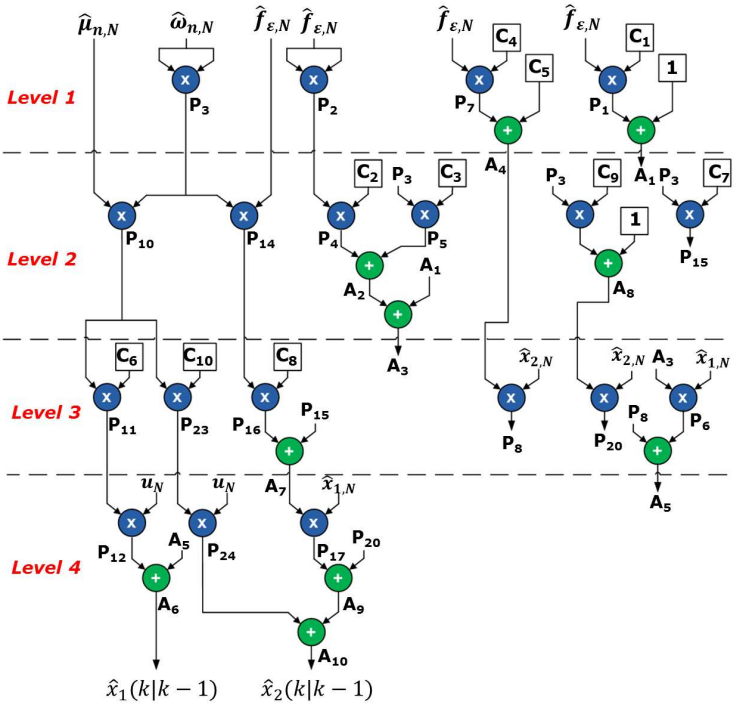


Figure 4.13 Data Flow Graph of State Prediction Module after multiplications ordering.

3. Factorized Data Flow Graph

After the data dependency study, the factorization of the DFG is made resulting in a Factorized DFG. According to the proposed factorization level, only one multiplier is used for evaluating the predicted state vector. Thus, a significant reduction of the hardware consumed resources is achieved at the cost of an increase of the latency. In order to delimit the factorized border some nodes have to be introduced: F("Fork"), J("Join") and I("Iterate") [80]. Figure 4.14 shows the final FDFG and Table 4.9 highlights the achieved hardware resource reduction.

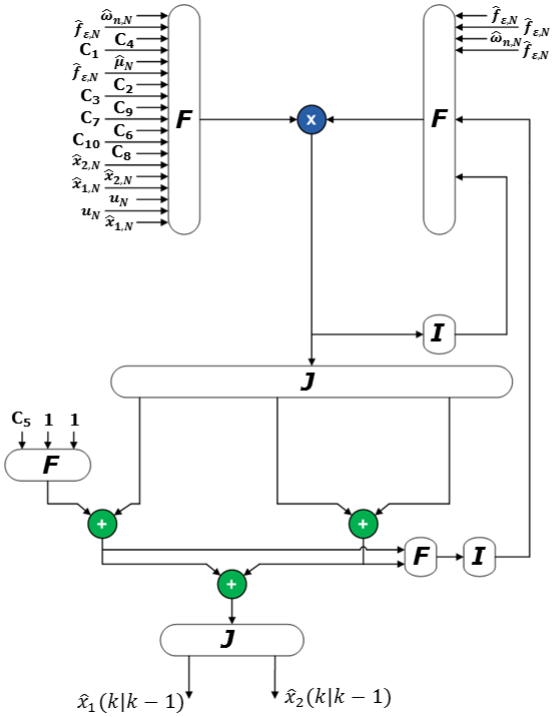


Figure 4.14 Factorized Data Flow Graph of the State Prediction Module.

Table 4.9 Estimation of FPGA Resources before and after the factorization (LUT: 6-bit Look-Up-Table; FF: Flip-Flop; DSP: DSP48E Slice).

Operator	Before		After	
	Number	Resources	Number	Resources
42-bit Adder	**91**	3822 LUTs 3822 FFs	**91**	3822 LUTs 3822 FFs
42-bit Subtractor	**34**	1428 LUTs 1428 FFs	**34**	1428 LUTs 1428 FFs
42-bit Multiplier	**212**	848 DSPs 120416 LUTs	**6**	24 DSPs 3408 LUTs
42-bit Divider	**5**	80 DSPs 7720 LUTs 8740 FFs	**2**	32 DSPs 3088 LUTs 3496 FFs
42-bit Register	**564**	23688 FFs	**564**	23688 FFs
TOTAL	**133386 LUTs 37678 FFs 928 DSPs**		**11746 LUTs 37678 FFs 56 DSPs**	

4.4.3 FPGA Architecture Design

The design of the hardware FPGA architecture of each module is made by respecting the given modular partitioning and the corresponding hierarchical level. According to the FDFG, obtained previously for each module, the hardware architecture is built by replacing each nodes (F,J,I) by their corresponding operators. Therefore, the node F is replaced by a multiplexer, whereas, the nodes J by a register and I by an accumulator. The global data path (3^{rd} and 4^{th} Level of the hierarchy) has to be designed by associating the required modules. A global control unit is associated to the global data path for synchronizing the whole treatment. In the following the state prediction module, the DFK module and the whole adaptive MPPT controller are presented.

State Prediction Architecture

Figure 4.15 shows the hardware FPGA architecture of the state prediction module. When the signal *start* is high, the control unit sends the synchronization signal to the data-path. The order of the register enables and of the multiplexer selection signals is decided during the data dependency study. Then, the inputs of the module are the state variable estimation and the perturbation signal on the previous instant $(k-1)$, as well as the predicted parameters. When the results are available the *end* signal becomes high, and the control unit returns in the $STATE\ 0$ and waits the next *start* signal. The computation time of the module architecture depends on the latency and on the clock frequency. The latency can be derived from the finite state machine.

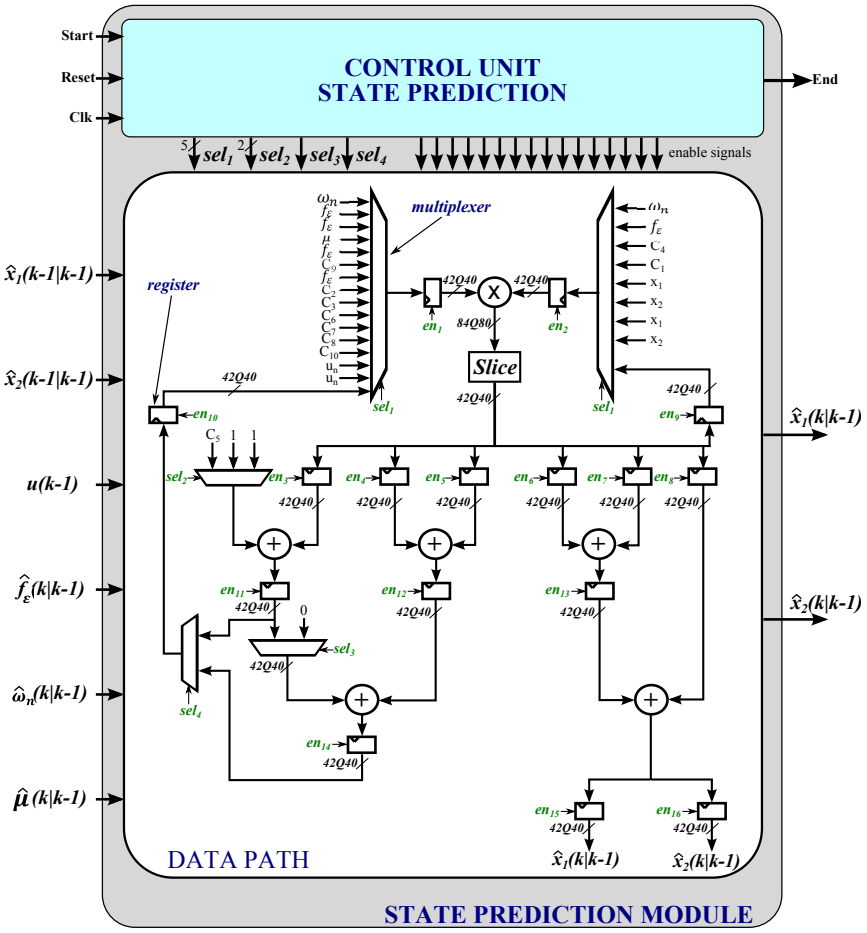


Figure 4.15 Hardware FPGA Architecture - State Prediction Module.

Dual Kalman Architecture

Figure 4.16 shows the data-path of the DKF Module. Its inputs are the small signal panel voltage $\tilde{v}_{pv}$ and the PRBS input signal. The state and the parameter estimates, as well their corresponding error variance matrices, are the outputs of the module.

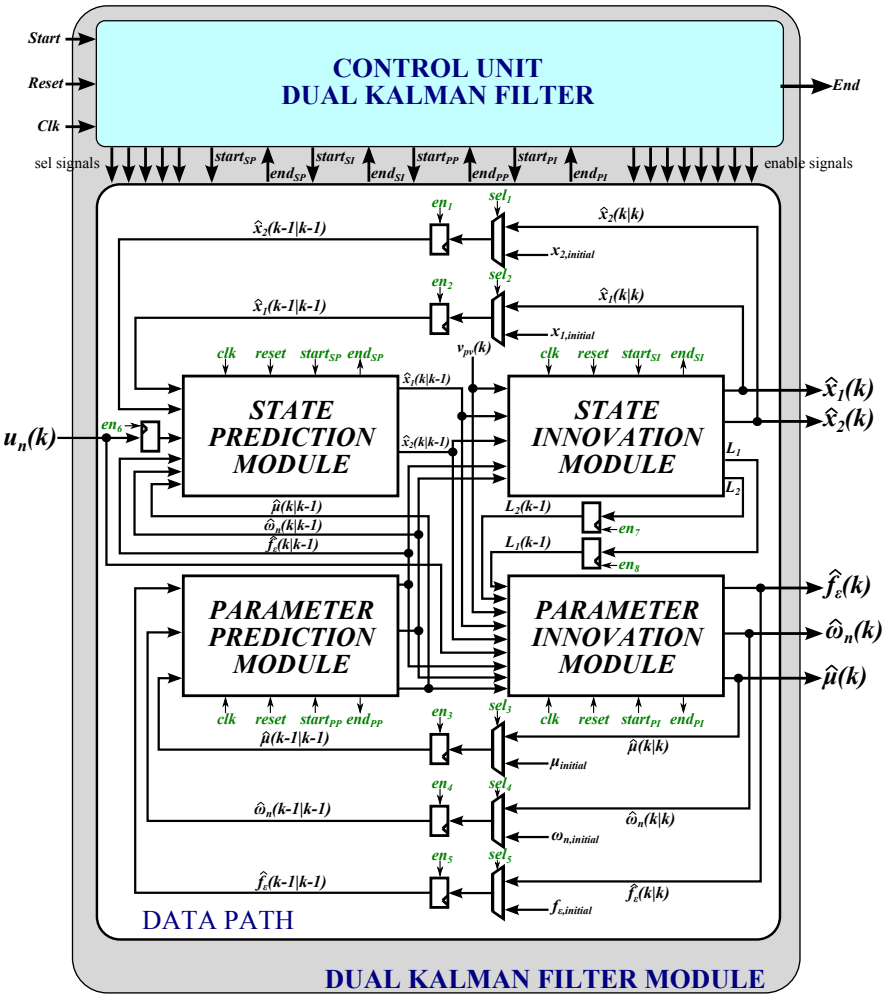


Figure 4.16 Hardware FPGA Architecture - Dual Kalman Filter Module.

Convergence Criterion

As described in Section 3.4.7, a convergence criterion has been developed for stopping the DKF module. This procedure consists in comparing the settling frequency standard deviation $\sigma_{\widehat{f_\varepsilon}}$ with the chosen error bound $f_{\varepsilon,err}$ (Section 3.4.7). The main implementation issue is due to the fact that the DKF evaluates the variance of the estimated parameters ($P_{\widehat{\theta}}$), so that a square root would be necessary to calculate their standard deviation. However, the stopping rule only requires the comparison between the standard deviation and the error of the settling frequeny. Then, instead of performing the square root of the variance, the square of the error is carried out. Hence, a multiplier and a comparator are sufficient for implementing the convergence criterion architecture.

T_p Calculation

Up to now the settling frequency is estimated thanks to the developed DKF. Thus, in order to evaluate an overestimated settling time, the KF variance is used. If the filter is consistent, the real value of the estimate falls within the standard deviation band. Therefore, estimating the settling frequency, the worst case for the settling time is the lower limit of the estimation band. This consideration leads to the relation (4.6).

$$T_{p,min} = \frac{1}{\widehat{f_\varepsilon} - \sqrt{\sigma^2_{\widehat{f_\varepsilon}}}} \tag{4.6}$$

The square root of the variance $\sigma^2_{f_\varepsilon}$ is again necessary. However, it is replaced by the settling frequency error $f_{\varepsilon,err}$ calculated in the stop procedure architecture. Indeed, the KF is stopped when the variance falls below the square of the settling frequency error, then it is assured that this last value is bigger than the variance but very close to it. Then, it is sure that the calculated minimum perturbation period, in (4.7), is overestimated.

$$T_{p,min} = \frac{1}{\widehat{f_\varepsilon} - f_{\varepsilon,err}} \tag{4.7}$$

This guarantees an additional security margin to the calculation of the MPPT period in comparison to the adaptive controller based on CCM. Besides, it is worth mentioning that the computation of the ratio in (4.7) does not require an additional divider since this computation, taking place after the DKF estimation, can use one of the dividers of this IP.

Adaptive MPPT Architecture

After having built all the required modules, the hardware architecture of the whole adaptive MPPT controller can be developed, as shown in Figure 4.17.

According to the algorithm optimization (Section 3.4.5), the inputs of the DKF must be normalized. For this reason, the *input interface* is introduced for normalizing the perturbation signal whereas the *voltage interface* is adapted for evaluating and normalizing the small signal panel voltage. With regards to the PRBS normalization, it has been simply assigned 1 when the perturbation is positive and -1 otherwise. Concerning the panel voltage, the normalization requires a multiplication. Indeed, the relation (4.8) holds.

$$\tilde{v}_{pv,N} = \tilde{v}_{pv,ADC}\frac{r_{ADC}}{v_{pv,B}} = \tilde{v}_{pv,ADC}\frac{0.0404}{2.5} = \tilde{v}_{pv,ADC}\cdot 0.01616 \qquad (4.8)$$

where $\tilde{v}_{pv,N}$ and $\tilde{v}_{pv,ADC}$ are the normalized small-signal panel voltage and the value given by the ADC block, respectively. r_{ADC} and $v_{pv,B}$ depend on the application and they are the ADC resolution and the based value given to the panel voltage, respectively. Another multiplier is then required for normalizing the panel voltage. However, it is possible to tune the base-value of the small signal panel voltage for avoiding the multiplication. Indeed, imposing a base-value equal to $2.5856V$, the relation (4.9) is obtained.

$$\tilde{v}_{pv,N} = \tilde{v}_{pv,ADC}\frac{0.0404}{2.5856} = \tilde{v}_{pv,ADC}\cdot 2^{-6} \qquad (4.9)$$

With this choice, the multiplier can be replaced by a simple shift operator, saving hardware resources.

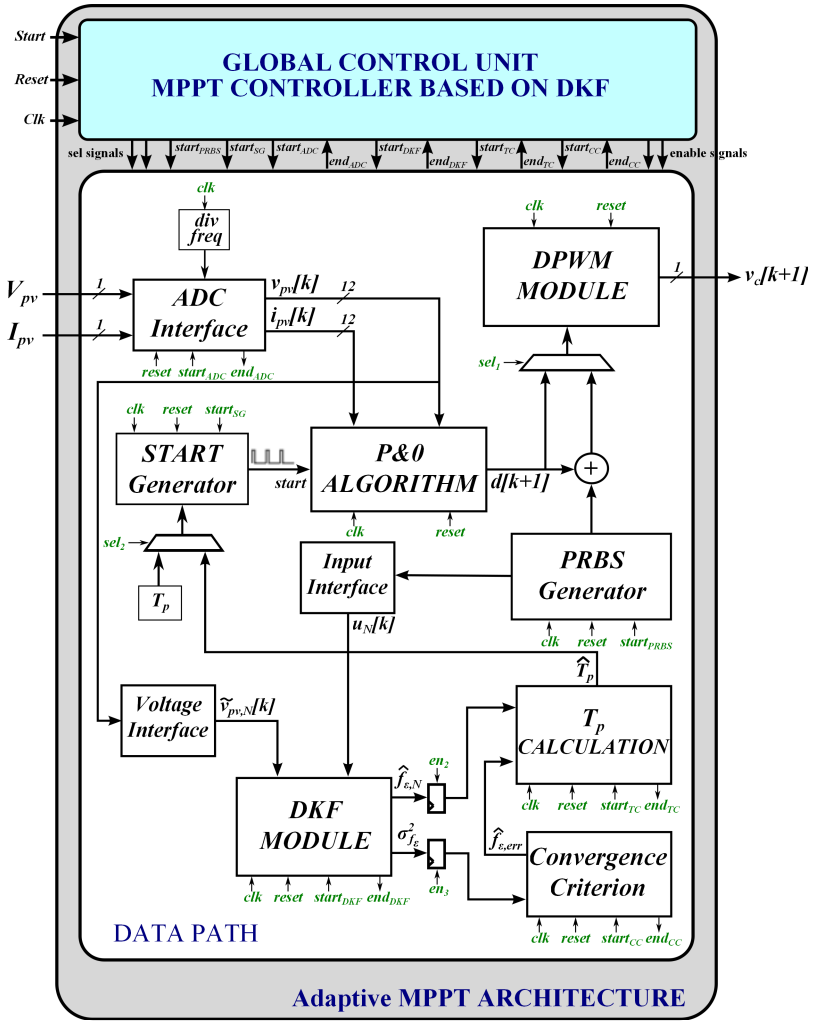


Figure 4.17 Hardware FPGA Architecture - adaptive MPPT controller based on DKF.

4.4.4 VHDL Coding

The designed FPGA architecture is programmed using the VHDL description language. Xilinx ISE design tool is used to this purpose. Different hierarchical VHDL files are programmed to preserve the modularity of the architecture. Structural VHDL is used then, each VHDL file corresponds with a given hardware module, used as a com-

ponent in the modules positioned in a higher hierarchical level (as in Figure 4.16). The *ADC interface* is included in the final project. It allows recovering the serial signal from the ADC and reconstructing the digital data for the treatment in the hardware FPGA architecture. The *ChipScope interface* is also implemented, allowing to catch some internal signal in order to perform a debugging of the architecture and to validate the implemented algorithm. Each module is validated individually before using it in the whole architecture. Finally, the algorithm is tested on the actual system and it is discussed in the next chapter.

4.4.5 Time/Area Performances Analysis

After having made the synthesis, the area analysis is carried out. It gives a summary of the necessary hardware resources for implementing the developed architecture. Table 4.10 shows the needed resources, showing the fulfilment of the area constraint.

Table 4.10 Consumed Hardware Resources of the Adaptive MPPT Controller based on DKF.

Resources	Available	Consumed	Percentage
Slice Registers	54576	20770	38 %
Slice LUTs	27288	15733	57 %
Occuped Slices	6822	6094	89 %
16-KByte RAM Blocks	116	99	85 %
DSP48A1s Slices	58	51	87 %

On the other hand, the execution time is determined by evaluating the latency of each module. Table 4.11 shows the latency and the corresponding execution time of the ADC block and the internal DKF modules.

Table 4.11 Execution Time for the DKF architecture.

	Latency	Ex. time
ADC Conversion t_{ADC}	16 $T_{clk,ADC}$	1.28 μs
State Prediction t_{SP}	$N_{SP} = 29$	0.29 μs
State Innovation t_{SI}	$N_{SI} = 123$	1.23 μs
Parameter Innovation t_{PI}	$N_{PI} = 192$	1.92 μs
TOTAL		**3.20** μs

where $T_{clk,ADC}$ is the clock of the ADC blocks, being equal to 12.5 MHz. According to (4.3), the execution time of the DKF algorithm is equal to 3.2 μs.

Hence, the implemented hardware architecture fulfils both the area constraint and the timing constraint, while preserving as well the modular constraint.

4.5 Conclusions

In this chapter, the FPGA hardware architectures of the algorithms developed in Chapter 3 have been presented. For this purpose, the design methodology shown in Section 2.3.3 has been adopted.

First of all, a pre-evaluation of the time/area performances has been made. It is aimed at verifying if the adaptive MPPT controllers can be implemented preserving the whole parallelism. At this stage it has been observed that the CCM-based adaptive MPPT controller does not require any optimizations and then, it can be implemented directly on the chosen platform. On the other hand, the adaptive MPPT controller based on DKF requires additional optimizations since the corresponding architecture is too heavy to be implemented in a low-cost FPGA device. Thus, the pipelined multiplier IP and the A^3 methodology have been used for increasing the maximum allowable clock frequency and facing the mismatching between the required resources and the available ones. The final architectures fulfil both the timing and area constraints and then they are ready for the implementation process.

The two identification procedures (CCM and DKF) are now compared in terms of execution time and required resources:

- Concerning the hardware resources, the CCM does not require any additional efforts its optimization is done during the algorithm development. On the other hand, the complexity of the DKF algorithm (see Table 4.7) has made necessary some architecture optimizations. The final hardware resources required for implementing the adaptive controllers are summarized in Table 4.5 and Table 4.10.

- The CCM is a batch method, thus it does not exhibit huge issue with regard to the execution time. The same cannot be said about the DKF. Indeed, it is a recursive identification method and then it requires an execution time lower than the sampling period.

In conclusion, it is possible to state that the adaptive MPPT controller based on DKF is heavier than the one based on the CCM and it works properly only when the system is well modelled. However, it guarantees an additional safety margin, concerning the perturbation period estimate, and it supplies also the confidence of the estimation, useful for diagnosis or monitoring purposes.

The last step concerns the experimental validation of both the adaptive MPPT controllers.

Chapter 5

Fully FPGA-based Implementation of Adaptive Digital Controller for PV Application - Experimental Results

5.1 Introduction

This chapter deals with the final experimental validation of the developed FPGA-based adaptive MPPT controllers. First of all, the experimental platform is illustrated. After that, in Section (5.3), the validation of the adaptive MPPT controller based on CCM is afforded. Here the CCM identification procedure is experimentally validated and some tests are performed for evaluating the robustness of the identification technique against parameter and irradiance variation. Moreover, the whole adaptive controller is verified and some experimental results are shown. Finally, Section (5.4) copes with the experimental validation of the adaptive MPPT controller based on DKF. As for the previous adaptive controller, firstly the identification procedure is validated and a robustness analysis is carried out. Then, the effectiveness of the adaptive controller is confirmed by experimental tests.

5.2 Overview of the Test Bench

The experimental set up is shown in Figure 5.1. According to the system specification (Section 2.3.3), the system under test is composed of a boost dc/dc power converter connected to its input to a Kyocera KC120-1 PV module and to its output to a 36 V battery. The adaptive P&O MPPT controllers have been implemented in a XC6SLX45 Xilinx Spartan-6 FPGA. The sampling and the measurements of the PV voltage and the PV current are realized by two 12-bits ADCs. This validation is necessary to verify the feasibility of the proposed adaptive P&O MPPT controller implementation.

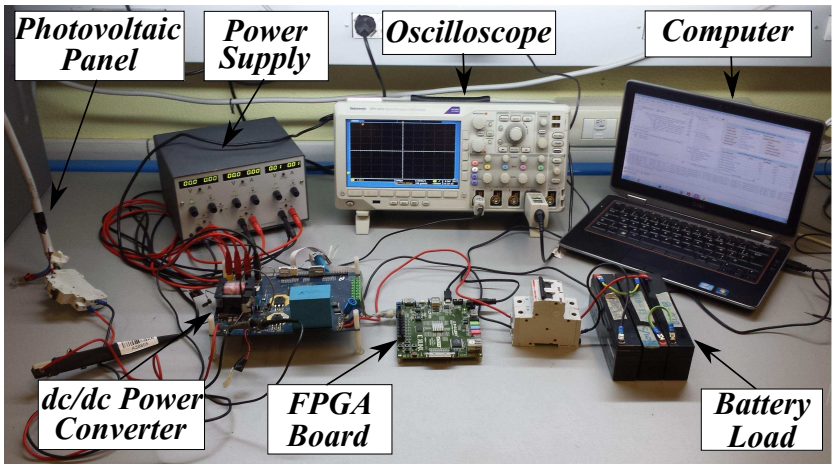


Figure 5.1 Prototyping Platform

Table 5.1 summarizes the parameters of the Kyocera KC120-1 PV panel. The dc/dc power converter components are summarized in Table 2.3. The operating conditions of the PV source and the switching frequency of the power converter are described in Table 5.3. After that, Table 5.3 reports the characteristics of the ADC board.

Table 5.1 Kyocera KC120-1 PV panel

Parameter	Absolute Value
Maximum Power	120 W
Open Circuit PV Voltage V_{OC}	21.5 V
Short Circuit PV Current I_{OC}	7.45 A

Table 5.2 Operating Conditions

Parameter	Absolute Value
Measured Irradiation Condition	500 $\frac{W}{m^2}$
PV Voltage @ MPP	16 V
PV Current @ MPP	3 A
Differential Resistance r_d @ MPP	$-V_{MPP}/I_{MPP} = -16/3 \simeq -5.33\Omega$
Converter switching frequency f_s	200 kHz

Table 5.3 ADC122S706

Parameter	Absolute Value
Resolution	12 bits
Maximum Conversion rate	1 MSPS
Clock Frequency $f_{clk,ADC}$	8 MHz to 16 MHz
Supply Voltage V_A	+4.5 V to +5.5 V
Digital Voltage V_D	+2.7 V to V_A
Reference Voltage V_{ref}	+1.0 V to V_A

Finally, the ATLYS Digilent board features are summarized:

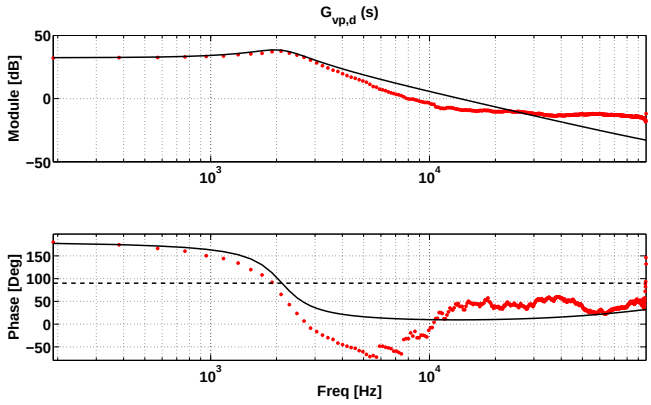
- ▷ Xilinx Spartan-6 XC6SLX45 FPGA
- ▷ 100 MHz CMOS oscillator
- ▷ 128 Mbyte DDR2
- ▷ 16 Mbyte x4 Quad SPI Flash
- ▷ 6 phased-locked loops
- ▷ USB-UART host port

5.3 Experimental Results of Adaptive MPPT Controller based on CCM

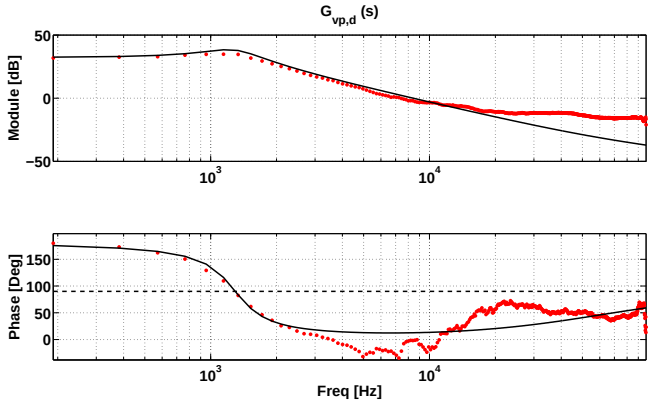
This section is aimed at validating the proposed adaptive MPPT controller based on CCM. Experimental tests have been carried out, using two different values of the capacitance that is placed in parallel to the PV module. In this way the effectiveness of the real time optimization in terms of settling time evaluation can be shown. First of all, the validation of the CCM has been demonstrated and then, the self-adaptivity of the MPPT controller with respect to the converter parameters has been put into evidence.

5.3.1 Validation of the Cross-Correlation Method

With respect to the initial value of $C = 50\,\mu F$, the converter input capacitance has been increased up to the value of $C = 134\,\mu F$. This large variation can be seen as an uncertainty equal to $\pm 50\%$ on a nominal $C = 90\,\mu F$ value. The $G_{vp,d}$ transfer function estimated experimentally by means of the CCM for the two different values of capacitance is shown in Figure 5.2. Such results are also compared with the corresponding transfer functions calculated with the linear circuit model shown in Figure 2.16. The good agreement obtained in the low frequency range results into an accurate estimation of the settling time by using the resonance frequency ω_n, the static gain $\left|G_{vp,d}(j0)\right|$ and the resonance peak amplitude $\left|G_{vp,d}(j\omega_n)\right|$, all extracted from the estimated $G_{vp,d}$. In the high frequency range the analytical model has a limited validity and the experimental data are affected by noise, so that a worse fitting to the second order theoretical model is obtained. However, it can be improved by using emphasis and de-emphasis filters, as explained in Section 3.3.4.



(a)



(b)

Figure 5.2 Transfer function evaluated by means of the CCM and the FFT algorithm for: (a) case #1 and (b) case #2 listed in Table 5.4. Red points are the experimental results, black curves are obtained by means of the linear model.

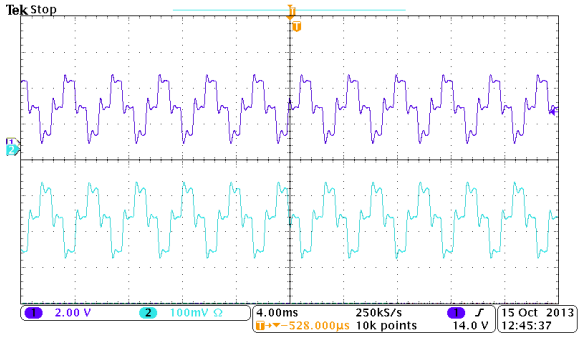
5.3.2 Validation of the Adaptive MPPT Controller

Starting from the identified frequency response, the required parameters can be estimated as described in Section 2.2.2. The results are summarized in Table 5.4. In all of the experiments, the step-size of the P&O MPPT has been selected by following the guidelines shown in [1], so that its value has been settled at $\Delta d = 0.03125$.

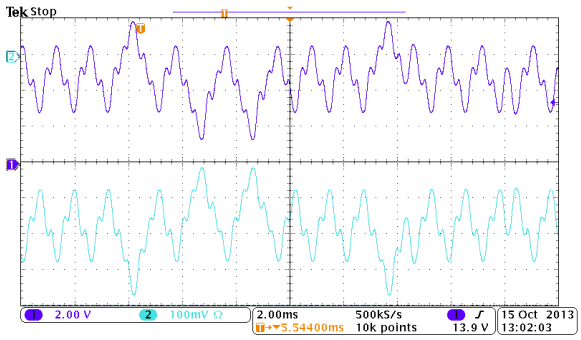
Table 5.4 Experimental results.

	Case #1 - Fig. 5.3	Case #2 - Fig. 5.4
C [μF]	50	134
Estimated Settling Time T_ε [μs]	880	1300

Figure 5.3(a) shows the behavior of the PV voltage and current when $C = 50\mu F$ and T_p is settled to its optimal value $T_p = T_\varepsilon$ calculated by the algorithm running on the FPGA. In this case the system exhibits a stable steady-state, i.e. it presents a stable three-point behaviour [1]. This confirms that the parameter T_p is correctly settled.



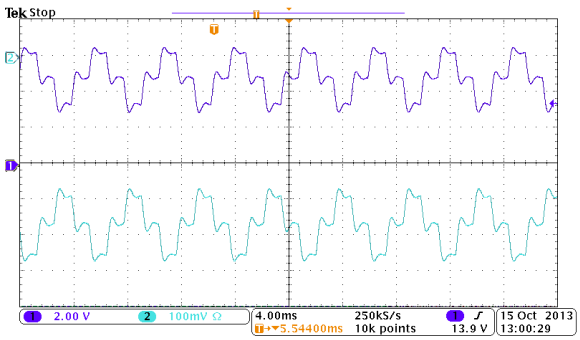
(a)



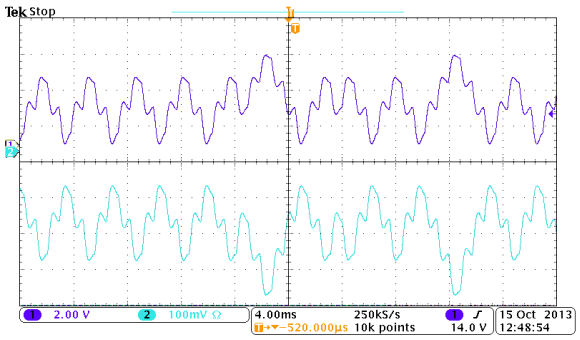
(b)

Figure 5.3 MPPT behaviour by using $C = 50\mu$F: (a) $T_p = T_\varepsilon = 880\mu$s, (b) $T_p = 310\mu$s.

Figure 5.3(b) shows the behaviour of the system in the same condition as above, except for a different value of the perturbation period



(a)



(b)

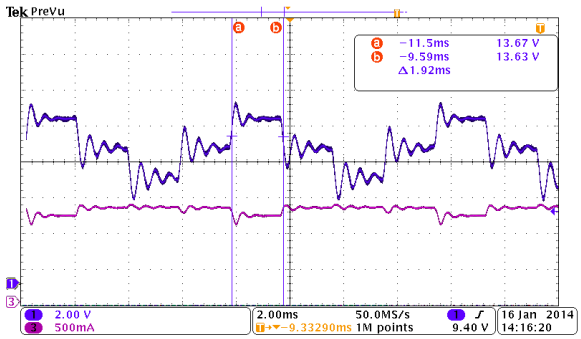
Figure 5.4 MPPT behaviour by using $C = 134\mu$F: (a) $T_p = T_\varepsilon = 1300\mu$s, (b) $T_p = 880\mu$s.

$T_p = 310\mu s < T_\varepsilon$. In this case the PV voltage and current oscillate with an unpredictable behaviour. As demonstrated in Section 2.1.3, this behaviour reduces the MPPT efficiency.

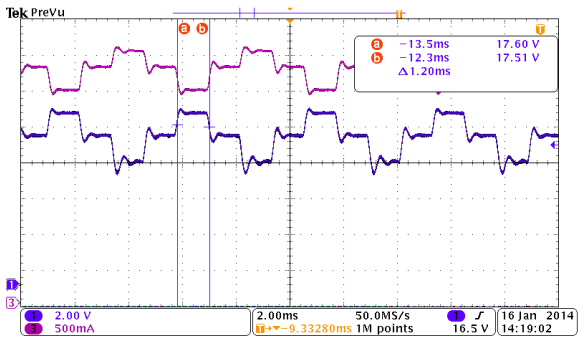
In the second experimental test ($C = 134\mu F$), the following settling time $T_\varepsilon = 1300\mu$s has been estimated. The results of this test are shown in Figure 5.4. As in the previous experimental case, the condition $T_p = T_\varepsilon$ ensures a stable system steady-state, whereas a perturbation period lower than the estimated settling time is not able to ensure a stable three-point steady-state of the system. It is worth to note that the example shown in Figure 5.4(b) has been obtained by using the perturbation period equal to the T_p optimal value used for the case shown in Figure 5.4(a).

In Figure 5.5 the real-time T_p identification is shown for low and high irradiance values: the procedure is triggered periodically, in the experiments every sixty seconds. As expected, the perturbation time is adapted accordingly: $T_p = 1.92ms$ at low irradiance and $T_p = 1.20ms$ at high irradiance.

All the tests confirm that the optimal selection of the perturbation period depends strongly on the PV system under analysis. A T_p tuned for a specific operating condition might be inadequate in other conditions or in presence of parametric drifts or aging effects.



(a)



(b)

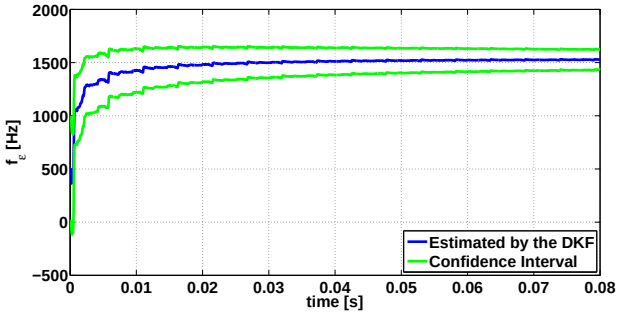
Figure 5.5 MPPT behaviour by using $C = 50\mu$F: (a) Low irradiance conditions, (b) High irradiance conditions.

5.4 Experimental Results of Adaptive MPPT Controller based on Kalman Filter

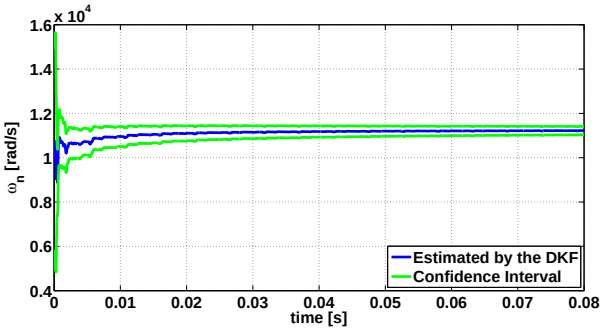
The aim of this section is to validate the adaptive MPPT controller based on DKF. First of all, the DKF has been validated and several tests have been carried out in order to demonstrate its robustness. After that, the whole adaptive MPPT controller has been tested by imposing the estimated perturbation period.

5.4.1 Validation of the Dual Kalman Filter

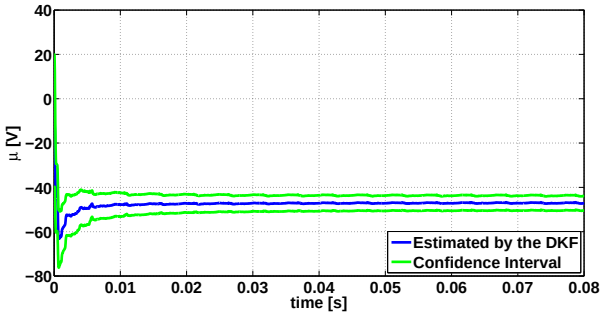
The experimental tests, presented in this section, are aimed at validating the implemented DKF. The experimentations have been carried out on the test bench presented in Section 5.2. During the identification procedure, the MPPT algorithm has been frozen and the PRBS has been superimposed to the converter duty-cycle. The corresponding small signal panel voltage has been fed to the DKF. As explained, it works recursively and it allows estimating the states and the parameters (f_ε, ω_n and μ) of the PV system under analysis. Figure 5.6 shows the parameter estimations and their corresponding confidence interval. The latter decreases with the time giving an increasing reliability of the estimation.



(a)



(b)



(c)

Figure 5.6 Parameters estimation with an input capacitance $C = 50\mu F$: (a) estimated f_ε, (b) estimated ω_n, (c) estimated μ

According to the proposed convergence criterion (Section 3.4.7), the time needed to reach the convergence T_{id} is equal to 5.9 ms as shown in Figure 5.7. Indeed, the Kalman filter reaches the convergence when the settling frequency standard deviation $\sigma_{\hat{f_\varepsilon}}$ (red line in figure) is lower than the acceptable error $f_{\varepsilon,err}$ (green line in figure).

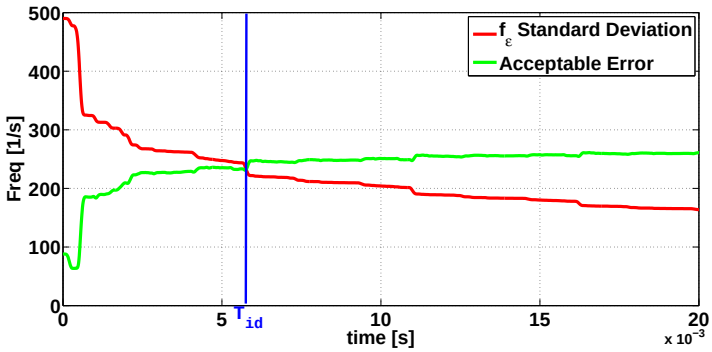


Figure 5.7 Filter Convergence with $C = 50\mu F$. Red line is $\sigma_{\hat{f_\varepsilon}}$; green line is f_ε imposing a relative error of 17.6%.

Only when the convergence is reached the estimation can be considered as reliable. Table 5.5 summarizes the estimated parameters at the convergence time.

Table 5.5 Estimated Parameters with an input capacitance equal to 50 μF.

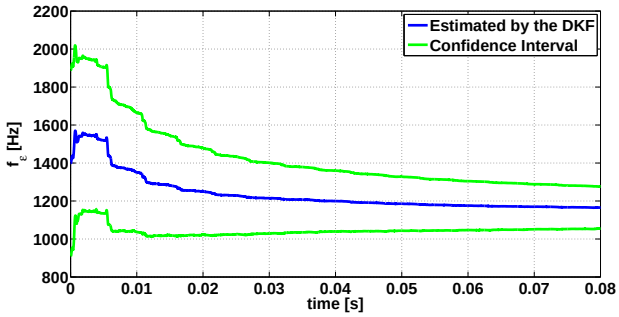
Estimated Parameter	Estimated Value
Estimated Settling Frequency f_ε @ T_{id}	1320 Hz
Estimated Natural Frequency ω_n @ T_{id}	10740 rad/s
Estimated DC Gain μ @ T_{id}	-47.48

5.4.2 Robustness of the Dual Kalman Filter

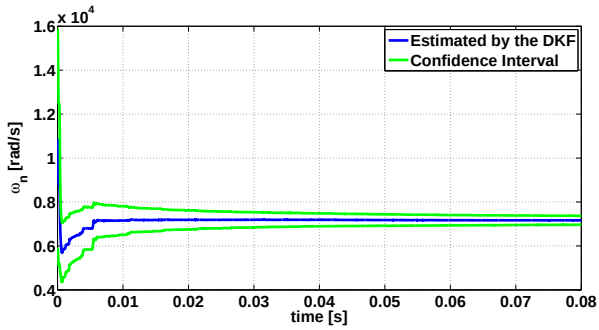
In this section the robustness of the DKF is afforded. Due to the environmental condition variations and the ageing of the components, system parameters can change. Thus, it has to demonstrate that the Kalman filter reaches the convergence also in these cases. Moreover, the final adaptive MPPT controller will must be able to identify the T_ε each time an adjustment of perturbation period is required. In these cases, the initial parameters in the Kalman filter will be set equal to the last estimation, then, the DKF has to be able to reach the convergence also with different starting points. Hence, the robustness of the Kalman filter in terms of parameter changes and initial conditions has to be verified. Firstly, the Kalman filter has been tested with another input capacitance, i.e. $C = 134\ \mu F$. After that, a test has been carried out in low irradiance condition, involving a differential resistance change. Finally, the DKF has been run with different initial conditions.

Different Input Capacitance

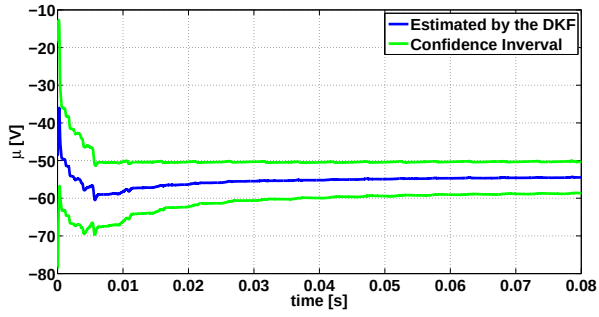
With respect to the initial capacitance value (50 μF), the converter input capacitance has been increased up to 134 μF during this experimental test. The PV parameters identified experimentally by means of the DKF are shown in Figure 5.8. It is evident that the DKF reaches its convergence also in this condition. However, as expected, the speed of convergence is slower having tuned the filter by using the nominal input capacitance. The convergence time can be deduced from Figure 5.9 and it is equal to 21.8 *ms*.



(a)



(b)



(c)

Figure 5.8 Parameters estimation with an input capacitance $C = 134\mu F$: (a) estimated f_ε, (b) estimated ω_n, (c) estimated μ

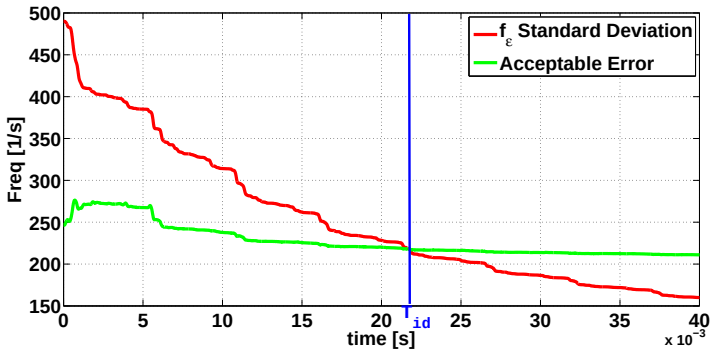


Figure 5.9 Filter Convergence with $C = 134\mu F$. Red line is $\sigma_{\hat{f_\varepsilon}}$; green line is f_ε imposing a relative error of 17.6%.

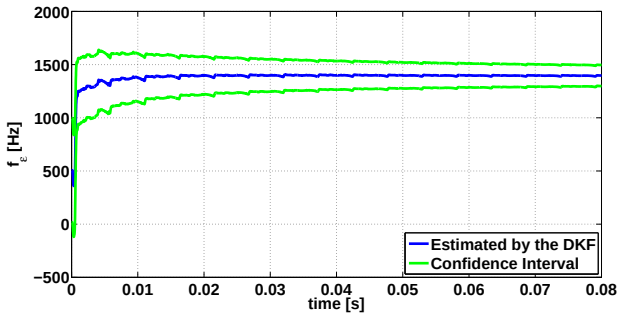
In Table 5.6 the estimated parameters in the case of $C = 134\mu F$ are compared with the one obtained with a 50 μF input capacitor.

Table 5.6 Estimated Parameters with two different input capacitance: Case 1 $C = 50\ \mu F$, Case 2 $C = 134\ \mu F$

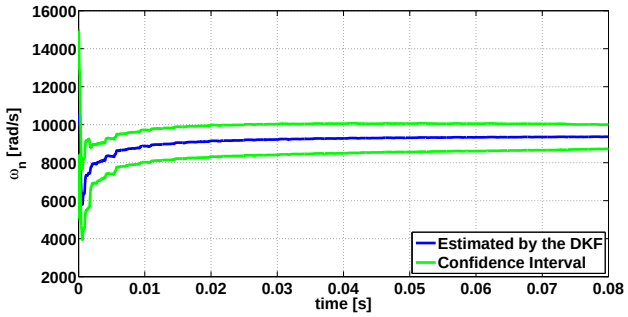
	Case 1	Case 2
Estimated Settling Frequency f_ε	1320 Hz	1230 Hz
Estimated Natural Frequency ω_n	10740 rad/s	7190 rad/s
Estimated DC Gain μ	-47.48	-55.97

Different Irrandiance Level

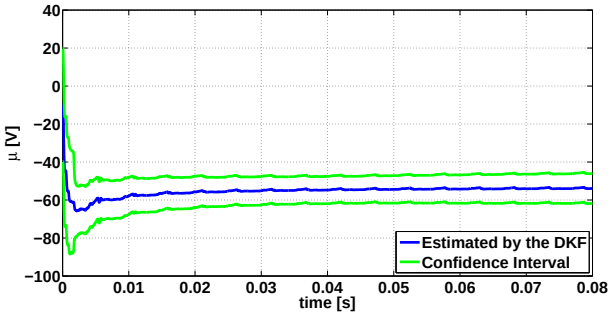
Up to now the tests have been carried out in presence of a high irradiance level. On the other hand, the identification results, presented in this section, have been obtained when low irradiance levels occur. The aim is to demonstrate that the developed Kalman filter is able to estimate the unknown parameters in different environmental condition and then with different differential resistances. Given the linear dependency of the short circuit current from the irradiance level, the test has been carried out when the PV current assumes low values. In particular, an I_{SC} equal to $0,5\ A$ has been measured during the test. Figure 5.10 shows the identified parameters. It is worth to note as, also in this case, the KF reaches the convergence. The time needed to the filter for the estimation procedure is equal to 7.4 ms, as shown in Figure 5.11.



(a)



(b)



(c)

Figure 5.10 Parameters estimation with a low irradiance condition: (a) estimated f_ε, (b) estimated ω_n, (c) estimated μ

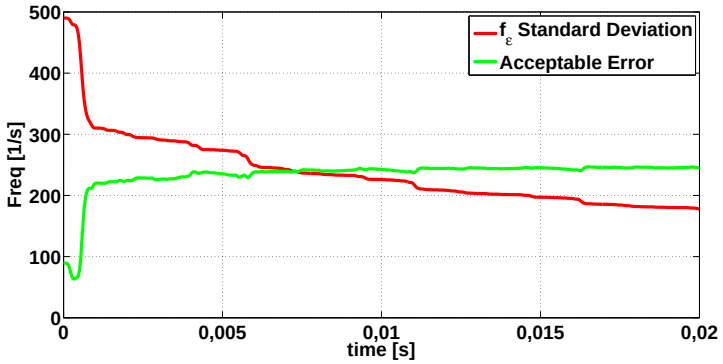


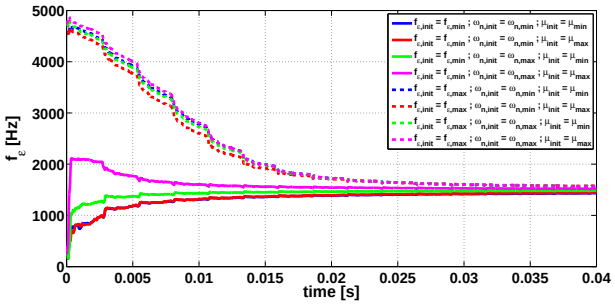
Figure 5.11 Filter Convergence with low irradiance condition. Red line is $\sigma_{\widehat{f_\varepsilon}}$; green line is f_ε imposing a relative error of 17.6%.

Different Initial Conditions

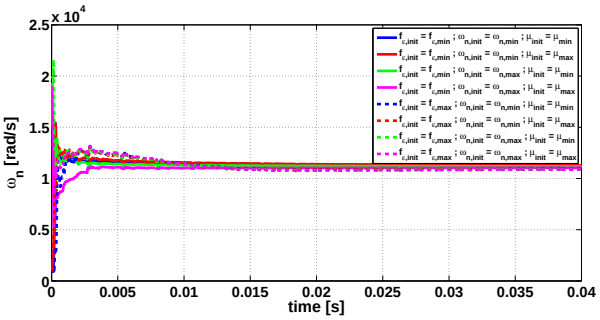
Table 5.7 shows the initial values of the DKF, adopted so far. In this section a vertex analysis is presented in order to demonstrate the robustness of the DKF to the initial value variations. Such an analysis has been performed as in section (3.4.9), i.e. a range from 5% to 95% of the maximum values has been considered for the initialization of the parameter estimations. As summarized in Table 3.14, the ranges of f_ε, ω_n and μ are $[250, 5000]$ Hz, $[1000, 19000]$ rad/s and $[-2.5, -47.5]$, respectively. Figure 5.12 proves the goodness of the estimation process: all parameter estimates tend to the same value.

Table 5.7 Initial Conditions

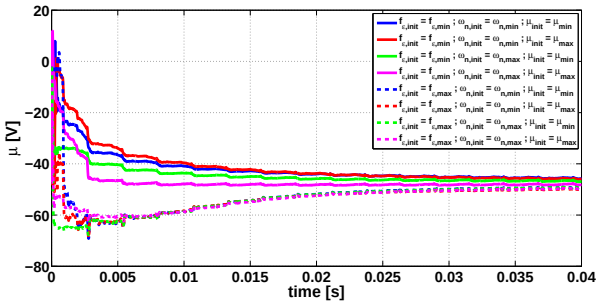
Estimation Parameters	Initial Value
$f_{\varepsilon,init}$	500 Hz
$\omega_{n,init}$	10000 rad/s
μ_{init}	-10



(a)



(b)



(c)

Figure 5.12 Parameters estimation with different initial conditions: (a) estimated f_ε, (b) estimated ω_n, (c) estimated μ.

5.4.3 Validation of the Adaptive MPPT Controller

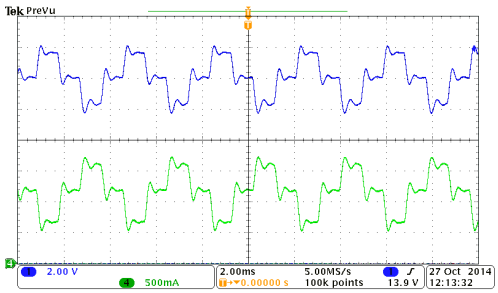
The objective now is to validate the adaptive MPPT controller based on DKF. To this aim, a first experimental test has been carried out by using the nominal input capacitance. After that, a second test has been performed with an increased input capacitance (134 μF) with initial conditions equal to the estimated parameters in the case of nominal capacitance. Starting from the parameter estimations, the minimum perturbation period has to be estimated in both cases. For this purpose, as explained previously in Chapter (2), the settling frequency and its variance are necessary to calculate the minimum perturbation period. Figure 5.6(a) shows the estimated settling frequency with its confidence interval in the case of $C = 50\ \mu F$, whereas Figure 5.12(a) depicts the case of $C = 134\ \mu F$.

The results, with both the capacitances, are summarized in Table 5.8. In all of the experiments, the step-size of the P&O MPPT has been selected according to the guidelines shown in [1], so that its value has been fixed to $\Delta d = 0.03125$. The minimum perturbation period is obtained thanks to eq. (4.7) and it is imposed in the MPPT module.

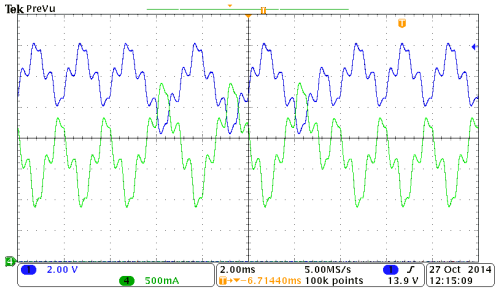
Table 5.8 Experimental Tests.

	Case #1 - Fig. 5.13	Case #2 - Fig. 5.14
C [μF]	50	134
Minimum T_p [μs]	940	990

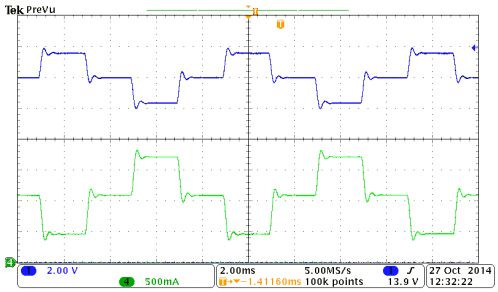
Figure 5.13(a) shows the behaviour of the PV voltage and current when the perturbation period T_p is settled to the estimated minimum perturbation period and the input capacitance is equal to $50\mu F$. It is worth to note that the system exhibits a stable steady-state behaviour. This means that a proper T_p has been settled. Choosing a smaller value of T_p provokes oscillations on the voltage and on the current, leading to an unpredictable behaviour, as shown in Figure 5.13(b). Figure 5.13(c) shows the behaviour of the MPPT algorithm when the perturbation period is too big.



(a)



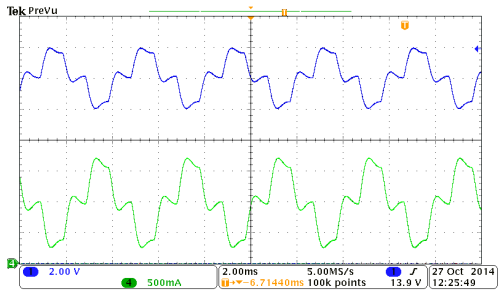
(b)



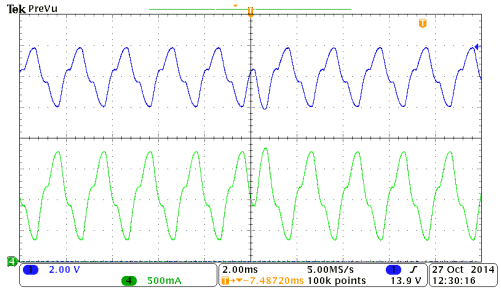
(c)

Figure 5.13 MPPT behaviour by using $C = 50\mu$F: (a) $T_p = T_{p,min} = 940\mu$s, (b) $T_p = 500\mu$s, (c) $T_p = 2ms$.

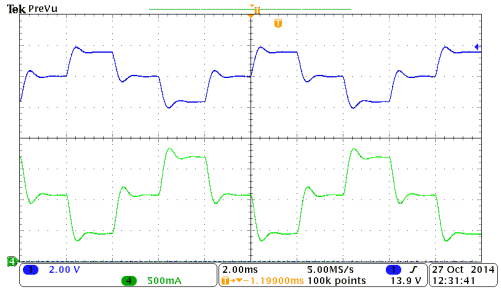
On the other hand, Figure 5.14(a) shows the behaviour of the MPPT algorithm with the perturbation period equal to the minimum perturbation period in the case of $C = 134\mu F$. As expected a stable steady-state behaviour is obtained. Otherwise, with a smaller value, an unpredictable behaviour is occurring, Figure 5.14(b). Figure 5.14(c) shows the behaviour with a bigger perturbation period.



(a)



(b)



(c)

Figure 5.14 MPPT behaviour by using $C = 134\mu$F: (a) $T_p = T_{p,min} = 990\mu$s, (b) $T_p = 500\mu$s, (c) $T_p = 2ms$.

These tests confirm the effectiveness of the proposed adaptive MPPT method and the estimated parameters can be used in diagnosis, monitoring or, as in the case proposed in this thesis, for adaptive control.

5.5 Conclusions

In this chapter, the experimental validation of the proposed fully FPGA-based adaptive MPPT controllers has been presented. At the beginning, an overview of the experimental test bench has been given. After that, the feasibility of the CCM identification procedure and the adaptive MPPT controller have been demonstrated. Finally, the DKF and the corresponding adaptive MPPT controller have been validated. In addition, the robustness against the parameter and irradiance variations has been tested.

Conclusions

General Conclusion

In this work two adaptive P&O MPPT control algorithms based on two different real-time identification procedures have been proposed. The aim has been to perform an on-line optimization of the perturbation period T_p in order to enhance the MPPT algorithm performances. Such controllers have been embedded on the same low-cost FPGA device in charge of controlling the power converter that processes the power produced by the PV array.

First of all it has been demonstrated that a proper choice of the value of T_p allows enhancing the MPPT algorithm performances. The need of a real-time identification procedure aimed at carrying out an on-line adaptation of such a parameter has been demonstrated. After having made an overview of the linear identification methods, two different techniques have been chosen: the Cross-Correlation Method (CCM) and the Dual Kalman Filter (DKF).

The CCM has been presented in Section (2.2.2): it belongs to the class of non-parametric approaches. No prior knowledge about the PV system is required and the developed algorithm can be applied to any dc/dc converter. Unfortunately, it is not able to give any indication about the reliability of the estimations provided. In this work, this technique has been adopted for evaluating the frequency response of a non-linear PV system that is operating at its MPP. Thus, an in-depth analysis for evaluating the effects of the non-linearities on the identified transfer function has been carried out. This study has led to the choice of a proper value of the perturbation signal amplitude.

Subsequently, in Section (2.2.3), the DKF has been presented. It is a model-based technique that estimates the states and the parameters of the PV model. The method is able to give a confidence interval that guarantees an additional security margin to the adaptive controller. However, reliable results are achieved only if the system is properly modelled and the consistency of the filter is ensured.

The usefulness of FPGA devices for PV identification application has been also demonstrated. Such devices exploit the inherent parallelism of the algorithm and then enhance the timing performances of the identification techniques. This allows real-time identification and then the on-line optimization of the P&O MPPT algorithm. The development of the adaptive MPPT controllers has been achieved by using an appropriate design methodology presented in Section (2.3.3). It is composed by four main steps.

Firstly hardware specifications have been presented. The second task has concerned the development of the algorithm (Chapter 3) composed by the modular partitioning, the digital realization, the algorithm optimizations and the validation by using Matlab/Simulink tool. Concerning the CCM, a transformation module has been designed in order to perform the Fast-Walsh Hadamar Transform and the Fast Fourier Transform. This choice has guaranteed the proper compromise between the execution time and the required resources. Furthermore, in order to enhance the quality of the identification, the impulse response truncation and the smoothing technique have been adopted. As far as the DKF is concerned, an analysis for choosing the better PV model has been made. This study has led to the choice of the small signal PV model. In order to reduce the number of multiplications and divisions of the model, the settling frequency has been estimated instead of the settling time and proper base-values have been chosen for the internal variables of the DKF. A dedicated filter tuning has been made for achieving the required consistency of the filter. The robustness of the both identification methods has been also demonstrated.

The third step has been devoted to the FPGA architecture development (Chapter 4). During this task an evaluation of the time/area performance of the whole adaptive algorithms has been made and additional architecture optimizations have been employed for the DKF

algorithm. In particular pipelined Xilinx multipliers have been used for increasing the maximum allowable clock frequency. Furthermore, the Algorithm Architecture Adequation has been adopted leading to a reduction of the required hardware resources. Concerning the CCM, having chosen a pipeline structure for the transformation module, no additional architecture optimizations have been necessary. After that, the VHDL coding and a time/area analysis have been made. The two identification architectures have been compared in terms of execution time and consumed hardware resources. Both identification techniques satisfy the given timing requirements and have been efficiently implemented in the chosen low-cost FPGA.

Finally, in Chapter (5), the implemented identification techniques and the corresponding adaptive controller have been validated through experimental tests. In both cases, the estimated perturbation period ensures a stable MPPT behaviour while enhancing the P&O MPPT algorithm performances.

Perspectives

The main perspectives of this work are listed below from the one which might be reached more easily to the one which requires more efforts:

- In this work only the adaptation of the perturbation period has been made. Then, a possible perspective is to adapt simultaneously, by using the proposed identification techniques, the perturbation amplitude and the perturbation period in order to improve the MPPT efficiency.
- A suitable procedure triggering the MPPT optimization process would improve the overall performances of the control strategy.
- As far as the FPGA development is concerned, a System on Chip (SoC) solution can be adopted for taking advantages by the combination of software and hardware solutions.
- The proposed identification techniques can be used for monitoring the PV system, or for the diagnosis the state of health of both

the PV source and the associated switching converter.

- It is expected to use the proposed identification techniques also in other fields, related to the green energy world, e.g. battery management systems, wind energy systems and fuel cell diagnosis.

Papers

Journal Papers

1. P. Manganiello, M. Ricco, G. Petrone, E. Monmasson, G. Spagnuolo, "Optimization of perturbative PV MPPT methods through on line system identification", *IEEE Transactions on Industrial Electronics*, Vol. 61, No. 12, pp. 6812 - 6821, 2014.

Conference Papers

1. M. Ricco, "On-line Identification of dc/dc Converter for Photovoltaic Applications", JCGE (Journées des Jeunes Chercheurs en Génie Electrique), Saint-Nazaire, France, 5-6 June, 2013.

2. P. Manganiello, M. Ricco, E. Monmasson, G. Petrone, G. Spagnuolo, "On-line optimization of the P&O MPPT method by means of the system identification", IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10-13 November, 2013.

3. M. Ricco, P. Manganiello, G. Petrone, E. Monmasson, G. Spagnuolo, "FPGA-Based Implementation of an Adaptive P&O MPPT Controller for PV Applications", 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Instanbul, Turkey, 1-4 June, 2014.

Bibliography

[1] N. Femia, G. Petrone, G. Spagnuolo, and M.Vitelli. *"Power Electronics and Control Techniques for Maximum Energy Harvesting in Photovoltaic Systems"*. CRC Press, 2013.

[2] Ljung Lennart. *"System Identification: Theory for the User Second Edition"*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[3] Sullivan Royer, Stéphane Thil, Thierry Talbert, and Monique Polit. "A procedure for modeling buildings and their thermal zones using co-simulation and system identification". *Energy and Buildings*, 78(0):231 – 237, 2014.

[4] Xiwang Li and Jin Wen. "building energy consumption on-line forecasting using physics based system identification". *Energy and Buildings*, 82(0):1 – 12, 2014.

[5] G. Reynders, J. Diriken, and D. Saelens. "quality of grey-box models and identified parameters as function of the accuracy of input and observation signals". *Energy and Buildings*, 82(0): 263 – 274, 2014.

[6] Kebin Yuan, Qining Wang, and Long Wang. "fuzzy-logic-based terrain identification with multisensor fusion for transtibial amputees". *Mechatronics, IEEE/ASME Transactions on*, 20(2): 618–630, April 2015.

[7] El Houssin El Bouchikhi, Vincent Choqueuse, and Mohamed Benbouzid. "induction machine faults detection using stator

current parametric spectral estimation". *Mechanical Systems and Signal Processing*, 52-53(0):447–464, 2015.

[8] H. Suwanwiwat, Vu Nguyen, M. Blumenstein, and U. Pal. "off-line handwritten thai name recognition for student identification in an automated assessment system". In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 2347–2353, July 2014.

[9] Naveed Iqbal, Azzedine Zerguine, and Naofal Al-Dhahir. "decision feedback equalization using particle swarm optimization". *Signal Processing*, 108(0):1 – 12, 2015.

[10] Li Niu, Dianguo Xu, Ming Yang, Xianguo Gui, and Zijian Liu. "on-line inertia identification algorithm for pi parameters optimization in speed loop". *Power Electronics, IEEE Transactions on*, 30(2):849–859, Feb 2015.

[11] Melanie N. Zeilinger, Davide M. Raimondo, Alexander Domahidi, Manfred Morari, and Colin N. Jones. "on real-time robust model predictive control". *Automatica*, 50(3):683 – 694, 2014.

[12] Wenxin Liu, Li Liu, Il-Yop Chung, David A. Cartes, and Wei Zhang. "modeling and detecting the stator winding fault of permanent magnet synchronous motors". *Simulation Modelling Practice and Theory*, 27(0):1 – 16, 2012.

[13] Shinji Ichikawa, Mutuwo Tomita, and Shinji Doki. "Sensorless Control of Permanent-Magnet Synchronous Motors Using Online Parameter". *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 53(2):363–372, 2006.

[14] Lahoucine Idkhajine, Eric Monmasson, and Amira Maalouf. "Fully FPGA-based sensorless control for synchronous AC drive using an Extended Kalman Filter". *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 59(10):3908–3918, 2012.

[15] V. Valdivia, A. Barrado, A. Lazaro, M. Sanz, D. Lopez del Moral, and C. Raga. "black-box behavioral modeling and identification of dc-dc converters with input current control for fuel cell power conditioning". *Industrial Electronics, IEEE Transactions on*, 61(4):1891–1903, April 2014.

[16] A. Vidal, A.G. Yepes, F.D. Freijedo, J. Malvar, O. Lopez, and J. Doval-Gandoy. "a technique to estimate the equivalent loss resistance of grid-tied converters for current control analysis and design". *Power Electronics, IEEE Transactions on*, 30(3):1747–1761, March 2015.

[17] Yan-Fei Liu, E. Meyer, and Xiaodong Liu. "recent developments in digital control strategies for dc/dc switching power converters". *Power Electronics, IEEE Transactions on*, 24(11): 2567–2577, Nov 2009.

[18] Torsten Soderstrom and Petre Stoica. *"System Identification"*. 2011 edition, 1989.

[19] Oliver Nelles. *"Non-linear System Identification"*. 2001.

[20] M.M. Peretz and S. Ben-Yaakov. "time-domain identification of pulse-width modulated converters". *Power Electronics, IET*, 5(2):166–172, Feb 2012.

[21] A.M. Dhirde, N.V. Dale, H. Salehfar, M.D. Mann, and T.H. Han. "equivalent electric circuit modeling and performance analysis of a pem fuel cell stack using impedance spectroscopy". *Energy Conversion, IEEE Transactions on*, 25(3): 778–786, Sept 2010.

[22] R. Al Nazer, V. Cattin, P. Granjon, M. Montaru, and M. Ranieri. "broadband identification of battery electrical impedance for hevs". *Vehicular Technology, IEEE Transactions on*, 62(7): 2896–2905, Sept 2013.

[23] Tom Breugelmans, Els TourwÃ©, Yves Van Ingelgem, Jan Wielant, Tom Hauffman, RenÃ© Hausbrand, Rik Pintelon, and

Annick Hubin. "odd random phase multisine {EIS} as a detection method for the onset of corrosion of coated steel". *Electrochemistry Communications*, 12(1):2 – 5, 2010.

[24] G. Dotelli, R. Ferrero, P.G. Stampino, and S. Latorrata. "analysis and compensation of pem fuel cell instabilities in low-frequency eis measurements". *Instrumentation and Measurement, IEEE Transactions on*, 63(7):1693–1700, July 2014.

[25] Mariko Shirazi, Jeffrey Morroni, Arseny Dolgov, Regan Zane, and Dragan Maksimović. "Integration of Frequency Response Measurement Capabilities in Digital Controllers for DC-DC Converters". *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 23(5):2524–2535, 2008.

[26] Muhammad Asif Zahoor Raja and Naveed Ishtiaq Chaudhary. "two-stage fractional least mean square identification algorithm for parameter estimation of {CARMA} systems". *Signal Processing*, (0):–, 2014.

[27] Feng Ding. "decomposition based fast least squares algorithm for output error systems". *Signal Processing*, 93(5):1235 – 1242, 2013.

[28] Robbert van Herpen, Tom Oomen, and Maarten Steinbuch. "optimally conditioned instrumental variable approach for frequency-domain system identification". *Automatica*, 50(9): 2281 – 2293, 2014.

[29] John Bound, David A. Jaeger, and Regina M. Baker. Problems with instrumental variables estimation when the correlation between the instruments and the endogenous explanatory variable is weak. *Journal of the American Statistical Association*, 90 (430):443–450, 1995.

[30] R E Kalman. "A New Approach to Linear Filtering and Prediction Problems". *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[31] François Auger, Mickael Hilairet, Josep M Guerrero, Eric Monmasson, Teresa Orlowska-kowalska, and Seiichiro Katsura. "Industrial Applications of the Kalman Filter : A Review". *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 60(12): 5458–5471, 2013.

[32] Gregory L. Plett. "extended kalman filtering for battery management systems of lipb-based {HEV} battery packs: Part 3. state and parameter estimation". *Journal of Power Sources*, 134 (2):277 – 292, 2004.

[33] K.K.C. Yu, N.R. Watson, and J. Arrillaga. "An Adaptive Kalman Filter for Dynamic Harmonic State Estimation and Harmonic Injection Tracking". *IEEE Transactions on Power Delivery*, 20(2):1577–1584, April 2005.

[34] Maria Carmela Di Piazza, Massimiliano Luna, and Gianpaolo Vitale. Dynamic pv model parameter identification by least-squares regression. *IEEE Journal of Photovoltaics*, 3(2):799–806, 2013.

[35] Loredana Cristaldi, Marco Faifer, Marco Rossi, and Ferdinanda Ponci. "A Simple Photovoltaic Panel Model : Characterization Procedure and Evaluation of the Role of Environmental Measurements". *IEEE Transactions on Instrumentation and Measurement*, 61(10):2632–2641, 2012.

[36] Byung-kwan Kang, Seung-tak Kim, Sun-Ho Bae, and Jung-Wook Park. "Diagnosis of Output Power Lowering in a PV Array by Using the Kalman-Filter Algorithm". *IEEE TRANSACTIONS ON ENERGY CONVERSION*, 27(4):885–894, 2012.

[37] Weidong Xiao, Magnus G J Lind, William G Dunford, and Antoine Capel. "Real-Time Identification of Optimal Operating Points in Photovoltaic Power Systems". *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 53(4):1017–1026, 2006.

[38] Abir Chatterjee, Ali Keyhani, and Dhruv Kapoor. "Identification of Photovoltaic Source Models". *IEEE TRANSACTIONS ON ENERGY CONVERSION*, 26(3):883–889, 2011.

[39] a. K. Abdelsalam, Shu Goh, O. Abdelkhalik, S. Ahmed, and A. Massoud. "Iterated unscented Kalman filter-based maximum power point tracking for photovoltaic applications". In *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pages 1685–1693. Ieee, November 2013. ISBN 978-1-4799-0224-8.

[40] Byung O Kang, Student Member, and Jae H Park. "Kalman Filter MPPT Method for a Solar Inverter". In *Power and Energy Conference at Illinois (PECI), 2011 IEEE*, pages 1 – 5, 2011.

[41] Ursula Eicker. *"Solar Technologies for Buildings"*. John Wiley & Sons, Ltd, Chichester, UK, June 2003.

[42] E. Romero-Cadaval, G. Spagnuolo, L. Garcia Franquelo, C.A. Ramos-Paja, T. Suntio, and W.M. Xiao. "grid-connected photovoltaic generation plants: Components and operation". *IEEE Ind. Electron. Mag.*, 7(3):6–20, 2013.

[43] Nicola Femia, Giovanni Petrone, Giovanni Spagnuolo, and Massimo Vitelli. "Optimization of Perturb and Observe Maximum Power Point Tracking Method". *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 20(4):963–973, 2005.

[44] L. Piegari and R. Rizzo. "adaptive perturb and observe algorithm for photovoltaic maximum power point tracking". *Renewable Power Generation, IET*, 4(4):317–328, 2010.

[45] Fan Zhang, K. Thanapalan, A. Procter, S. Carr, and J. Maddy. "adaptive hybrid maximum power point tracking method for a photovoltaic system". *IEEE Trans. Energy Conversion*, 28(2): 353–360, 2013.

[46] A. Al Nabulsi and R. Dhaouadi. "efficiency optimization of a dsp-based standalone pv system using fuzzy logic and dual-mppt control". *IEEE Trans. Ind. Informat.*, 8(3):573–584, 2012.

[47] P.E. Kakosimos, A.G. Kladas, and S.N. Manias. "fast photovoltaic-system voltage- or current-oriented mppt employing a predictive digital current-controlled converter". *IEEE Trans. Ind. Electron.*, 60(12):5673–5685, 2013.

[48] R. Kadri, J-P Gaubert, and G. Champenois. "an improved maximum power point tracking for photovoltaic grid-connected inverter based on voltage-oriented control". *IEEE Trans. Ind. Electron.*, 58(1):66–75, 2011.

[49] Yuanye Xia, K.H. Ahmed, and B.W. Williams. "wind turbine power coefficient analysis of a new maximum power point tracking technique". *IEEE Trans. Ind. Electron.*, 60(3):1122–1132, 2013.

[50] L. Belhadji, S. Bacha, I. Munteanu, A. Rumeau, and D. Roye. "adaptive MPPT applied to variable-speed microhydropower plant". *IEEE Trans. Energy Conversion*, 28(1):34–43, 2013.

[51] Yuncong Jiang, Jaber A Abu Qahouq, and Tim A Haskew. "Adaptive Step Size With Adaptive-Perturbation- Frequency Digital MPPT Controller for a Single-Sensor Photovoltaic Solar System". *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 28(7):3195–3205, 2013.

[52] Fang-Lin Luo and Hong Ye. "small signal analysis of energy factor and mathematical modeling for power DC-DC converters". *IEEE Trans. Power Electron.*, 22(1):69–79, 2007.

[53] Gregory L. Plett. "extended kalman filtering for battery management systems of lipb-based {HEV} battery packs: Part 1. background". *Journal of Power Sources*, 134(2):252 – 261, 2004.

[54] Gregory L. Plett. "extended kalman filtering for battery management systems of lipb-based {HEV} battery packs: Part 2. modeling and identification". *Journal of Power Sources*, 134 (2):262 – 276, 2004.

[55] K.R. Godfrey, H.A. Barker, and A.J. Tucker. "comparison of perturbation signals for linear system identification in the frequency domain". *Control Theory and Applications, IEE Proceedings* -, 146(6):535–548, Nov 1999.

[56] N. Zierler. "linear recurring sequences". *Journal of the Society for Industrial and Applied Mathematics*, 7(1):31–48, 1959.

[57] Keith Godfrey. *"Perturbation Signals for System Identification"*. 1993.

[58] D.D. Koleske and S.J. Sibener. "generation of pseudorandom sequences for use in cross-correlation modulation". *Review of Scientific Instruments*, 63(8):3852–3855, Aug 1992.

[59] Z. Syroka, T. Zajac, and P. Dubi?owicz. "generation of linear maximum length sequences". In *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*, pages 309–313, Oct 2010.

[60] A J Fainveathert, M P Foster, and D A Stone. "VRLA BATTERY PARAMETER IDENTIFICATION USING PSEUDO RANDOM BINARY SEQUENCES (PRBS)". In *Power Electronics, Machines and Drives (PEMD 2010), 5th IET International Conference on*, pages 1–6, 2010.

[61] Matti Vilkko and Tomi Roinila. "Designing Maximum Length Sequence Signal for Frequency Response Measurement of Switched Mode Converters". In *Nordic Workshop on Power and Industrial Electronics*, 2008.

[62] Botao Miao, Regan Zane, and Dragan Maksimovic. "Practical On-Line Identification of Power Converter Dynamic Responses". In *Applied Power Electronics Conference and Exposition, 2005. APEC 2005. Twentieth Annual IEEE*, pages 57–62, 2005.

[63] B. Miao, R. Zane, and D. Maksimovic. "A modified cross-correlation method for system identification of power converters

with digital control". *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, pages 3728–3733, 2004.

[64] Botao Miao, Regan Zane, and Dragan Maksimović. "Automated Digital Controller Design for Switching Converters". In *Power Electronics Specialists Conference, 2005. PESC '05. IEEE 36th*, pages 2729–2735, 2005. ISBN 0780390334.

[65] Adam Barkley and Enrico Santi. "Improved Online Identification of a DC-DC Converter and Its Control Loop Gain Using Cross-Correlation Methods". *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 24(8):2021–2031, 2009.

[66] Daniel Martin, Enrico Santi, and Adam Barkley. "Wide bandwidth system identification of AC system impedances by applying pertubations to an existing converter". *2011 IEEE Energy Conversion Congress and Exposition*, pages 2549–2556, September 2011.

[67] Jean Sawma, Flavia Khatounian, and Eric Monmasson. "A modified cross-correlation method for the identification of systems with large bandwidth". *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pages 2121–2126, October 2012.

[68] Erich E Sutter. "THE FAST m-TRANSFORM: A FAST COMPUTATION OF CROSS-CORRELATIONS WITH BINARY m-SEQUENCES". *Society for Industrial and Applied Mathematics*, 20(4):686–694, 1991.

[69] B.J. Fino and V. Algazi. "unified matrix treatment of the fast walsh-hadamard transform". *Computers, IEEE Transactions on*, C-25(11):1142–1146, Nov 1976.

[70] John R. Buck Alan V. Oppenheim, Ronald W. Schaferm. *"Discrete-Time Signal Processing"*. Second Edition, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1999.

[71] R.W. Erikson and D. Maksimovic. *"Fundamentals of Power Electronics"*. Second Edition, Kluwer academic Publishers, 2000.

[72] Hugh Durrant-Whyte. "introduction to estimation and the kalman filter", 2006.

[73] Eric Monmasson, Lahoucine Idkhajine, Marcian N Cirstea, Imene Bahri, Alin Tisan, and Mohamed Wissem Naouar. "FPGAs in Industrial Control Applications". *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 7(2):224–243, 2011.

[74] E. Monmasson and M.N. Cirstea. "FPGA design methodology for industrial control systems - a review". *Industrial Electronics, IEEE Transactions on*, 54(4):1824–1842, 2007.

[75] Lahoucine Idkhajine. *"Fully FPGA-based Sensorless Control for Synchronous AC Drive using an Extended Kalman Filter"*. PhD thesis, 2010.

[76] Eric Monmasson, Lahoucine Idkhajine, and Mohamed Wissem Naouar. "FPGA-Based Controllers". *IEEE Industrial Electronics Magazine*, (March):14–26, 2011.

[77] Andre´ Dehon Scott Hauck. *"Reconfigurable Computing, the theory and practice of FPGA-Based computing"*. Elseviere and Morgan Kaufmann Publishers, 2008.

[78] K. Eshraghian. "soc emerging technologies". *Proceedings of the IEEE*, 94(6):1197–1213, June 2006.

[79] Imen Bahri, Lahoucine Idkhajine, Eric Monmasson, and Mohamed El Amine Benkhelifa. "Hardware/Software Codesign Guidelines for System on Chip FPGA-Based Sensorless AC Drive Applications". *IEEE Transactions on Industrial Informatics*, 9(4):2165–2176, November 2013.

[80] E Monmasson, L Idkhajine, I Bahri, M-w Naouar, and L Charaabi. "Design methodology and FPGA-based controllers for Power Electronics and drive applications". In *5th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2010*, pages 2328–2338, 2010.

[81] E. Koutroulis, K. Kalaitzakis, and N.C. Voulgaris. "development of a microcontroller-based, photovoltaic maximum power point tracking control system". *Power Electronics, IEEE Transactions on*, 16(1):46–54, Jan 2001.

[82] O. Lopez-Lapena, M.T. Penella, and M. Gasulla. "a closed-loop maximum power point tracker for subwatt photovoltaic panels". *Industrial Electronics, IEEE Transactions on*, 59(3): 1588–1596, March 2012.

[83] A. Mellit, H. Rezzouk, A. Messai, and B. Medjahed. "fpga-based real time implementation of mppt-controller for photovoltaic systems". *Renewable Energy*, 36(5):1652 – 1661, 2011.

[84] Eftichios Koutroulis, Kostas Kalaitzakis, and Vasileios Tzitzilonis. "development of an fpga-based system for real-time simulation of photovoltaic modules". *Microelectronics Journal*, 40(7): 1094 – 1102, 2009. Mixed-Technology Testing Rapid System Prototyping.

[85] Mohamed-wissem Naouar, Eric Monmasson, Ahmad Ammar Naassani, Ilhem Slama-belkhodja, and Nicolas Patin. "FPGA-Based Current Controllers for AC Machine Drives - A Review". *IEEE Transactions on Industrial Electronics*, 54(4):1907–1925, 2007.

[86] Stephen Trimberger, J.A. Rowson, C. Lang, and J.P. Gray. "a structured design methodology and associated software tools". *Circuits and Systems, IEEE Transactions on*, 28(7):618–634, Jul 1981.

[87] Mohamed Wissem Naouar. *"Commande numérique à base de composants FPGA dúne machine synchrone"*. PhD thesis, 2007.

[88] F. Ricci and Hoang Le-Huy. "an fpga-based rapid prototyping platform for variable-speed drives". In *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, volume 2, pages 1156–1161 vol.2, Nov 2002.

[89] T. Grandpierre, C. Lavarenne, and Y. Sorel. "optimized rapid prototyping for real-time embedded heterogeneous multiprocessors". In *Hardware/Software Codesign, 1999. (CODES '99) Proceedings of the Seventh International Workshop on*, pages 74–78, 1999.

[90] Aung Myaing and V. Dinavahi. "fpga-based real-time emulation of power electronic systems with detailed representation of device characteristics". *Industrial Electronics, IEEE Transactions on*, 58(1):358–368, Jan 2011. ISSN 0278-0046. doi: 10.1109/TIE.2010.2044738.

[91] Bin Lu, Xin Wu, H. Figueroa, and A. Monti. "a low-cost real-time hardware-in-the-loop testing approach of power electronics controls". *Industrial Electronics, IEEE Transactions on*, 54 (2):919–931, April 2007. ISSN 0278-0046. doi: 10.1109/TIE.2007.892253.

[92] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli. "optimization of perturb and observe maximum power point tracking method". *IEEE Trans. Power Electron.*, 20(4):963–973, 2005.

[93] Yanxia Gao, Shuibao Guo, Yanping Xu, Shi Xuefang Lin, and B. Allard. "fpga-based dpwm for digitally controlled high-frequency dc-dc smps". In *Power Electronics Systems and Applications, 2009. PESA 2009. 3rd International Conference on*, pages 1–7, May 2009.

[94] S.C. Huerta, A. de Castro, O. Garcia, and J.A. Cobos. Fpga based digital pulse width modulator with time resolution under 2 ns. In *Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE*, pages 877–881, Feb 2007.

[95] B.J. Patella, A. Prodic, A. Zirger, and D. Maksimovic. High-frequency digital controller ic for dc/dc converters. In *Applied Power Electronics Conference and Exposition, 2002. APEC 2002. Seventeenth Annual IEEE*, volume 1, pages 374–380 vol.1, 2002.

[96] Z. Lukic, N. Rahman, and A. Prodic. Pwm digital controller ic for dc-dc converters operating at switching frequencies beyond 10 mhz. *Power Electronics, IEEE Transactions on*, 22(5):1693–1707, Sept 2007.

[97] K Harikrishna, T Rama Rao, and Vladimir A Labay. "FPGA Implementation of FFT Algorithm for IEEE 802 . 16e (Mobile WiMAX)". *International Journal of Computer Theory and Engineering*, 3(2):197–203, 2011.

[98] Ahmed Saeed, M Elbably, G Abdelfadeel, and M I Eladawy. "Efficient FPGA implementation of FFT / IFFT Processor". *International Journal of Circuits, Systems and Signal Processing*, 3(3):103–110, 2009.

[99] N Sneha and Meghana Hasamnis. "Efficient Design and Implementation of FFT". *International Journal of Engineering Science and Technology*, pages 10–13, 2011.

[100] Bin Zhou, Yingning Peng, and David Hwang. "Pipeline FFT Architectures Optimized for FPGAs". *International Journal of Reconfigurable Computing*, 2009:1–9, 2009.

[101] S. Mariko. *"Embedded Frequency Response Measurement Capability for Monitoring and Tuning of System Dynamics in Digitally-Controlled DC-DC Converters"*. PhD thesis, University of Colorado, Department of Electrical, Computer and Energy Engineering, 2009.

[102] Panagiotis D. Hatziantoniou and John N. Mourjopoulos. Generalized fractional-octave smoothing of audio and acoustic responses. *J. Audio Eng. Soc*, 48(4):259–280, 2000.

[103] M. Ricco, P. Manganiello, G. Petrone, E. Monmasson, and G. Spagnuolo. Fpga-based implementation of an adaptive p amp;o mppt controller for pv applications. In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, pages 1876–1881, June 2014.

[104] Kendall E. Atkinson. *"An Introduction to Numerical Analysis"*. Second Edition, John Wiley & Sons, 1989.