Waveform Relaxation Based Hardware-in-the-Loop Simulation

By

Mohammad (Monty) Goulkhah

A thesis

Submitted to the Faculty of Graduate Studies in partial fulfilment of

the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

Faculty of Engineering

University of Manitoba

Winnipeg, Manitoba

Abstract

This thesis introduces an alternative potentially low cost solution for hardware-in-the-loop (HIL) simulation based on the waveform relaxation (WR) method. The WR technique is extended so that, without the need for a real-time simulator, the behaviour of an actual piece of physical hardware can nevertheless be tested as though it were connected to a large external electrical network. This is achieved by simulating the external network on an off-line electromagnetic transients (EMT) simulation program, and utilizing iterative exchange of waveforms between the simulation and the hardware by means of a specialized Real-Time Player/Recorder (RTPR) interface device. The approach is referred to as waveform relaxation based hardware-in-the-loop (WR-HIL) simulation.

To make the method possible, the thesis introduces several new innovations for stabilizing and accelerating the WR-HIL algorithm. It is shown that the classical WR shows poor or no convergence when at least one of the subsystems is an actual device. The noise and analog-digital converters' quantization errors and other hardware disturbances can affect the waveforms and cause the WR to diverge. Therefore, the application of the WR method in performing HIL simulation is not straightforward and the classical WR need to be modified accordingly.

Three convergence techniques are proposed to improve the WR-HIL simulation convergence. Each technique is evaluated by an experimental example. The stability of the

WR-HIL simulation is studied and a stabilization technique is proposed to provide sufficient conditions for the simulation stability.

The approach is also extended to include the optimization of the parameters of power system controllers located in geographically distant places. The WR-HIL simulation technique is presented with several examples. At the end of the thesis, suggestions for the future work are presented.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Prof. A. Gole, for the continuous support of my PhD work, for his patience, motivation, and immense knowledge. His guidance helped me in accomplishing this research work and writing of this thesis.

I would like to thank the thesis examining committee, Dr. S. Filizadeh, Dr. J. Morrison, and Dr. B. Johnson, for their insightful comments and encouragement to widen my research from various perspectives and help me present a better work.

I thank Dr. A. M. Kulkarni, who introduced the waveform relaxation simulation method to me when he visited our university in 2011. He provided me with a number of references to start studying the waveform relaxation method. We had many useful discussions about this project while he was in Winnipeg and even after he left.

I thank my fellow labmates for the stimulating discussions, for making an ideal environment to research, and for all fun and unforgettable moments they created during my PhD study. I especially thank Dr. Tomas Yebra Vega, who dedicated a lot of time and effort to implement the proposed algorithms as a neat and easy-to-use simulation package.

I would like to thank my wife, Mana Yazdani, who devotedly dedicated a significant amount of time and effort to help me design many of the proposed approaches in this thesis while she was working on her PhD work as well. I could not imagine a more enjoyable PhD work without her love and her great support academically and spiritually.

Dedication

I proudly dedicate this thesis to my wife, Mana Yazdani, my parents, Zahra and Gholamali Goulkhah, my mother in law, Guity Khadjehzadeh, my father in law, Manouchehr Yazdani, and my PhD supervisor, Prof. A. M. Gole.

Contents

List of Tables

List of Figures

List of Copyrighted Material

I. Figure 1-1 from the University of Manitoba dissertation (Xi Lin) [1].

II. Figure 1-3 from the University of Saskatchewan dissertation (A. D. Jalnapurkar) [17].

III. Figures 3-7 and 3-8 from Cigre-Canada conference paper (M. Goulkhah et al) [45].

IV. Figures 5-22 to 5-31 from IEEE conference paper (M. Goulkhah et al) [56].

Chapter 1

Introduction

This thesis extends a traditional iterative method called waveform relaxation (WR) to perform hardware-in-the-loop (HIL) simulations in a new approach called Waveform Relaxation based Hardware-in-the-Loop simulation (WR-HIL).

In this chapter, an introduction to the HIL simulation and the related challenges are presented. Two popular WR methods are introduced and acceleration techniques, which are used to speed up the convergence of the traditional WR methods, are presented. At the end of the chapter, the thesis outline is presented.

1.1 Power Systems Simulation

Power systems are large and complex dynamic systems with thousands of buses, transmission lines, generators, transformers, …, which all work together to supply the necessary power to the world. The nonlinear and complex behavior of a power system makes it difficult to study the transient and steady state response of the network within the context of variety of system disturbances such as faults, lightning strikes, switching,

equipment failures, etc. Additionally, the time frames of important behaviors of the power systems range from microseconds to hours. For example, a voltage collapse may involve induction machine stalling, which occurs within a few seconds; or a transformer tap changer, whose time interval is tens of seconds; or an automatic generation control (AGC) actions, whose time frame is minutes; and many other actions with time frames ranging from seconds to tens of minutes. Power system dynamics are complicated and precise classification of the time frames in which various actions take place is difficult. Figure 1-1 shows a rough classification of these phenomena [1].

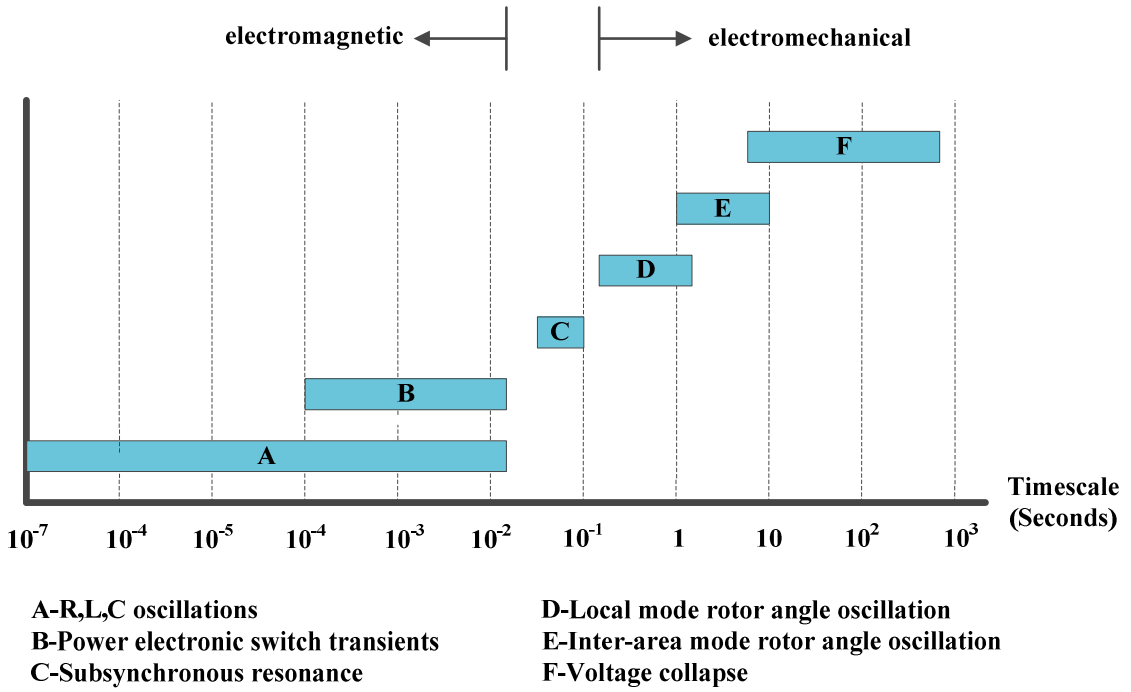


Figure 1-1. Time frames of different power system transient phenomena [1].

Load-flow study tools are used to analyze the steady state operation of power systems. These tools do not use detailed models of power system components and therefore cannot study the transient behaviour of power grids.

Digital time-domain simulation tools have been extensively used in power system studies to accurately study power system transients. In digital time-domain simulations, nonlinear differential equations, representing the model of the power system, are solved using numerical integration techniques. Two types of digital time-domain simulation tools are considered in power system studies: electromechanical transient simulation (also known as transient stability analysis (TSA)) and electromagnetic transient simulation (EMT). Different power system simulation tools are briefly described in the following section.

1.1.1 Load flow simulation

A load- or power-flow study is a numerical calculation method for the flow of power from generators to loads under typical steady state conditions of power systems. Steady state voltage magnitudes and phase angles for every bus, real power and reactive power in generation and load buses, as well as transmission line flows can be obtained from a load-flow analysis. Load-flow analysis can identify flow congestion on power transmission corridors and can also determine whether voltages in the network are within their permissible thresholds. A load-flow study is valuable in planning future expansion of power systems and determining the best operation of existing power systems [2].

1.1.2 Transient stability simulation

This type of simulation tool is mainly used for predicting whether a large power system can regain a stable equilibrium point after being subjected to large disturbances such

as short circuit faults, disconnection of transmission lines, generators, transformers, controller failures, etc. [1]. In transient stability analysis (TSA), the dynamic behaviors of large-inertia power system components, such as electrical generators, and the dynamic interactions between them, which cause large energy flow variations are studied. In such studies, the major part of the electrical network is represented with a phasor (complex-algebraic) model, with the large-inertia components being represented with time-domain differential equations. TSA analysis provides information about the long term stability of the electro-mechanical network, but is not meant for fast dynamics such as power electronic interactions or transients caused by switching surges or lightning.

1.1.3 Electromagnetic transient simulation (EMT)

EMT simulation is a digital time-domain simulation tool that focuses on emulating the detailed behavior of the components of a power system, such as voltage spikes, current surges, voltage and current waveform distortions, harmonics, etc. [1]. One of the major applications of EMT simulation tools is simulating detailed semiconductor switching transients in power electronic devices. EMTP and PSCAD/EMTDC [3] are well known EMT simulation tools. EMT simulation uses the most detailed representation of the power network, and hence gives the most accurate results. However, it is computationally slower than TSA simulation.

1.1.4 HIL simulation

Hardware-in-the-Loop (HIL) is a closed loop system simulation, where part of the system is a computer model simulated in real-time and the other part is the actual hardware component under test. In this scheme, the real-time simulation of the computer model mimics an actual system to which the hardware component is to be connected in the real-life application. The real-time simulator is generally a real-time implementation of the EMT simulation described in Section 1.1.3.

A. The Power Systems HIL method and applications

Today, the design of power system controllers, protective relays, power electronic converters, electric drives, and aerospace systems are becoming increasingly complex and expensive. The increased complexity and demand for the control systems increase the time and cost of design, implementation, and testing of new systems [4]. Traditionally, off-line simulation of computer models of systems was used to design controllers. The hardware implementation of the designed systems was then verified in field, i.e., by connecting the hardware implementation to an actual system. This, of course, increases not only the design time, but also the risk of damaging the control system itself and the controlled system, which is usually prohibitively expensive. Also, it is impractical, inaccurate, and too expensive and risky to test the system designs under a variety of system conditions, e.g., different fault locations and durations in a power system.

In recent years, HIL simulators have become commercially available as a safe solution to test and verify newly designed controller hardware by interfacing them with the

simulated computer models of power systems. The principal requirement of HIL simulators is to ensure that the system simulation is performed in real-time by means of several powerful processing units. The instantaneous real-time values are exchanged at the interface between the simulation and the hardware. All signals are exchanged within each simulation time-step specified in the real-time (digital) simulator. Appropriate interface algorithms and hardware are required to minimize the delays in data communication between the simulation and hardware systems. Existence of such communication delays may lead to inaccurate and in some cases unstable simulations.

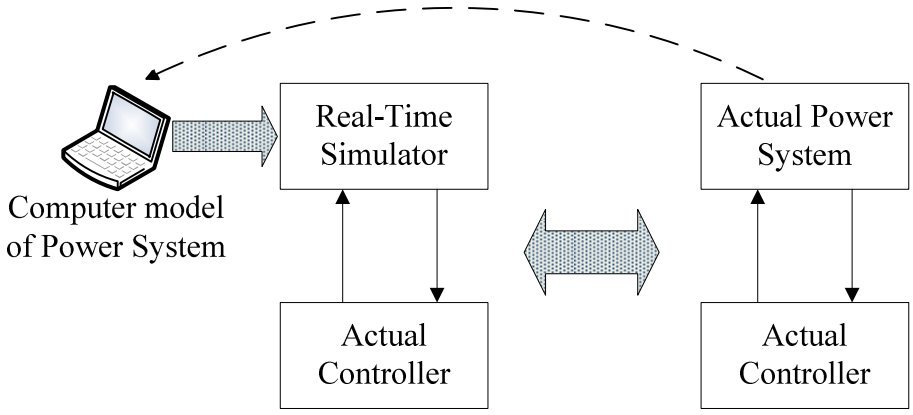


Figure 1-2. HIL simulation configuration.

Figure 1-2 displays a typical HIL simulation configuration, where a computer model of the power system is simulated in real-time by using the real-time simulators and the actual controller is connected to the system simulation for designing, testing, and verification steps. The closed loop results from connecting the controller and the simulated power system and then to the actual power system are identical if the simulated power system is accurate. In cases where the hardware under test (HUT) needs higher voltage or

current levels than what the real-time simulator interface can provide, voltage or current amplifiers are used in the interface between the real-time simulator and the HUT.

B. HIL simulation challenges

High costs: The cost of HIL simulation can be very high. The main portion of the high costs is the expensive and often customized powerful processing units needed to simulate the system models in real-time.

Power system simulation size: Modern power systems often cover large geographic areas and can include components such as HVDC and FACTS controllers for the simulation for which small time-steps are required. The large size of the simulated power network models coupled with the small simulation time-steps imposes significant challenges for performing real-time simulations [4] as it requires fast parallel processing computers. This problem is partially addressed by advances in development of fast processors such as in digital signal processing (DSP) and FPGA technologies [5]. Currently, synchronized parallel processors are used for real-time simulation of large scale power systems that include power electronic components [5, 6, 7]. For instance, FPGAs with multiple processor boards are used in [7] for the real-time simulation of power electronic circuits.

Interface: When real-time digital simulators are used in HIL simulation, the simulation software and the real-world hardware must exchange data in every time-step. Particularly for small time-steps, this requires sophisticated and nontrivial interface designs between the real and simulated worlds. The challenge is that when a real-time simulator is interfaced to external hardware, the interface can often result in inaccurate simulation and even in instability [8-16] due to issues such as amplifier bandwidth and delays. There is

no unique interface algorithm that gives the best stable and accurate results for every simulated system. Instead, depending on the system to be simulated, an appropriate interface algorithm must be selected [8]. As a consequence of simulation instability caused by communication delays (either through wire or internet), HIL simulation with a real-time simulator is infeasible for the cases where the hardware under test is geographically remote from the real-time simulator, such as in another city or even another country.

1.2 Introduction to Waveform Relaxation

In classical definition, relaxation techniques are iterative numerical techniques to solve sets of linear or nonlinear algebraic, differential, and algebraic-differential equations. Relaxation techniques are usually classified as Linear, Nonlinear, and Waveform [17]. This classification is based on the level of the solution in which the relaxation technique is applied.

Consider a system of nonlinear differential equations. In the following discussion, direct methods as well as different ways of applying relaxation methods are separately described in solving the system of nonlinear differential equations [17].

Direct methods: An integration formula, such as Backward Euler or Trapezoidal Rule, is used to discretize the time derivative terms in the system of nonlinear equations. The application of integration formulas results in a set of nonlinear algebraic equations. These equations are then solved using a Newton-Raphson algorithm. Application of the Newton-Raphson algorithm results in a set of linear algebraic equations which then can be solved using any direct method, such as Gaussian elimination. Direct methods such as

the Newton-Raphson can solve a set of equations simultaneously. Depending on the size and complexity of the circuit, the cost of solution by such direct methods can be high. Relaxation techniques are used in order to decouple a set of equations and solve one variable at a time as described below.

Relaxation method: The relaxation technique can be applied at different stages of the solution described above as shown in Figure 1-3, i.e.; before application of the integration formula (labeled as Waveform Relaxation); after application of the integration formula and before applying the Newton-Raphson algorithm (labeled as Nonlinear Relaxation); or ultimately, after both stages of application of the integration formula and the Newton-Raphson algorithm (labeled as Linear Relaxation). All these types of relaxation techniques share a common feature, which is decoupling the resulting set of equations to solve only one variable at a time.

Comparison between Nonlinear Relaxation and direct Newton-Raphson algorithm: The Newton-Raphson algorithm is faster in convergence than Nonlinear Relaxation in solving a set of nonlinear algebraic equations [17]. However, the direct Newton-Raphson method solves a set of coupled simultaneous equations while the Nonlinear Relaxation method involves a set of decoupled equations. Thus, less memory is needed to solve the equations using the relaxation technique. Additionally, the inherent parallelism feature of the relaxation technique allows parallel simulation of the resulting decoupled equations.

Comparison between Linear and Nonlinear Relaxation techniques: This comparison can be made taking into account the size of the resulting equations which are solved by the Newton-Raphson algorithm. In Linear Relaxation technique, the Newton-Raphson algorithm needs to compute the Jacobian matrix of the resulting nonlinear algebraic equa-

tions which is quite expensive. However, in Nonlinear Relaxation technique, the Newton-Raphson algorithm only needs the partial derivative of the resulting nonlinear algebraic function with respect to a single variable, resulting in a considerable saving of the computer time per iteration [17].

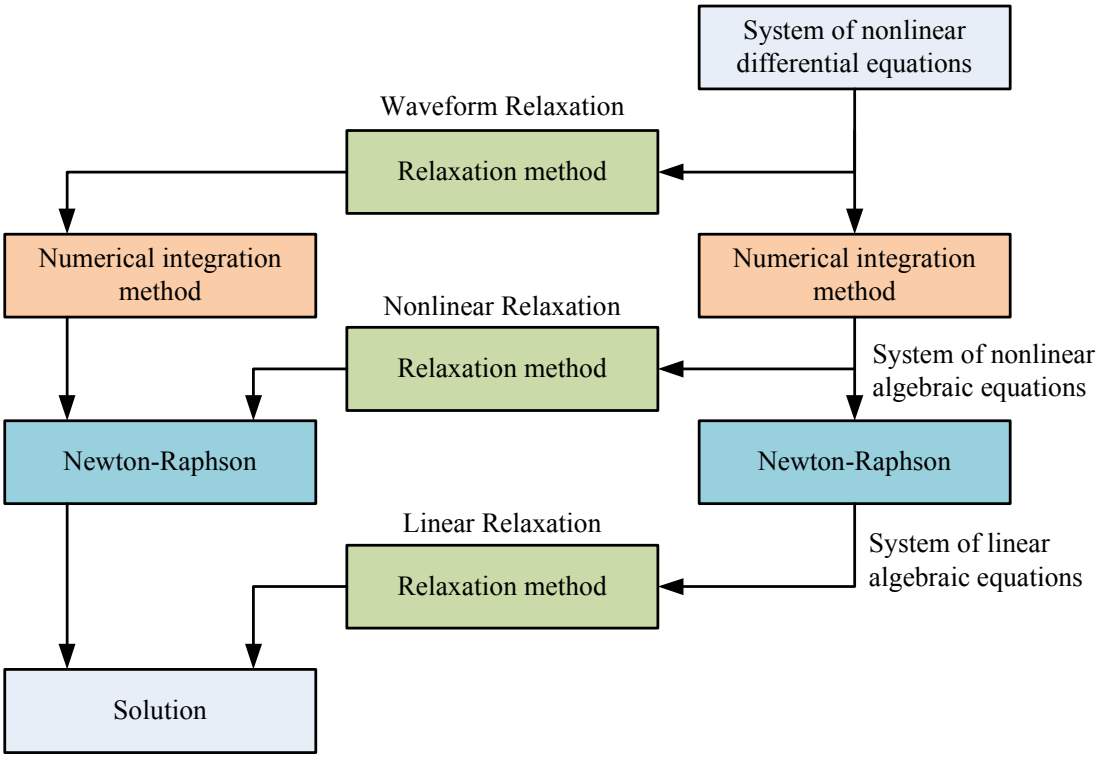


Figure 1-3. Application of Relaxation techniques at different stages of solving a system of nonlinear differential equations [17].

Waveform Relaxation*:* If relaxation methods are applied to solve differential equations, they are called *Waveform Relaxation* methods. Different algorithms are classified by how the relaxation techniques are applied. *Gauss-Jacobi WR and Gauss-Seidel WR* are two of the common methods.

The following provides a brief classification of different types of equation sets that can be solved by WR methods. Such sets of equations, for example, may represent an electric circuit's mathematical model. In the following, it is assumed that [0, T] is the interval of interest in the simulation study.

A) Linear algebraic equations: In (1.1), **A** is an $m \times m$ matrix, **x** is a vector of m variables, **b** is the vector representing independent sources, and T is the total simulation time.

$$\mathbf{A}\mathbf{x}(t) = \mathbf{b} \quad , \quad t \in [0, T] \tag{1.1}$$

B) Nonlinear algebraic equations: The equations might appear as a set of nonlinear functions with respect to variables. This could be caused by existence of nonlinear elements in circuits such as variable resistors, power electronics switches, etc. Such systems may be represented as (1.2), where **f** is a vector of nonlinear functions with respect to the variables, and **u** is a vector of the inputs (i.e. independent sources).

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = 0 \quad , \quad t \in [0, T] \tag{1.2}$$

C) Linear ordinary differential equations (ODE):

$$\mathbf{B}\dot{\mathbf{x}}(t) + \mathbf{A}\mathbf{x}(t) = \mathbf{b} \ , \ \mathbf{x}(0) = \mathbf{x}_0 \quad , \quad t \in [0, T] \tag{1.3}$$

D) Nonlinear ordinary differential equations: (**f** is a vector of nonlinear functions).

$$\mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{u}(t)) = 0 \quad , \quad \mathbf{x}(0) = \mathbf{x}_0 \quad , \quad t \in [0, T] \tag{1.4}$$

E) Differential-algebraic Equation (DAE): Later in Chapter 4, two main classes of such problems, *Index-I* and *Index-II*, are discussed and the WR convergence for DAE equations is studied. **f** and **g** in (1.5) are vectors of linear or nonlinear functions, **x** is a vector of differential variables, **z** is a vector of algebraic variables, and **u** is inputs vector.

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(\mathbf{t}), \mathbf{z}(\mathbf{t}), \mathbf{u}(t), t) \ , \ \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{z}(t) = \mathbf{g}(\mathbf{x}(\mathbf{t}), \mathbf{z}(\mathbf{t}), \mathbf{u}(t), t) \ , \ t \in [0, T] \end{cases} \tag{1.5}$$

Specific properties of different classes of equations, as briefly mentioned above, and the WR convergence properties when applied to solve these equations can be found in [17-27]. Also, a brief discussion about the WR convergence properties as applied to each set of above equations is presented in Chapter 4.

In following, the two common WR techniques, Gauss-Seidel and Gauss-Jacobi, are briefly explained.

1.2.1 Gauss-Seidel WR

Gauss-Seidel Waveform Relaxation (GS-WR) method [17] is the sequential iterative solution of subsystems generated by partitioning the original set of equations. Here, this algorithm is explained for the initial value problem of the ODE system of (1.6) as an example, where f_1 and f_2 are linear or nonlinear functions, and x_1 and x_2 are the system variables with initial values X_1 and X_2, respectively.

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1, x_2, t); & x_1(0) = X_1 \\ \frac{dx_2}{dt} = f_2(x_1, x_2, t); & x_2(0) = X_2 \end{cases} \tag{1.6}$$

First, the order of subsystems to be solved is specified. In this example, the first equation of (1.6) is solved first. Then, the total simulation time T as well as the simulation time-step for each subsystem, Δt_1 and Δt_2, are assigned. If the time-steps Δt_1 and Δt_2 are different values, interpolation methods are used to reconstruct the waveforms. As it will be seen later in this section, the algorithm needs to start from initial guess waveforms for the variables. The initial waveform guess for a variable must include the initial condition $x_2(0)=X_2$. For example, consider the initial guess (k=0, where k is the iteration count) for

the waveform $\mathbf{x}_{2w}^{k}$ for variable x_2. It is of the form $\mathbf{x}_{2w}^{k=0} = [x_2^0(0), x_2^0(\Delta t_2), x_2^0(2\Delta t_2), \ldots, x_2^0(N\Delta t_2 = T)]$, where the w subscript implies that $\mathbf{x}_{2w}$ is a waveform within the time interval of [0, T] constructed by sampled values of the variable x_2. The initial guess waveform for the variable x_2 can be, for example, formed as $\mathbf{x}_{2w}^{k=0} = [X_2, 0, 0, \ldots, 0]$. Figure 1-4 describes the Gauss-Seidel WR algorithm for solving the set of differential equations as in (1.6). In Figure 1-4, it is assumed that both equations are solved with the same time-step of $\Delta t_1 = \Delta t_2 = \Delta t$ for simplicity.

The initial guess of the waveform $\mathbf{x}_{2w}$, i.e. $\mathbf{x}^0{}_{2w}$, is fed into the first equation as a known fixed input. Then a numerical integration method is used to calculate the value of x_1 at each simulation time-step Δt_1 taking into account that $x_1(0)=X_1$. The value of x_1 at each simulation time-step Δt_1 is recorded into the memory as the waveform $\mathbf{x}^1{}_{1w}$. When T seconds of simulation ends, the recorded waveform $\mathbf{x}^1{}_{1w}$ is fed back into the second equation as a fixed input signal. Then the numerical integration method solves for x_2 and the value of x_2 is recorded at every simulation time-step, Δt_2 to give $\mathbf{x}^1{}_{2w}$. This process is continued iteratively to convergence, i.e., when both $\mathbf{x}^k{}_{1w}$ and $\mathbf{x}^k{}_{2w}$ do not change significantly from their previous iterates. A convergence criterion, which can be the norm of the mathematical difference between the most recent and the past values of the waveform $\mathbf{x}_{1w}$ is compared with threshold ε as shown in (1.7). If it is smaller than the threshold, the system is assumed to have converged.

$$\left\| \mathbf{x}_{1w}^{k+1} - \mathbf{x}_{1w}^{k} \right\| < \epsilon \tag{1.7}$$

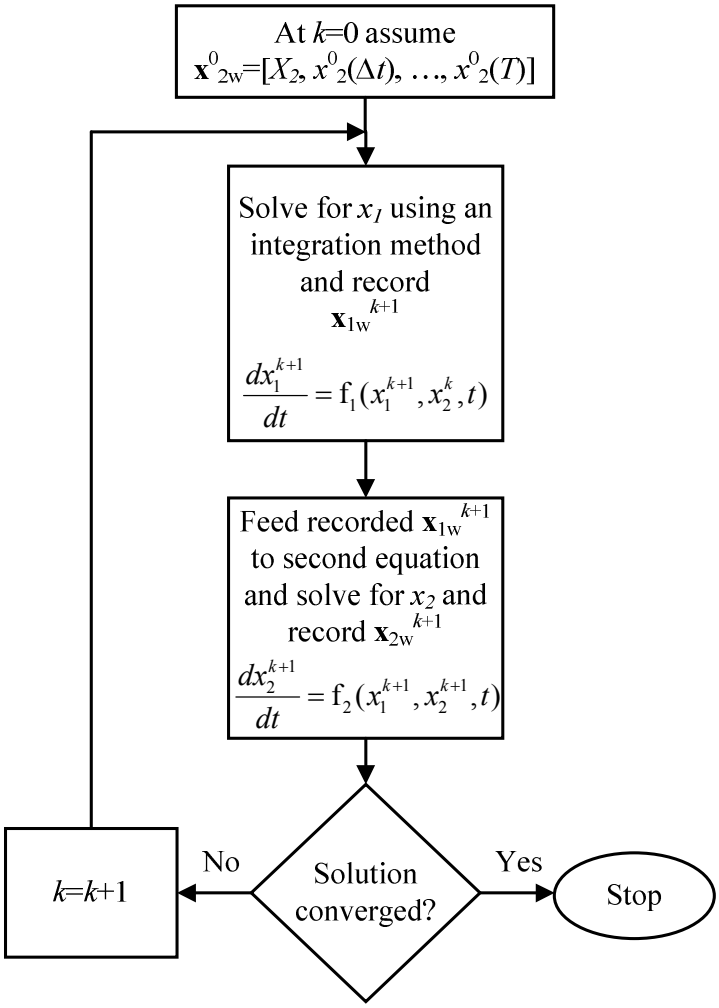


Figure 1-4. GS-WR method for solving a set of two ordinary differential equations.

The waveform norm can be defined for example as (1.8).

$$\left\|\mathbf{x}_{1w}^{k+1} - \mathbf{x}_{1w}^{k}\right\| = \max_{m=1,\ldots,N}\left|\mathbf{x}_{1w}^{k+1}(m\Delta t) - \mathbf{x}_{1w}^{k}(m\Delta t)\right| \quad (1.8)$$

If the convergence criterion in (1.7) is satisfied, the algorithm ends; otherwise, the updated $\mathbf{x}_{2w}$ is fed back into the first equation and the next iteration of the GS-WR is started and the iterations continue until the convergence condition is satisfied.

The ODEs of (1.6), when the GS-WR applied, can be rewritten as (1.9) [17].

$$\begin{cases} \frac{dx_1^{k+1}}{dt} = \mathrm{f}_1({x_1}^{k+1}, {x_2}^{k}, t); \\ \frac{dx_2^{k+1}}{dt} = \mathrm{f}_2({x_1}^{k+1}, {x_2}^{k+1}, t); \end{cases} \tag{1.9}$$

1.2.2 Gauss-Jacobi WR

In Gauss-Jacobi Waveform Relaxation, GJ-WR, [17] all the subsystems are solved in parallel. This means that, for example, in the WR iteration k+1, the input waveforms to all subsystems are from the simulation results from the iteration k.

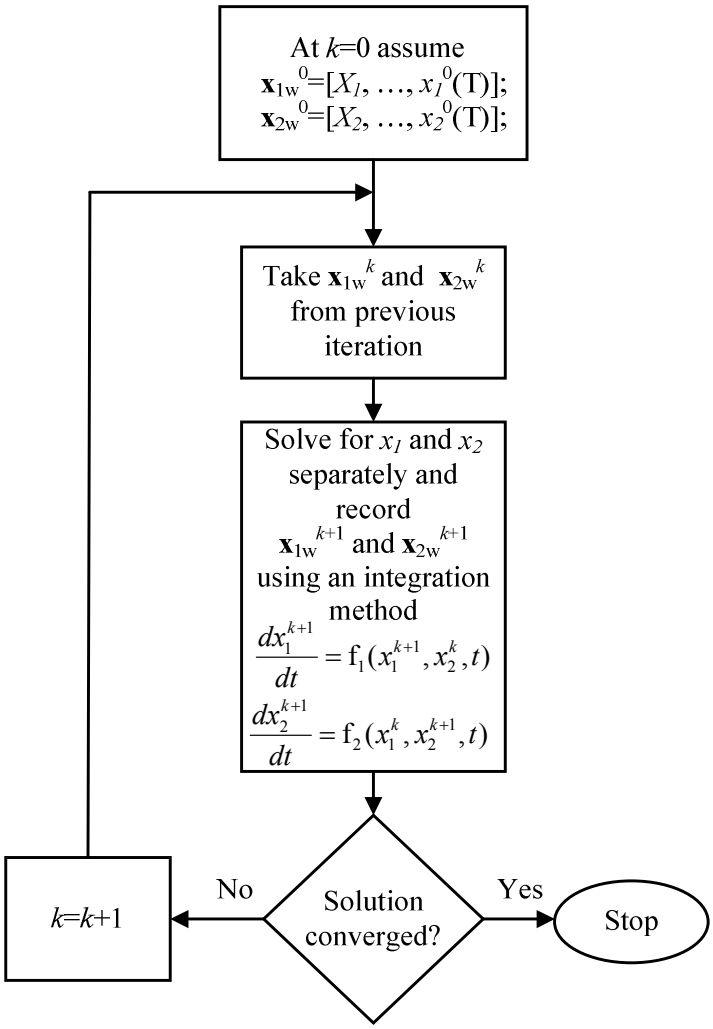


Figure 1-5. GJ-WR method for solving a set of two ordinary differential equations.

This method is popular in parallel simulation of the subsystems as the decoupled subsystems can be distributed on multiple processing units for parallel simulation. This algorithm is presented in Figure 1-5.

The ODEs of (1.6) can be rewritten as (1.10) [17], when the GJ-WR applied.

$$\begin{cases} \frac{dx_1^{k+1}}{dt} = \mathrm{f}_1({x_1}^{k+1}, {x_2}^{k}, t); \\ \frac{dx_2^{k+1}}{dt} = \mathrm{f}_2({x_1}^{k}, {x_2}^{k+1}, t); \end{cases} \tag{1.10}$$

1.2.3 Advantages of Waveform Relaxation

The standard circuit simulators use *standard time-stepping schemes*, i.e. they employ numerical integration methods to discretize the nonlinear DAEs in time. Then, the extracted systems of algebraic equations are solved at each time-step [19]. Usually, calculation of the inverse of the resultant Jacobian matrix of the linearized system is expensive (large memory size and powerful processor units are required) and time consuming especially in cases where the matrix elements change in time (e.g. switching actions in a power electronic circuit). On the other hand, consider the nonlinear DAE problem as shown in (1.5). The rate of change of each variable can be different from the others depending on the structure of the original system. The appropriate simulation time-step is generally determined by the variable whose rate of change in time is the fastest among other variables. By choosing an adequately small time-step for the fastest variables, slow changing variables will also be calculated with the small time-step, which leads to redundant high resolution of the solution for the slow variables.

The WR methods could be applied to overcome some of the issues mentioned above. In this scheme, the circuit model is decomposed into smaller subsystems. In classical applications of the WR method, the partitioning can be done carefully to increase the convergence rate of the simulation. A proper partitioning decomposes weakly connected groups in a circuit, while each group consists of strongly connected elements. A simulation time-step is assigned to each subsystem based on the rate of change of the fastest mode. The generated subsystems are thus solved separately using a suitable numerical integration method for the entire simulation time. One loop of the WR algorithm is accomplished after all the subsystems are simulated for the time interval of interest sequentially (Gauss-Seidel WR), or in parallel (Gauss-Jacobi WR). In cases where the rate of change of signals for different variables is different, the WR method can significantly decrease the time consumed for the simulation of the system in comparison to direct solutions.

The main drawback of the WR algorithms is the high memory usage since the WR waveforms resulting from simulation of all subsystems are recorded to be used in the current or the next WR iterations. Also, a weak partitioning (i.e. partitioning a system within strongly connected points) may dramatically increase the number of iterations required for the WR to converge. It also should be noted that the classical WR methods have lost their popularity for circuits simulation recently because of the advanced technology in manufacturing powerful and cost-effective processor units and cheap memory. In this thesis, the WR method is used to create a new cost-effective HIL simulation method, which is a different approach than its classical use.

1.2.4 Classical WR Acceleration techniques

Without any special speed-up method, GS-WR and GJ-WR can be excessively slow in convergence especially when there are strong feedback loops in the circuits being modeled [6]. Significant research has been conducted to propose new acceleration methods for WR algorithms. A selection of popular acceleration techniques suitable for waveform relaxation based circuits simulation are briefly discussed here.

Partitioning refers to the decomposition of the system into two or more subsystems such that each subsystem shares a common characteristic among its members [28-32]. Multi-rate simulation of the generated subsystems, i.e., simulating subsystems using different time-steps, may accelerate the WR algorithm by solving the slow changing variables with larger time-steps and vice versa. Multi-rate simulation has been used in conventional fixed time-step programs by splitting the larger system into smaller blocks, each running with a different fixed time-step and interfacing them [33].

Time Windowing breaks down the entire simulation time interval into smaller time segments in order to avoid solving the subsystems within sections of time in which the solution is already converged. By applying this technique appropriately, the time spent for the WR convergence can be reduced. However, determination of the size of the time segments whether fixed or variable, statically (before starting the simulation) or dynamically (during the simulation), in a simulation case could be a challenge [25, 28, 34, 35].

Scheduling occurs after partitioning the original system, where the order of solving the subsystems may affect the WR convergence speed. For instance, it is recommended in [28] to solve subsystems with slow changing variables first and then those with fast

changing rates, especially when the windowing technique is applied. For instance, in a current time window, the slow subsystem is first analyzed based on the response of the fast subsystem from the previous time window. Then, the results obtained from the slow subsystem are applied to the fast subsystem in the current time window. The response of the fast subsystem is once again used in the slow subsystem to refine the results acquired in the previous iteration. Specifically, for short time windows, for many applications, these results are accurate enough and no more iterations would be necessary [28].

With *Time-step Acceleration* technique, the subsystems are initially solved with a large simulation time-step for a sufficiently large number of iterations, i.e. until a convergence criterion is satisfied, where that convergence criterion needs not be too small. The resulting low-resolution waveforms are used as inputs for further simulation of the subsystems with the precise time-steps. In this way, only a few iterations may be required to achieve the high-resolution simulation results after changing the time-steps from large to small values [36, 37]. This method can reduce the WR convergence time because the simulation of the subsystems with larger time-steps in the earlier iterations is fast.

Successive Overrelaxation (SOR) manipulates the resulted output waveform from the simulation of a subsystem which will be then fed back as input into another subsystem. This is done by adding a multiple of the "error waveform" to the waveform from the previous iteration [19, 38, 39]. Here, the "error waveform" is the subtraction of the two waveforms recorded from the last two WR iterations for a system variable. For example, once $\mathbf{x}_{1\mathrm{w}}^{k+1}$ is obtained from the simulation of the first equation in (1.9), the manipulated waveform $\mathbf{x}'^{k+1}_{1\mathrm{w}} = \mathbf{x}_{1\mathrm{w}}^{k} + K(\mathbf{x}_{1\mathrm{w}}^{k+1} - \mathbf{x}_{1\mathrm{w}}^{k})$ is provided as the input to the second equation,

where K is a constant. Similarly, $\mathbf{x}'^{k+1}_{2\mathrm{w}}$ is calculated and fed as input to the first equation in the next WR iteration. In order to gain the fastest convergence, some analysis needs to be conducted to optimize the constant K to obtain the fastest rate of WR convergence [19]. In this work, for solving algebraic-differential equations generated by partitioning a circuit, $K \approx 0.9$ resulted in adequate acceleration of the WR methods in most of the cases.

1.3 Thesis Outline

Chapter 2 gives an overview for the simulation of electric circuits using the classical waveform relaxation method. Further in this chapter, the waveform relaxation based hardware-in-the-loop (WR-HIL) simulation as a new application for the waveform relaxation method is introduced.

The WR-HIL method is implemented in hardware by means of a specialized Real-Time Player/Recorder (RTPR) device designed and presented in Chapter 3. The device receives the waveforms generated from the off-line EMT simulation and plays them back in real-time to the actual hardware under test (HUT). The RTPR also samples, digitizes, and records the HUT's output response waveforms and makes them available to the EMT simulation in each WR iteration. Two simple examples of the WR-HIL simulation using the designed RTPR are given. In one example, the WR-HIL simulation technique is used to conduct a Power Hardware-in-the-Loop (PHIL) simulation, where the connection of a 120V transformer to the off-line PSCAD/EMTDC simulation program is simulated. In the second example, the approach is used to test two coordinated instantaneous overcur-

rent protective relays. In this example also, the power system model is simulated in the off-line PSCAD/EMTDC program.

The application of the WR method for performing HIL simulation as described above is not straight-forward. The use of real hardware creates many challenges for convergence of the WR method not reported in its classical, purely software-based use. In classical WR, for a simulated system the partitioning points between subsystems can be carefully selected to ensure fast convergence of the WR method. This luxury is not available in WR-HIL, as the interface point must of course be the point at which the physical device would connect to the remainder of system, making convergence more difficult. Also existence of noise (including quantization errors in analog-digital conversion) can cause the WR to diverge. Chapter 4 discusses such practical and theoretical issues. Furthermore, an analytical study for the stability of the WR based Power Hardware-in-the-Loop (WR-PHIL) simulation is given. It will be shown that the WR may face instability under certain conditions just as in the traditional HIL simulation technique.

In Chapter 5, the convergence problems reported in Chapter 4 are solved by several proposed solutions. Also a stabilization technique is proposed to stabilize the WR-HIL simulation for the cases where the stability criteria found in Chapter 4 are not met.

Chapter 6 extends the WR-HIL simulation approach to enable the optimization of parameters of power system controllers. The proposed algorithm is validated by optimal tuning of two physical controller HIL systems. In these examples, the power system models are simulated in the off-line PSCAD/EMTDC simulation program and the RTPR device is used as the interface to exchange the waveforms between the simulator and the

controllers. The examples include the optimization of the parameters of High Voltage Direct Current (HVDC) and Static Compensator (STATCOM) controllers.

Chapter 7 proposes some potential industrial applications in which the proposed WR-HIL simulation could be applied. In the end of the thesis, conclusions are given and a number of suggestions for the future work are presented.

Chapter 2

Circuit Simulation Using WR

The traditional Waveform Relaxation method has been used in the past to iteratively solve a set of coupled differential equations. This method was later used to speed up the simulation of electric circuits including power systems by partitioning the original circuit into two or more subsystems. In this chapter, this kind of WR application is explained and further the method is extended to perform Hardware-in-the-Loop simulation, which is the principal contribution of this thesis.

2.1 WR Applications in Circuit Simulation

The Waveform Relaxation method has been used for the simulation of electric circuits such as Very Large Scale Integration (VLSI) circuits [25], power systems transient stability studies [34, 35, 40, 41, 42], power electronics simulation [28], and closed loop simulation of wide-area control systems in power networks [43]. These applications of the WR method are briefly discussed in this section.

2.1.1 Application in transient stability

The Waveform Relaxation method has been applied in the past to study the transient stability of typical power systems. In [34], the sets of coherent generators were partitioned into separate equation sets for applying the WR method. In this way, the generators in the power system that are strongly coupled to each other but weakly coupled to the rest of the system are clustered as one group. The fastest mode, i.e., the variable with highest rate of change, in each group determines the largest possible simulation time-step for that group. Power systems are stiff systems, i.e., the difference between the rates of change of the fastest and the slowest modes are significant; and partitioning is the most efficient acceleration technique [34] as was mentioned in Section 1.2.4.

The application of the WR in the simulation of power systems for transient stability studies is also investigated in [35]. It is shown in [35] that the WR methods always converge when applied for the transient stability analysis of the power systems if the coupling between the subsystems (groups of electric machines) is weak. For a typical multimachine power system, the simulation results in [35] show approximately 50% reduction in the WR convergence time when windowing and coherence based partitioning techniques were applied [19].

2.1.2 Application in power electronics

The application of the WR method for fast time-domain simulation of power electronic systems is investigated in [28]. A typical power electronic circuit is first decomposed into slow and fast subsystems. The main energy storage elements, input sources, and con-

trollers in feedback loops are chosen as the elements of the slow subsystem. On the other hand, the fast subsystem consists of power electronic switches, resonant circuits, snubber circuits, and parasitic elements. “Time windowing” is mentioned as the most suitable acceleration technique for the WR based simulation of such power electronic circuits. In [28], the size of the windows is set equal to the switching period of the converter, which could be either fixed or variable.

2.1.3 Extension of WR to perform HIL simulation

Hardware-in-the-Loop (HIL) simulation is a commonly used approach for testing physical hardware and controllers connected to a power system. In this approach, the power system is represented using a real-time simulated model, to which the physical controller is connected. Nevertheless, in some cases, the use of such simulators is either expensive or even infeasible [43]. The latter might be the case when the hardware under the test is located geographically away from the simulation station. System-level testing of protective relays distributed in a large, geographically vast power system and Wide Area Power System Stabilizer (WA-PSS) controllers are such examples [43]. It is suggested in [43] that the classical WR method may be used to tackle the above issue and this thesis follows up on this suggestion. This thesis shows by experimental results that the classical WR methods may at times exhibit non-convergence when one or more of the subsystems are physical hardware. Noise and analog-digital converter errors and other hardware distortions can cause the classical WR methods to diverge in HIL application. Several convergence techniques are proposed and tested in this thesis to modify the classical WR to make it feasible to conduct HIL simulations.

2.2 WR Based Circuit Simulation Example

In this section, a buck converter, a typical power electronic circuit, is simulated. The circuit is split into two subsystems and the Gauss-Seidel WR method, as described in Section 1.2.1, is applied to attain the closed loop response of the two subsystems and the converged result is compared to the direct simulation of the original circuit. Additionally, an acceleration technique (Time Windowing) is applied and the results are presented and discussed.

Figure 2-1 displays a pulse width modulated (PWM) buck converter circuit with the control loop consisting of a Proportional-Integral (PI) controller simulated in Matlab/Simulink. The PI controller regulates the output voltage (V_o) by changing the duty cycle of the switching pulses applied to the series switch (IGBT). In this way, the difference between the output voltage (V_O) and the reference voltage (V_{ref}) is fed back into the PI controller as the error signal and the PI controller outputs the firing angle order for the pulse generator. The input voltage source is 100 V and the reference output voltage is 40 V in the example. The frequency of the saw-tooth waveform used to generate the switching pulses is 10 kHz.

The circuit in Figure 2-1(a) is divided into two subsystems labeled “converter” and “PI controller” as shown in Figure 2-1(b). The simulation time-step for both subsystems is set to $\Delta t = 1\mu S$, due to the high switching frequency of 10 kHz in the PWM converter. The total simulation time is T=0.02 seconds.

The voltage error (x), which is the difference between the output voltage and the reference voltage, and the PI controller’s output (y) are the WR interface variables. The

Gauss-Seidel WR method for solving the two subsystems of Figure 2-1 is presented in Figure 2-2.

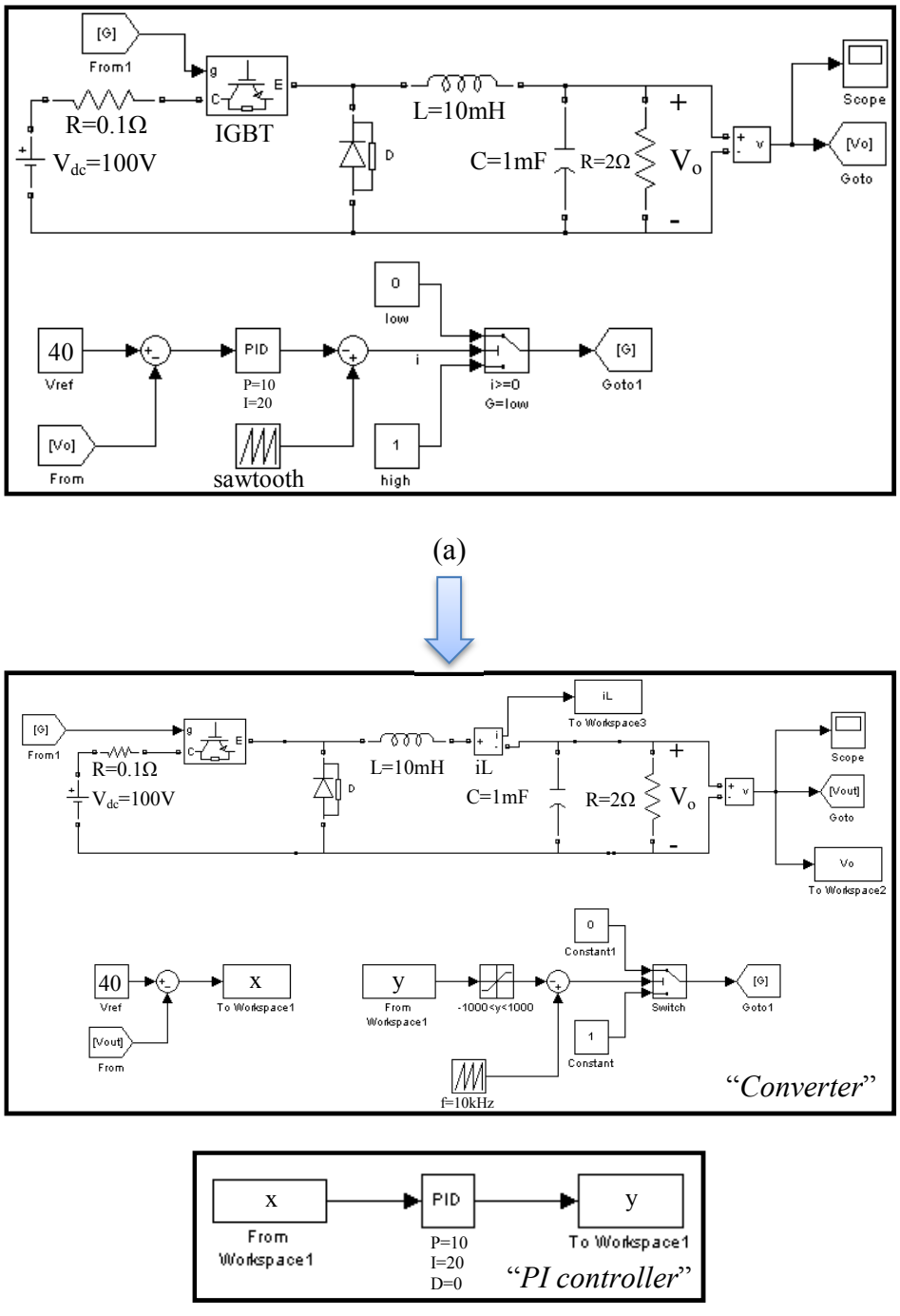


Figure 2-1. (a) A buck converter, (b) partitioning of the system of (a) into two subsystems.

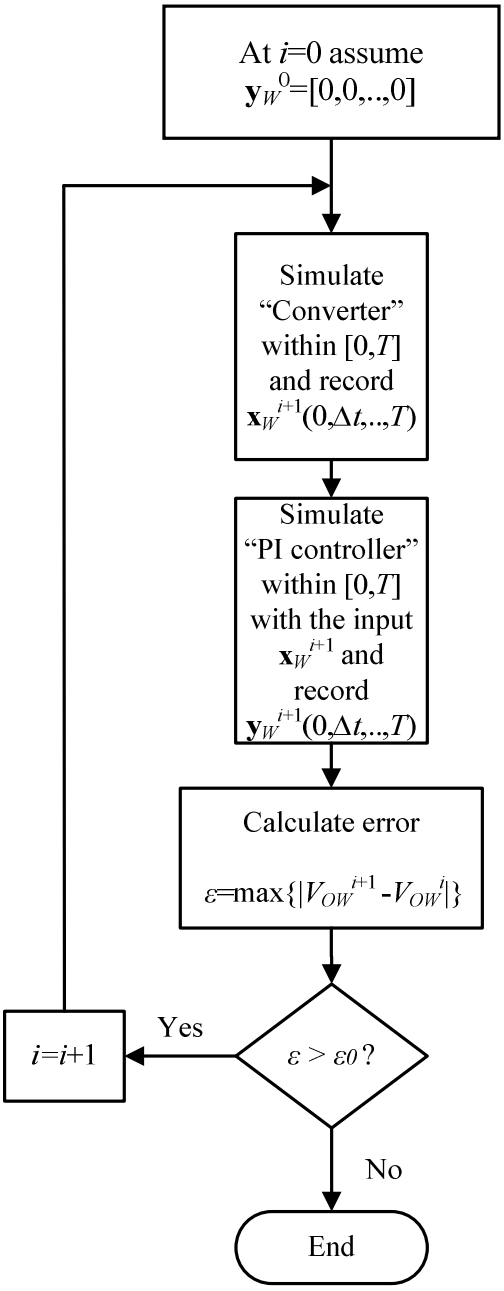


Figure 2-2. Flow chart for WR implementation of Figure 2-1.

An initial guess waveform of $\mathbf{y}_w^0$=[0,0,..,0] is generated for the algorithm. The "converter" subsystem is simulated for T=0.02 seconds and the value of the variable *x* is sampled and recorded at every simulation time-step. This waveform ($\mathbf{x}_w$) is then fed to the "PI controller" subsystem, which is simulated for 0.02 seconds and the output variable *y*

is sampled and recorded at every simulation time-step. The convergence criterion is checked by comparing a convergence threshold (ε_0) with the largest value in the waveform difference between the most recently updated output voltage waveform $\mathbf{V}_{ow}$ and the previous WR iteration, i.e., $\varepsilon_0 = \max\left\{\left|\mathbf{V}_{OW}^{i+1} - \mathbf{V}_{OW}^{i}\right|\right\}$. If the error is less than the threshold value, then the simulation is assumed to have converged; otherwise, the next WR iteration is started.

The GS-WR based simulation results for the output voltage of the circuit of Figure 2-1 are presented for each WR iteration in Figure 2-3. The WR converged in 40 iterations with ε_0=1. The total GS-WR simulation time was 23 seconds with a 2.8 GHz Intel® processor. The converged GS-WR response for the output voltage is compared to the direct simulation results in Figure 2-4. The agreement between the WR simulation result at convergence and the direct simulation result of the circuit shows that the post-convergence WR method results are accurate.

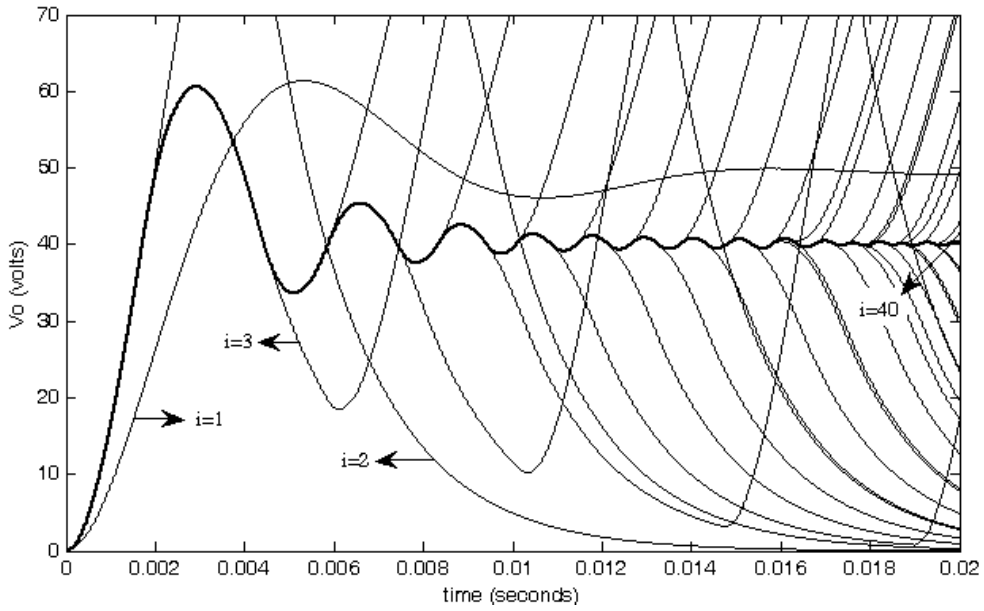


Figure 2-3. Output voltage response in each WR iteration.

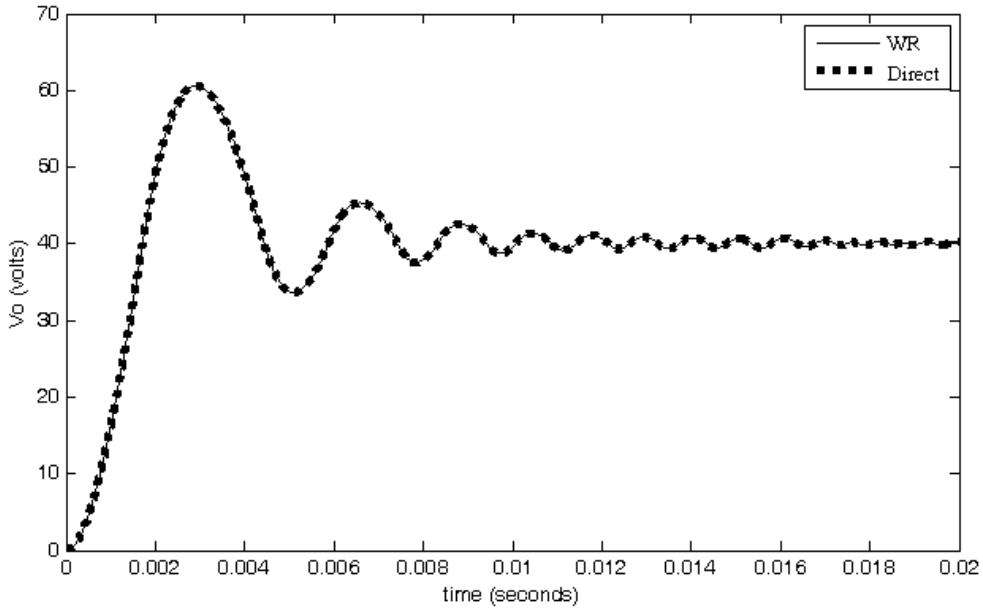


Figure 2-4. Comparison of output voltage responses from WR and direct simulation.

Application of Time Windowing [25, 28, 33, 34]:

WR convergence is piecewise, i.e. as iterations continue, the initial portion of the waveform begins to converge first; and the later portions of the waveform converge in later iterations, without affecting the earlier converged part. This means that the converged parts of the waveforms are repeated in each WR iteration until the rest of the waveform also converges. Hence, the time spent to repeat the converged parts is wasted. In order to prevent such redundant time consumption, time windowing can be applied [25, 28, 33, 34].

Fixed time-window sizes are applied to the example of Figure 2-1 to investigate the effect of this technique on the total number of iterations to achieve the convergence as well as the total simulation elapsed time. The total simulation time T=0.02 seconds is divided into four windows, each 0.005 seconds duration. WR is iterated within the first

time window, and the convergence criterion is checked till satisfied. The converged result for the output voltage waveform within the first window ($\boldsymbol{V}_{OW}[0, \Delta t, 2\Delta t, \ldots, 0.005]$) is recorded. Additionally, the converged values of the inductor's current and the capacitor's voltage at the last time point of the first window, here, $\frac{T}{4} = \frac{0.02}{4} = 0.005$ seconds, are recorded to be used at the next time window as new initial conditions. Ultimately, if the waveform relaxation converges in all four time windows, four pieces of converged output voltages are available. This can be seen in Figure 2-5. Note that here both subsystems are purely digitally simulated models and therefore are re-initializable at the start of each time window, i.e. the initial values for the inductor current and the capacitance voltage can be set to desired values. This luxury is not available in the WR-HIL simulation if the physical hardware is not re-initializable, which is often true.

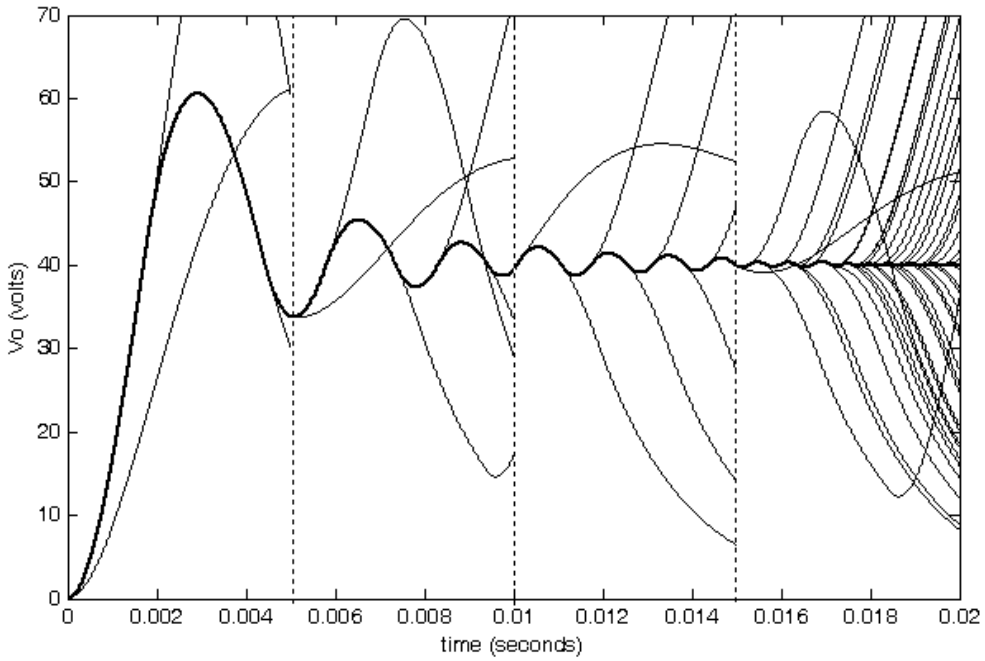


Figure 2-5. Output voltage response for each iteration in case of 4 time windows.

The simulation results show an increase in the total number of iterations. However, there is an overall reduction in the total simulation time. This is because the total simulation time for each window is only $1/4^{th}$ of the time required for an undivided window. The time windowing technique for different number of windows was examined for this example and a summary of results for two cases of four and ten time windows are given in Table 2-1.

Table 2-1. Summary of WR simulation results with time windowing acceleration technique.

Number of Time windows	Number of iterations	Total elapsed Time in seconds
1	40	23.0
4	65	18.1
10	89	19.5

Table 2-1 shows that the total simulation time for 4 windows is decreased by 21.2 %, even though the total number of iterations is increased by 62.5%, as each iteration covers a smaller interval. This might be acceptable in simulation studies, however for the HIL simulation, it is found that the substantial portion of the elapsed time is the communication between the hardware and the simulator computer in exchanging the recorded waveforms. As a result, increased elapsed time of data communication caused by the increased number of iterations is very likely to more than offset the total time saved by the time windowing technique. It is also deduced from Table 2-1 that further increasing of the number of the time windows may lead to less time savings. In this example, application of four windows was more efficient than ten windows. It was observed that, in order to

achieve convergence is a certain time window, many of the iterations took place at the ending portions of the waveforms. This is acceptable because the convergence improves from left to right in the time axis. With 10 time windows, although the duration of the simulation for each window is shorter than that of 4 time windows, the large number of iterations for each time window needed to satisfy the convergence criterion, especially in the ending portions of the waveforms, led to larger total simulation time than 4 windows. This clearly shows that there is always an optimal number of time windows to achieve the fastest convergence.

As mentioned before, it is possible to reinitialize the energy storage elements in simulation models; however, if the actual hardware under test consists of such memories, then it may be impractical to apply the time windowing technique. So in general, time windowing is not suited for WR-HIL simulation.

2.3 WR Based HIL Simulation

2.3.1 WR-HIL method

So far, the classical Waveform Relaxation methods for solving purely simulated computer models were briefly discussed. In this section, the application of WR for conducting HIL simulation is introduced. In this scheme, one of the subsystems in the WR simulation is physical hardware under test (HUT).

Figure 2-5 shows a general block diagram of the actual hardware connection in a real system and two different simulation approaches. Figure 2-5(a) displays the physical

hardware connected to the actual power system. The hardware may be an analog or digital controller. In this case, the simulation is referred to as controller hardware in the loop (CHIL) simulation. It could also represent actual power hardware such as a transformer, generator, or an HVDC converter interfaced to the simulation using high power amplifiers. In this case the simulation is referred to as power hardware in the loop simulation (PHIL).

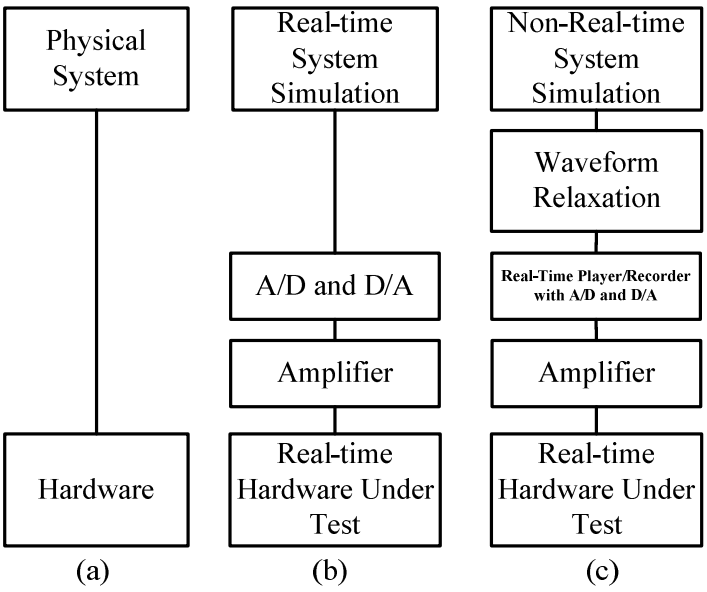


Figure 2-5. (a) Hardware connected to the physical system, (b) testing the hardware using real-time simulators, (c) testing hardware using WR-HIL technique.

Figure 2-5(b) shows the traditional HIL simulation methods where the computer model of the physical system is simulated in real-time by means of a real-time digital simulator. The digital output of the simulator is converted to analog using a D/A converter and amplified if necessary to match the voltage/current levels of the equipment under test. Likewise, the feedback signal from the equipment is sampled and converted to digital for sending to the simulator. Thus, the HUT is connected to the simulation system as

though it is connected to the actual physical system if the model of the system is accurate. In this scheme, the exchange of information between the simulator and the HUT occurs in every time-step.

Figure 2-5(c) shows the classical WR-HIL simulation scheme proposed in [43]. The model of the power system is simulated in an off-line Electromagnetic Transient (EMT) simulation program. The WR method (GS-WR) is used to obtain the closed loop response of the non-real-time system simulation and the real-time HUT. The EMT Simulation results are stored and played back in real-time to the HUT, using D/A converters and amplifiers as needed. The HUT response is sampled and recorded using A/D converters and made available to the EMT for the next iteration. This process is continued to convergence. The method is potentially cost-effective as it permits testing of the HUT without the use of a real-time digital simulator. Also note that unlike the case of Figure 2-5(b), the exchange of information only occurs at the end of the simulation time window and not in each time-step; hence, there is no issue with any time-step delay as would be the case in the traditional HIL simulator. It is declared in [43] that the classical WR fails to converge in many practical applications due to noise and other hardware errors when used in HIL simulation. These issues are further discussed in Chapter 4. In Chapter 5, the classical WR is modified to include three new convergence techniques proposed in this thesis in order to successfully implement for the first time, the WR-HIL technique for practical use. The WR-HIL simulation scheme is described in more detail below. Figure 2-6 shows a device connected to a power system (PS).

The device is a real piece of hardware and PS is simulated on an off-line EMT program. The output waveforms from PS that form the inputs to the device (HUT) are la-

belled "*x*", and the HUT inputs to PS are labelled "*y*". The PS simulation time-step is Δt_1. The HUT may be analog or digital (such as a digital controller or relay), and if it is the latter, its real-time time-step is referred to as Δt_2. Generally, *x* and *y* may be multi-dimensional. A voltage or current amplifier should be used with sufficient power rating if the HUT requires input "*x*" with magnitudes greater than the waveform player's D/A output level.

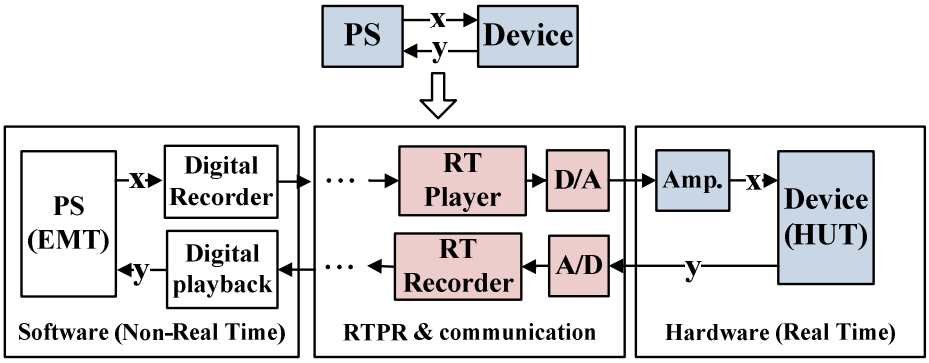


Figure 2-6. Hardware-in-the-Loop simulation using the WR method.

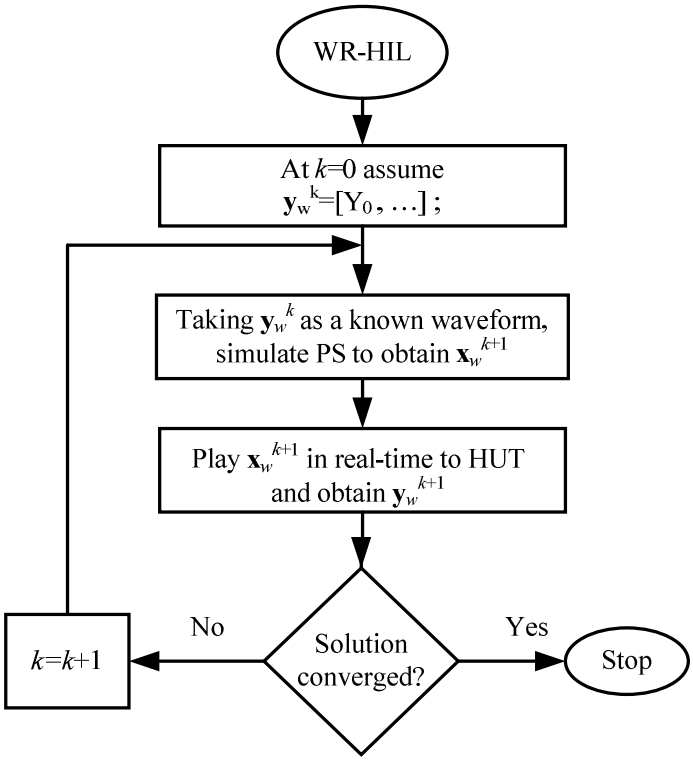


Figure 2-7. Iterative WR-HIL simulation algorithm.

The classical WR algorithm is described in Figure 2-7. The total simulation time is T seconds. An arbitrary initial (digitized) waveform $\mathbf{y}_w^0 = \{y^0(0), y^0(\Delta t_1), y^0(2\Delta t_1)...y^0(T)\}$ is applied to the PS via the "Digital Playback" unit.

In the first WR iteration, the PS is simulated over T seconds and the waveform at the interface is recorded in the "Digital Recorder" as $\mathbf{x}_w^1 = \{x^1(0), x^1(\Delta t_1),...x^1(T)\}$. The RTPR receives and plays $\mathbf{x}_w{}^1$ in real-time as an analog waveform to the HUT. This would be similar to what the device would see in the field, as its input would be measurements in the system. Depending on the available facilities, the RTPR could interface to high power current or voltage amplifiers to give full levels and then measurement transducers (e.g., real current transformers) to make the environment for the HUT as realistic as possible. The waveform is received by the A/D converter of the HUT and sampled with the HUT time-step Δt_2 to yield $\mathbf{x}_w^{1c} = \{x^{1c}(0), x^{1c}(\Delta t_2),...x^{1c}(T)\}$. The RTPR samples and records the HUT's response over the period T to the input $\mathbf{x}^{1c}$ as $\mathbf{y}_w^{1c} = \{y^{1c}(0), y^{1c}(\Delta t_2),...y^{1c}(T)\}$. Again, this is resampled (with time-step Δt_1) as $\mathbf{y}_w^1 = \{y^1(0), y^1(\Delta t_1),...y^1(T)\}$ and applied to the off-line PS simulation as its input for the next iteration. This process is repeated until a convergence criterion, such as (2.1), is satisfied, where δ is the convergence threshold.

$$\underset{t=0,\ \Delta t_2,\ ...,\ T}{\text{Max}} \ (|\ \mathbf{y}_w^{k+1}(t) - \mathbf{y}_w^k(t)\ |) < \delta \tag{2.1}$$

2.3.2 Advantages of WR-HIL method

Cost-effective: In WR-HIL, the simulation and hardware subsystems are decoupled and an ordinary personal computer can be used as the simulation processor regardless of

the size and complexity of the power system, making it a cost-effective alternative to using a real-time digital simulator.

Geographically distributed applications: In classical HIL simulators, any amount of communication delay between the real-time simulator and the HUT introduces some degree of inaccuracy and in some cases instability to the simulation. However WR-HIL is an iterative process where the simulation is off-line and the recorded waveforms are made available to the HUT only after the EMT simulation is finished in every WR iteration. Thus, the waveform timelines can be synchronized with each other as the data is not being exchanged in real-time. Therefore, the communication speed in transferring the recorded waveforms from the EMT simulator to the RTPR (which is connected to the HUT) and vice versa only determines the total time needed for the WR convergence but does not affect the accuracy or stability of the simulation. Consequently, test of single or multiple geographically distributed HUTs is feasible with the WR-HIL approach, which saves time and cost as the HUTs can be tested without having to be moved.

Distributed simulation: The power system model in WR-HIL approach can be partitioned in two or more subsystems in the same or different EMT simulator programs. As a result, the technique allows combination of different simulation tools available in different simulation programs such as PSCAD/EMTDC and Matlab to test the HUT.

Flat cost profile: Power systems are modelled in off-line non-real-time simulation programs. Therefore, the cost of HIL simulation is independent from the size of the power system models. The larger size of the power system models and smaller simulation time-steps only cause longer WR convergence time and unlike real-time digital simulators, there is no requirement to have more processing units. In real-time digital simula-

tors, the size and complexity of the power system models determine the number of the processing units and, therefore, both the simulation cost and the simulator hardware size.

2.4 Concluding Remarks

Waveform Relaxation has been used in the simulation of electric circuits and power systems. In the classical approach, the way the original system is partitioned significantly impacts the WR convergence speed. Time windowing has been successfully used in most of classical WR based circuits simulation to accelerate the WR convergence but is seen here to be impractical for HIL simulation.

The WR method is extended in this thesis for performing the HIL simulation by addition of a real-time player/recorded interface. This new HIL simulation technique is particularly encouraging because it potentially reduces the simulation costs by enabling the simulation of the system model on ordinary personal computers. Also, multiple internet distributed hardware devices can be included in the simulation. Error from latency delays of conventional real-time HIL simulators can be avoided as the exchange of the waveforms take place only after the simulation window.

The following chapter describes the design of the interface hardware for successfully implementing WR-HIL simulation technique.

Chapter 3

Physical Implementation of WR-HIL

The waveform relaxation based hardware-in-the-loop simulation (WR-HIL) procedure described in Section 2.3 requires the playback of simulated waveforms to the hardware under test (HUT) and the collection of HUT response waveforms to the EMT simulator. As no off-the-shelf interface device to achieve this was available, it had to be constructed. This chapter describes the design and implementation of this interface apparatus referred to as the Real-Time Player/Recorder (RTPR) device. Two experimental examples of testing a physical transformer and a protective relay are given with the use of the designed and implemented RTPR device.

3.1 Function of the Real-Time Player/Recorder

The RTPR device records real-time waveforms from the HUT and makes them available to the EMT simulator. It also plays back the EMT simulator waveforms to the HUT in real-time.

Assume the WR-HIL simulation of a physical controller with the power system model simulated in an EMT simulation program. In a WR iteration, the waveforms generated from the off-line EMT simulation of the power system model are recorded in the simulator computer. The RTPR obtains these recorded waveform samples via a secure communication protocol through LAN or WiFi and stores them in local memory. Of course, the number of samples depends on the EMT simulation time interval as well as the simulation time-step. These waveforms may be analog, e.g., measured voltage, current, speed, or digital, e.g., switching pulses. The RTPR plays back every recorded sample of all waveforms to its digital to analog converter (DAC) output channels for analog waveforms and digital output (DO) channels for digital waveforms. An on-board real-time clock is used to synchronise all the input/output channels and measure the time duration of the playback of each sample to be equal to the specified time-step for a simulation case. The HUT's input terminals, here the physical controller, are connected to the RTPR's output channels and the controller's output terminals are connected to the RTPR's input channels. The RTPR samples, digitizes, and records the output response of the controller at every instant that a recorded sample is played back. The RTPR is equipped with analog to digital converters (ADC) and digital input (DI) channels in order to sample the analog and digital signals at the output terminals of the controller. The RTPR records the controller's response samples as "waveforms" in the local memory. After all the recorded waveform samples are played back to the controller and accordingly the response of the controller is sampled and recorded for all time-steps, the RTPR sends the recorded samples to the EMT simulator for the next iteration of the WR. After transmitting the samples and before receiving the new set of waveform samples

from the EMT simulation, the RTPR plays back a specified value to the controller, such as 0 volts. This way, in some cases, the controller could be re-initialized to the same state at the beginning of every waveform playback/record in each WR iteration.

3.2 Designed RTPR Prototype

Figure 3-1 shows a prototype implementing the RTPR functions mentioned above.

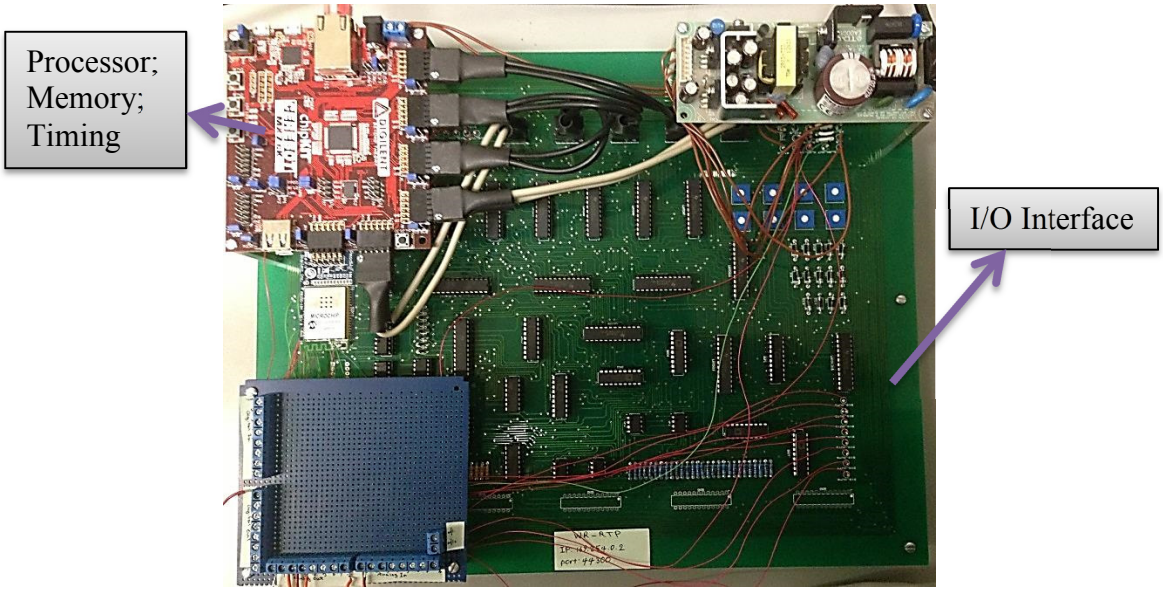


Figure 3-1. Designed RTPR device circuit.

The CEREBOT MX7cK from chipKIT™ [44] is used as the processor for the implementation of the above functions and also for timing. These specifications were designed to permit the range of simulations with power electronic converters targeted in this thesis.

The main design specifications of the RTPR prototype are as follows.

- Data communication: Serial, LAN, WiFi;
- Internet protocol: TCP/IP with 2Mbps speed;
- Authentication: WPA2, WPA-PSK, WEP104, WEP40, Open networks;
- Data ports: 7 analog input, 7 analog output, 8 digital input, 8 digital output

channels;

- Data resolution: 10 bits for ADCs and DACs;
- Sampling/Playback maximum frequency: 20 kHz.
- Data memory (buffer) size: 100 kB.
- Analog output voltage range: ±5 V.

The size of the memory available in the RTPR limits the total simulation time depending on the number of I/O channels used in a simulation as well as the simulation time-step. The EMT simulation time-step and the playback/sampling time-step could be optionally different if needed. In this case, an interpolation method should be used to interface the waveforms with different number of samples in the EMT simulation and the RTPR. For the given RTPR prototype, taking into account that every analog data sample has a 10-bit resolution and, therefore, takes 2 bytes from the memory (for an 8-bit memory here), and every digital sample takes one bit, (3.1) can be used as the formula to calculate the maximum possible simulation time, where Δt is the hardware time-step, "a" is the maximum of the number of input and output analog channels used, and "d" is the maximum of the number of input and output digital channels used. The size of available memory is denoted here by "m" and it is noted that each waveform sample is erased from memory after being played back to the controller and is replaced by the corresponding controller's output sample. As an example, with m=100 kB buffer, Δt=50 μS, 2 analog channels and 4 digital channels used, and with a 16 bit word length,

$$T_{\max} = \frac{(100\times10^{3})\times(50\times10^{-6})}{2\times2+\frac{4}{8}} = 1.11 \text{ Sec.}$$

$$T_{max} = \frac{m\Delta t}{2a+\frac{d}{8}} \tag{3.1}$$

3.3 Server-Based WR-HIL Implementation

An internet-based server is included in the above design to facilitate WR-HIL. This makes it possible to have the HUT and the simulator in different locations; it is only necessary to have the RTPR next to the HUT. Communication of the waveforms between the HUT (via RTPR) and the simulation computer is done over the internet. A GPS-based time stamp is used in the synchronization if needed. This can allow for novel multi-area controller tuning as will be discussed in Chapter 7.

A general scheme of a web server-based WR-HIL approach is shown in Figure 3-2.

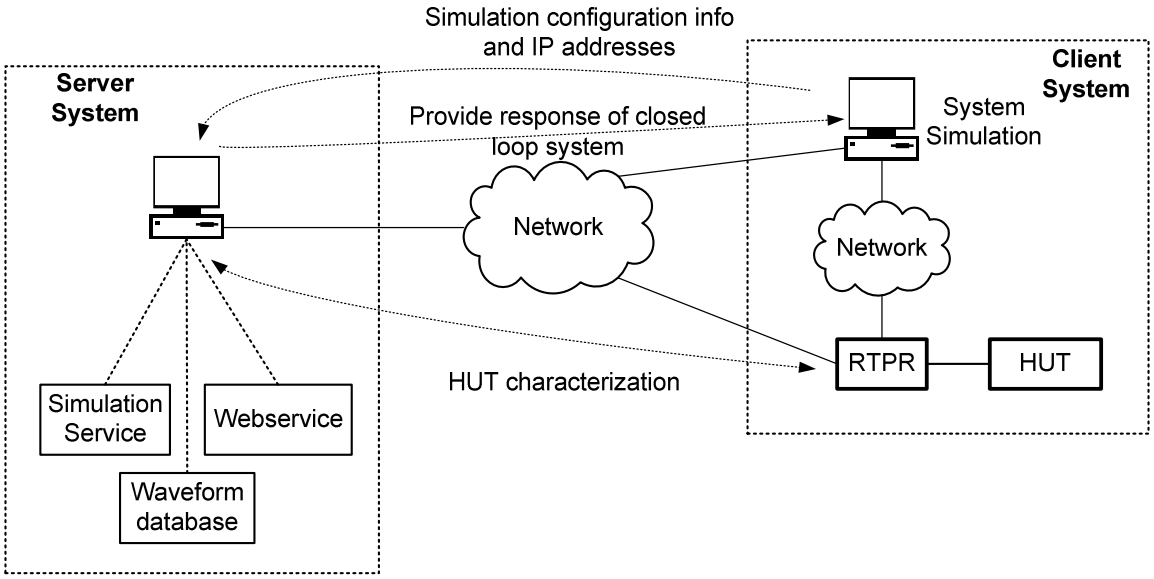


Figure 3-2. Server based WR-HIL simulation.

The major parts of the configuration are:

Server: A server computer in which all algorithms are implemented;

Client: One or more PC computers in which one or more power system models are simulated in an EMT simulation program (such as PSCAD/EMTDC). The simulations are not necessarily in real-time. Each simulation PC is connected to internet with an IP address;

HUT: One or more hardware under test (HUT), either local or remote to server and to each other;

RTPR: One RTPR for each HUT. The RTPR is the interface between non-real-time simulation and real-time HUT. Each RTPR is connected to internet with an IP address;

Network: A network, that performs the communication between all systems.

In this configuration, many HIL simulations can be conducted at the same time with proper design of the server infrastructure.

3.4 WR-HIL Examples

Here, two examples are presented to demonstrate the WR-HIL simulation using the designed RTPR prototype. The first one is a Power Hardware-in-the-Loop (PHIL) simulation example based on the WR approach. In the second example, two coordinated protective relays, implemented in physical hardware, are tested. In both examples, the off-line PSCAD/EMTDC simulation program is used to simulate the power system models and the RTPR device is used as the interface to the HUT.

It should be noted that WR-HIL can be unstable if some conditions are not satisfied. This will be described in detail in Chapter 4.

3.4.1 PHIL simulation example

The WR method is demonstrated using a PHIL simulation example. Here, an unloaded single phase transformer (i.e., the power hardware), is energized from a source through a pi-section representing a transmission line (simulated) as shown in Figure 3-3.

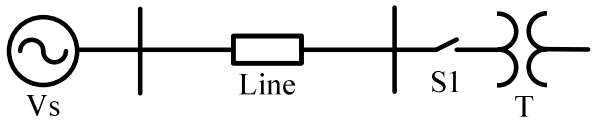


Figure 3-3. Power system configuration.

The PHIL simulation configuration is as in Figure 3-4. The source and line are simulated and the simulation is interfaced to the power hardware via the RTPR and a voltage amplifier.

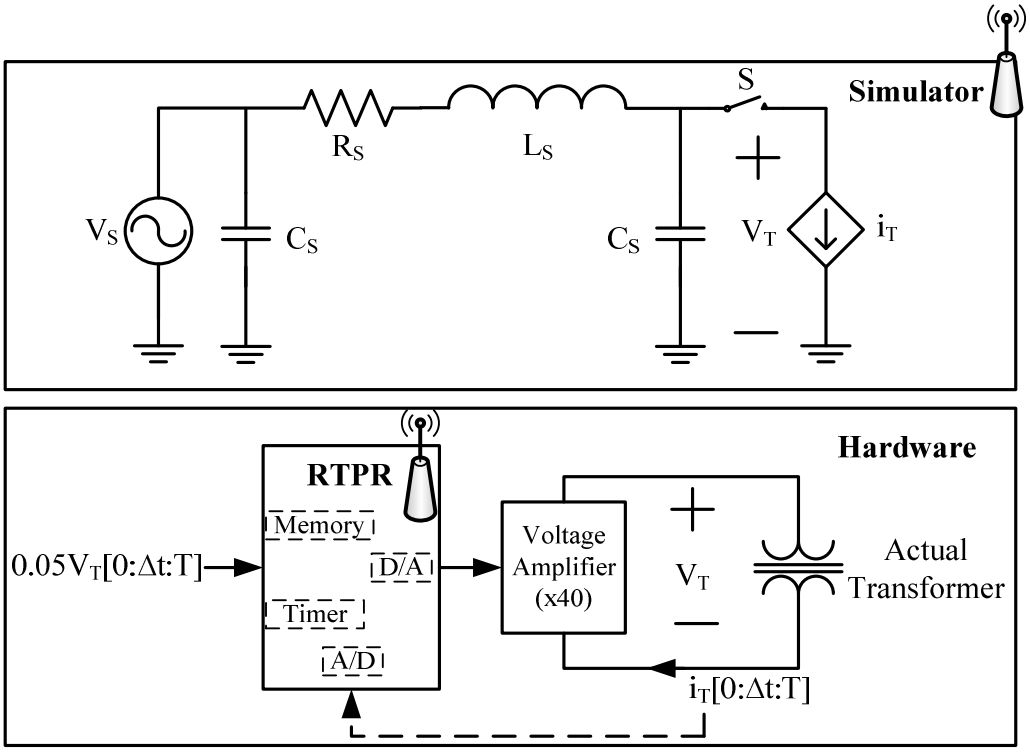


Figure 3-4. Simulation and hardware setup for the WR-PHIL simulation of the actual transformer.

WR-PHIL simulation of the circuit in Figure 3-3 is conducted using an actual single phase transformer rated at 60 VA, 120/120 V. The leakage inductance is 6%. The network side is simulated in PSCAD/EMTDC with the following parameters (Figure 3-4): V_S=120 V, f=60 Hz, R_S=0.5 Ω, C_S=1 μF, L_S=10 mH (e.g., a short line). Note that the length of the transmission line (represented here by L_S) greatly determines the WR-PHIL

simulation stability in this example as will be discussed in Chapters 4 and 5. Here, the given value of L_S=10 mH results in stable WR-PHIL simulation without having to use any stabilization technique.

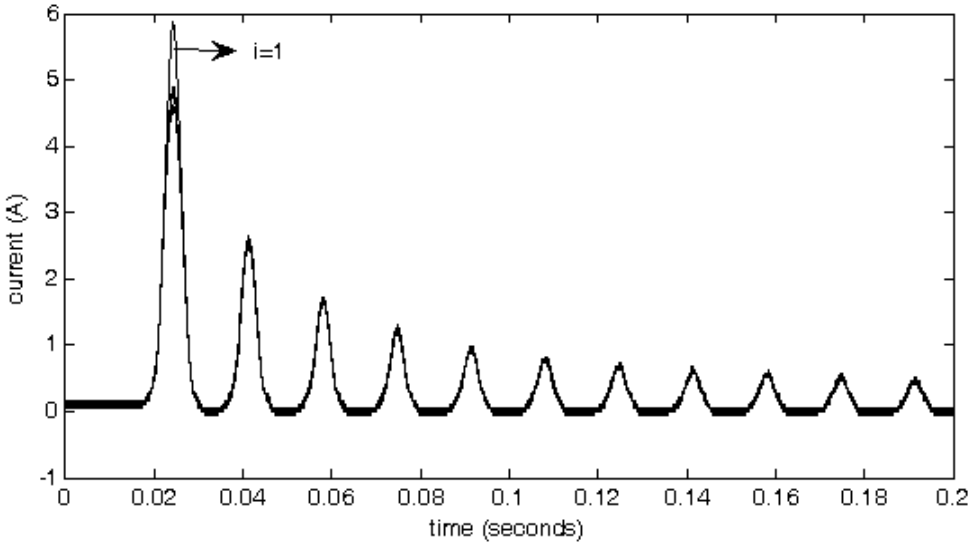


Figure 3-5. WR-HIL response for the transformer's inrush current in each WR iteration.

The interface between the simulation and power hardware sides appears as a current source in the simulation. Its time series is a playback of the recorded current waveform of the actual transformer from the previous WR iteration. Likewise, the simulated voltage across this current source is sent to the RTPR using inbuilt WiFi communication, which converts it to an analog waveform. The maximum peak voltage from the RTPR is 5 V, so an additional scaling by a factor of 40 is implemented with an external voltage amplifier. This waveform is applied to the transformer in the next WR iteration. The recorded current waveform is once again made available to the simulation part in the next WR iteration, and the iterative process repeated till convergence. The switch 'S' is closed at positive zero crossing of the voltage source about 0.02 seconds. For the short transmission

line model, WR converges in 4 iterations. Figure 3-5 shows the experimental result of the inrush current of the transformer in every WR iteration.

Figure 3-6 also plots for comparison, fully PSCAD/EMTDC simulated transformer inrush current waveforms. The comparison shows that the WR-PHIL simulation result agrees with that of the PSCAD/EMTDC direct simulation result of the simulated transformer model, giving confidence in the WR-PHIL simulation approach. Of course, the PSCAD/EMTDC simulation used a model for the transformer, whereas WR-PHIL used the actual transformer.

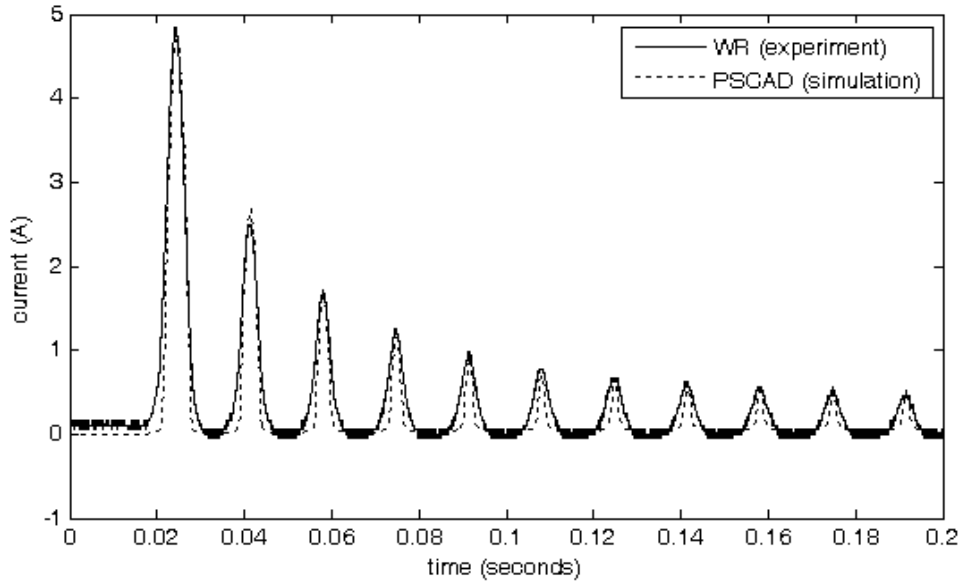


Figure 3-6. WR-HIL converged response for the transformer's inrush current compared to PSCAD/EMTDC transformer direct simulation.

3.4.2 WR-HIL based test of relay example

Figure 3-7 shows the setup for the Waveform Relaxation based closed loop relay hardware test example [45]. The power system is modelled in PSCAD/EMTDC [3]. The

power system model in Figure 3-7 includes a single phase voltage source of V_S=230 kV (rms), 60 Hz. Under rated conditions, there are four parallel resistive loads each with R=700 Ω connected to the system, two of which are switchable by means of two breakers controlled by the open/close signals *BRK1* and *BRK2*. The temporary load, R_2=350 Ω, is connected to the system by the switch S for a short time interval.

The protection scheme is to ensure the transmission line current (I) does not settle at a value more than a marginal current (e.g. I_{margin}=1.2 pu) during the connection of the temporary load to the system. If there is an overcurrent, i.e., $I > I_{margin}$, the breaker controlled by *BRK1* trips a portion of the load (one resistor R) after a specified time (t_{BRK1}=0.05 sec). If the line current is reduced to a value smaller than I_{margin}, then no further action is taken. Otherwise, if the current is still larger than the marginal current after the breaker controlled by the signal *BRK1* trips, then the second breaker, which is controlled by the trip signal *BRK2*, trips after a specified trip time (t_{BRK2}=0.1 sec), disconnecting an additional resistor R to reduce the current even further. Here t_{BRKi} is the time needed for the 'i'th breaker to open after the "first" occurrence of an overcurrent. After the temporary load is disconnected from the system, the breakers reconnect the loads with the same order in which they were disconnected, i.e., first the breaker controlled by the signal *BRK*1, then the breaker controlled by the signal *BRK*2 is closed. This function is implemented in real-time hardware with two inverse-time overcurrent protection relay elements [46] both on the same platform. Rather than starting from scratch, the HIL relay was implemented using the programming facility and PB5 cards used in the RTDS simulator.

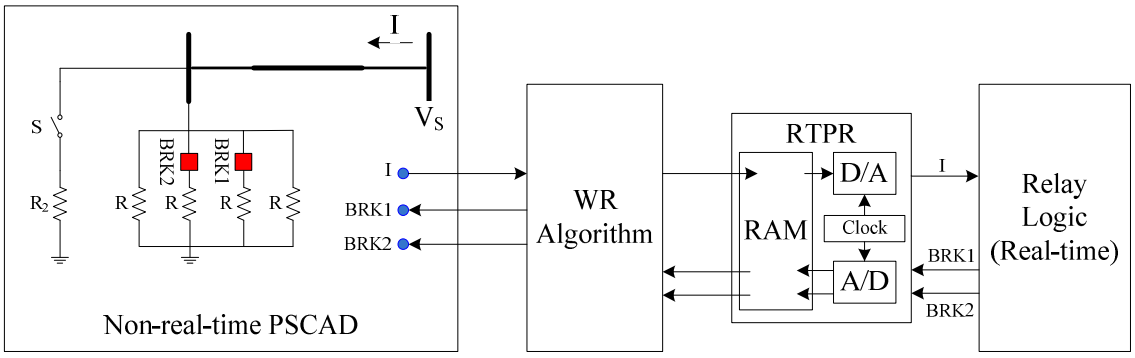


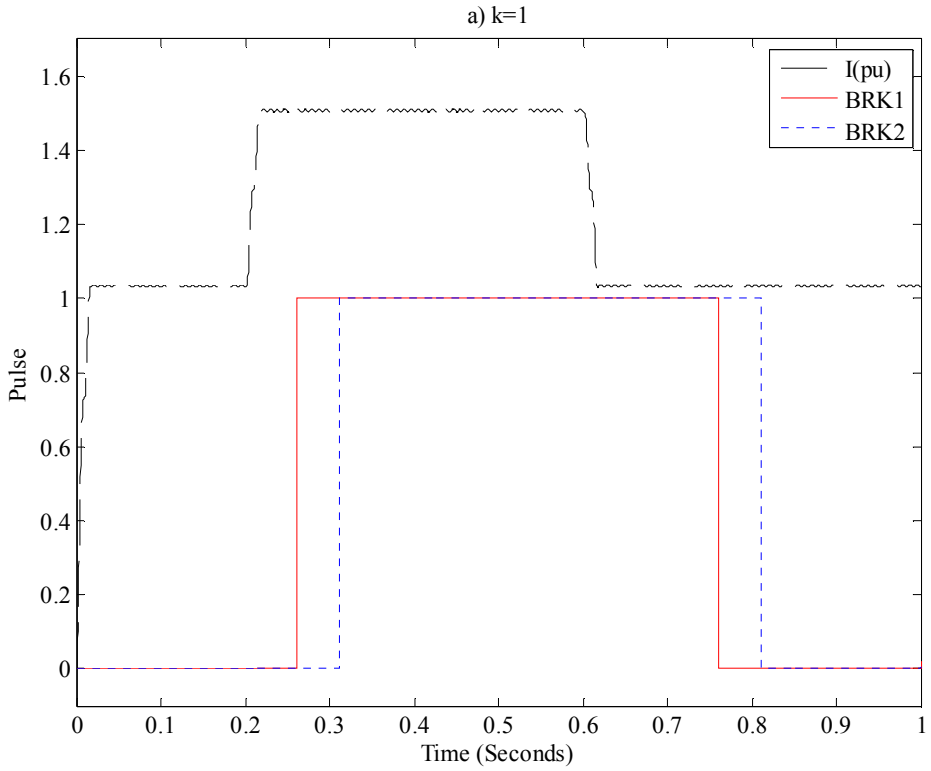
Figure 3-7. Waveform Relaxation based closed loop relay test example [45].

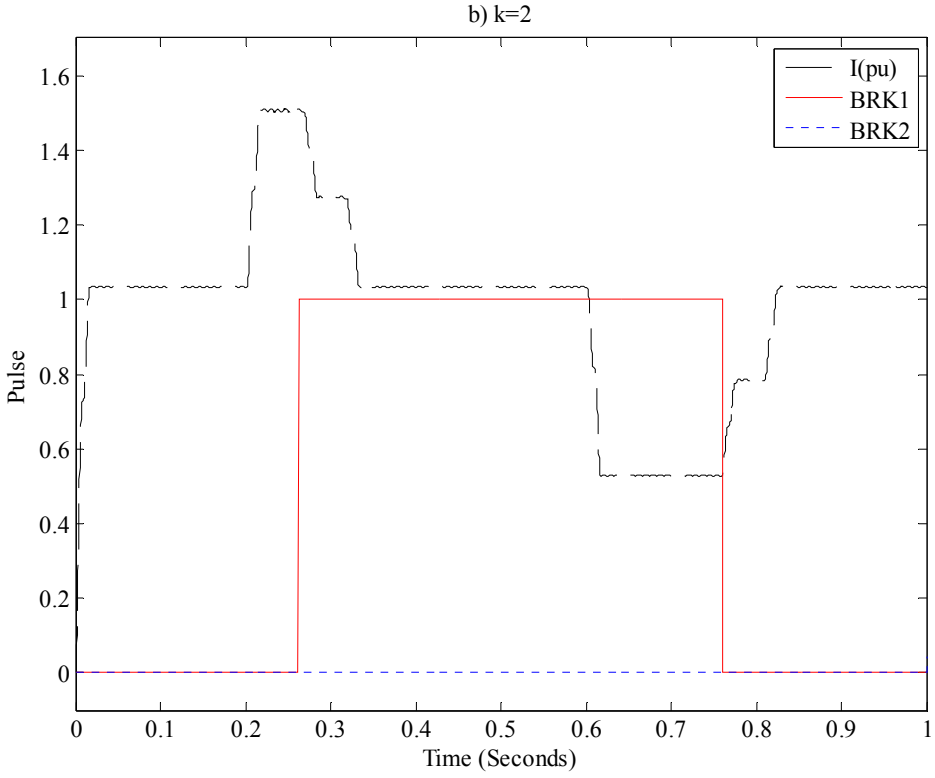
The RTPR receives the waveform of the measured rms value of current (*I*) resulting from the PSCAD/EMTDC simulation and plays it back in real-time to the relay hardware and simultaneously digitizes and records the real-time trip signals, *BRK*1 and *BRK*2, generated by the relay. These trip signals are sent to the WR algorithm to begin the next PSCAD/EMTDC simulation iteration. This iterative procedure continues till convergence.

Figure 3-8 presents the PSCAD/EMTDC simulation generated current in pu, which is also the input signal to the relay hardware, and the relay's response (trip signals), *BRK*1 and *BRK*2, in every WR iteration (*k*). Two waveforms with all zero values as $\mathbf{BRK1}_w^0$ and $\mathbf{BRK2}_w^0$ are generated by the WR algorithm and are fed to the PSCAD/EMTDC simulation in the first iteration (*k*=1). The resulting waveform of $\mathbf{I}_w^1$ is shown in Figure 3-8-a. The temporary load is connected to the system at *t*=0.2 seconds and removed at *t*=0.6 seconds. No trip output has been generated by the relays as $\mathbf{BRK1}_w^0(t)=\mathbf{BRK2}_w^0(t)=0$ for the whole PSCAD/EMTDC simulation time in the first WR iteration. Then, the recorded waveform, $\mathbf{I}_w^1$, is made available to the relay hardware via the RTPR device in real-time and the relay's responses ($\mathbf{BRK1}_w^1$ and $\mathbf{BRK2}_w^1$) are recorded as shown in Figure 3-8-a.

The relay issues the trip signals for both breakers because the relay logic is performed as off-line to the PSCAD/EMTDC simulation and, obviously, the recorded current waveform does not change regardless of the first breaker's trip signal.

The convergence criterion is not met in the first WR iteration because the trip signals in k=0 and k=1 are not identical. Hence, WR starts the second iteration. $\mathbf{BRK1}_w^1$ and $\mathbf{BRK2}_w^1$ are fed to the PSCAD/EMTDC simulation. The simulation result for the line current, $\mathbf{I}_w^2$, is shown in Figure 3-8-b. It can be seen that the PSCAD/EMTDC breakers open accordingly with the relay output trip signals $\mathbf{BRK1}_w^1$ and $\mathbf{BRK2}_w^1$. The recorded line current waveform from the second iteration, $\mathbf{I}_w^2$, is played back to the relay and the relay's response signals, $\mathbf{BRK1}_w^2$ and $\mathbf{BRK2}_w^2$, are sampled and recorded. From Figure 3-8-b, the relay issues a trip signal for the first breaker and removes the trip signal for the second breaker because $\mathbf{I}_w^2$ is less than I_{margin}=1.2 pu after the first breaker is opened. The updated relay's response waveforms are once again fed to the PSCAD/EMTDC simulation for the third WR iteration. The simulation result, $\mathbf{I}_w^3$, is shown in Figure 3-8-c. Accordingly, the first breaker opens and the second breaker remains closed. $\mathbf{I}_w^3$ is applied to the relay hardware and the relay response waveforms, $\mathbf{BRK1}_w^3$ and $\mathbf{BRK2}_w^3$, are shown in Figure 3-8-c. It can be found, from comparing the Figure 3-8-b and Figure 3-8-c, that the relay response waveforms are identical in the second and third iterations, i.e. $\max\{|\mathbf{BRKi}_w^3 - \mathbf{BRKi}_w^2|\} < \varepsilon$=0.001; from which can be said that the WR has converged in iteration number 3, i.e. (k=3).


a) k=1
I(pu)
BRK1
BRK2
Pulse
Time (Seconds)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0.1
0.3
0.5
0.7
0.9


b) k=2
I(pu)
BRK1
BRK2
Pulse
Time (Seconds)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0.1
0.3
0.5
0.7
0.9

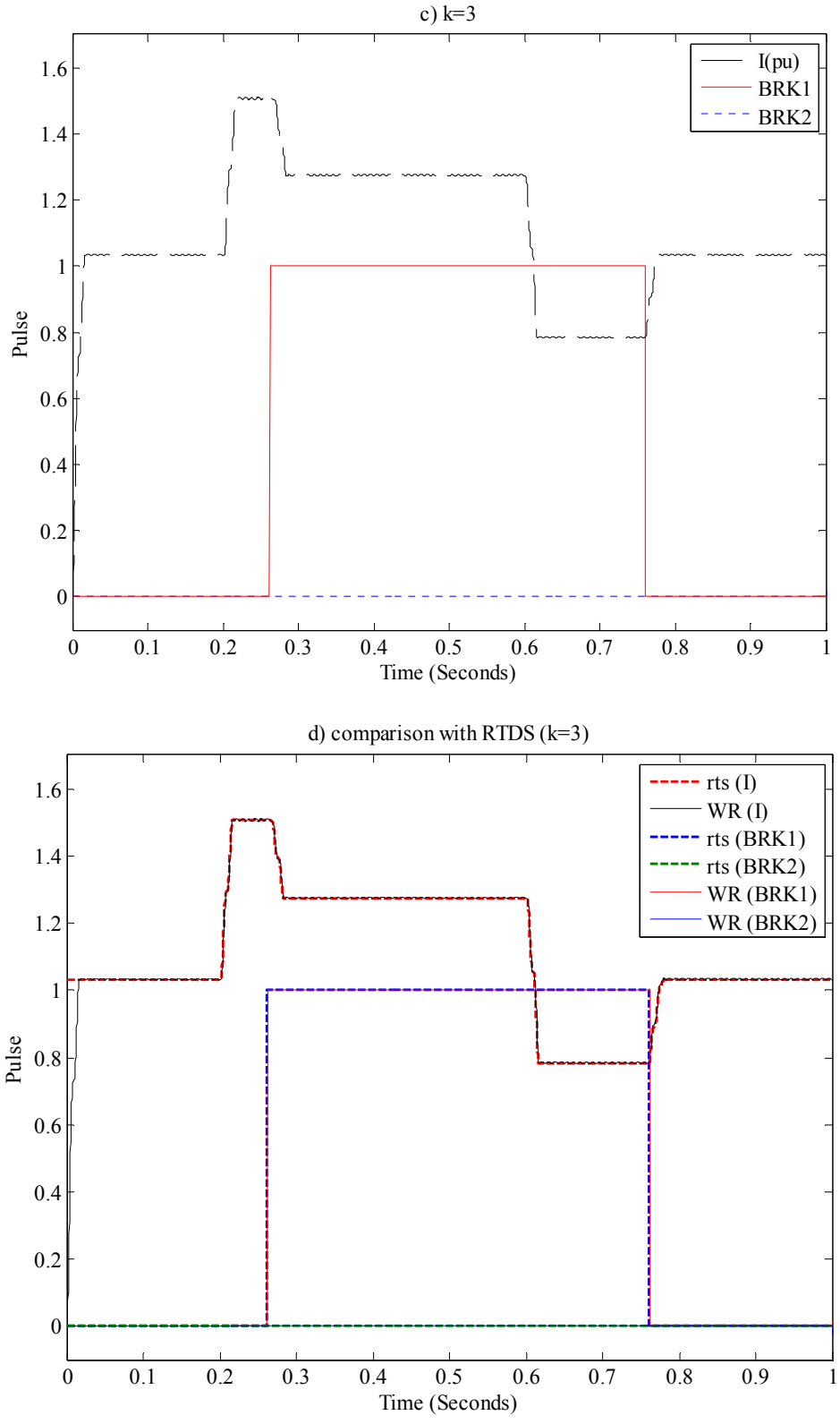


Figure 3-8. Experimental results of the WR based closed loop relay test example compared to RTDS [45].

Comparison with real-time HIL simulator*:* Conventional real-time simulation has been used for the same simulation case of Figure 3-7 using an RTDS platform. Figure 3-8-d also shows the HIL simulation result with the relay connected to the RTDS platform. The results are identical to those obtained from the WR-HIL simulation, thereby validating the proposed method.

3.5 Concluding Remarks

A hardware prototype for the Real-Time Player/Recorder (RTPR) device was designed to facilitate the Waveform Relaxation based Hardware-in-the-Loop (WR-HIL) simulation. The prototype shows that the RTPR can be simple in design, low cost, portable, and can communicate with the EMT simulator wirelessly via secured internet protocols. Two experimental examples of a transformer (PHIL) and a protective relay (HIL) were presented. In the PHIL simulation example, a voltage amplifier was used to amplify the output voltage generated by the RTPR. The WR-PHIL simulation example exhibited a fast convergence with the given power system model (convergence in 4 iterations). In the second example, the relay hardware changed the generated trip signals in each WR iteration according to the system simulation response. In convergence (at the third iteration), it was shown that the relay output trip signals remained the same as those in the second iteration. The two simple examples demonstrated the basic concept of the WR-HIL simulation. Additional validation was also carried by comparing WR-HIL with traditional real-time simulators.

The following two chapters discuss challenges in WR-HIL simulation convergence arising from noise and the topology being simulated. Convergence criteria and methods to improve the simulation convergence are also discussed.

Chapter 4

Challenges in WR-HIL Convergence

This chapter discusses convergence properties for WR and shows challenges faced by the algorithm when WR-HIL is attempted.

4.1 Convergence of WR

Two conditions are necessary for the WR to converge. Firstly, the theoretical convergence conditions, as is discussed in Section 4.2, should be met; and secondly, there should be no variations in any of the subsystems' parameters so that already converged parts of the waveforms do not change in following WR iterations.

In the first part of the section, the theoretical stability criteria of the WR as applied to the simulation of electric circuits are discussed. In the second part, it is shown that even with the stability conditions met, due to interfacing hardware inaccuracies, the classical WR can diverge when used with HIL simulation.

The condition for the classical WR convergence is that there should be no variation in any of the subsystems' parameters during WR iterations. When all subsystems are nu-

merically simulated as in the classical use of WR, this condition is easy to satisfy. However with WR-HIL, at least one subsystem is a hardware component, where the exchanged waveforms are analog. Hence, even though theoretical convergence conditions [18-25] are met, WR-HIL may not converge due to small interface errors as discussed below.

4.2 WR Stability for Circuits Simulation

The convergence properties of the WR algorithm are discussed in this section by considering the simulation of a circuit that is split into two independent subsystems. Each is simulated independently and interface waveforms are exchanged at the end of the simulation period. The resulting equations for the two subsystems can be coupled algebraic equations (AE), ordinary differential equations (ODE), or differential-algebraic equations (DAE) depending on the circuit structure. A WR operator matrix, by which the WR convergence characteristics are revealed, can be defined for each type as follows. This approach to WR convergence was investigated by J. White et al. [25].

4.2.1 Algebraic equations

When the resultant equations from a linear circuit split into two sub-circuits appear in the form of **Ax**=**b**, the WR method performs as an operator that decomposes the matrix **A** in three strictly lower triangular, strictly upper triangular, and diagonal matrices, **L**, **U**, and **D**, respectively [17, 19] where **A**=**L**+**D**+**U**. Here, the Gauss-Seidel method of waveform relaxation (GS-WR) described in Chapter 1 is considered. The GS-WR iteration

equations for the set of algebraic equations of **Ax=b** can be written as (4.1) (refer to Appendix 1 for more details).

$$\begin{aligned}\mathbf{x}^{k+1} &= -(\mathbf{L}+\mathbf{D})^{-1}\mathbf{U}\mathbf{x}^{k} + (\mathbf{L}+\mathbf{D})^{-1}\mathbf{b} \\ &= \mathbf{M}\mathbf{x}^{k} + (\mathbf{L}+\mathbf{D})^{-1}\mathbf{b}\end{aligned} \tag{4.1}$$

In (4.1), x is the vector of the WR interface waveforms. The "spectral radius" of the WR operator matrix **M**, is the magnitude of its largest eigenvalue, i.e., $\rho(\mathbf{M}_{n\times n}) = \max(|\lambda_i|), i \in \{1,2..\mathrm{n}\}$. The condition for convergence of the GS-WR method is that $\rho(\mathbf{M})$ should be less than unity [17, 22].

4.2.2 Ordinary differential equations

If the linear system equations are ODEs of the form $\mathbf{B}\dot{\mathbf{x}} + \mathbf{A}\mathbf{x} = \mathbf{b}$, the WR operator matrix **M** is defined as in (4.2) [27].

$$\mathbf{M} = -(\mathbf{L}_{\mathbf{B}} + \mathbf{D}_{\mathbf{B}})^{-1}\mathbf{U}_{\mathbf{B}} \quad , \quad \mathbf{B} = \mathbf{L}_{\mathbf{B}} + \mathbf{D}_{\mathbf{B}} + \mathbf{U}_{\mathbf{B}} \tag{4.2}$$

Again, WR converges when $\rho(\mathbf{M})$ <1.

4.2.3 Differential-algebraic equations

More often, the resulting equations from splitting a circuit are differential-algebraic (DAE). The DAE-Index-I [22, 26] form is as in (4.3). As before, $\mathbf{x}^k$ is the interface variable (could be a vector of variables) at iteration k, and $\dot{\mathbf{x}}^k$ is its derivative. Variable *x* corresponds to the differential equations and *z* to the algebraic equations. Superscript *k*+1 corresponds to the (*k*+1)th iteration.

$$\begin{cases} \dot{\mathbf{x}}^{k+1} = \mathbf{f}\left(\mathbf{x}^{k+1}, \mathbf{x}^{k}, \dot{\mathbf{x}}^{k+1}, \dot{\mathbf{x}}^{k}, \mathbf{z}^{k+1}, \mathbf{z}^{k}\right) \\ \mathbf{z}^{k+1} = \mathbf{g}(\mathbf{x}^{k+1}, \mathbf{x}^{k}, \dot{\mathbf{x}}^{k+1}, \dot{\mathbf{x}}^{k}, \mathbf{z}^{k+1}, \mathbf{z}^{k}) \end{cases} \tag{4.3}$$

From (4.3), the Lipschitz coefficients a_i and b_i, i={1,2,...,6} [18, 22] of both sets of ODEs and AEs can be obtained. A Lipschitz coefficient is described as the maximum rate of change of a function with respect to a variable. In (4.3), a_i is the maximum rate of change of **f** with respect to the i^{th} argument of **f** and b_i is the maximum rate of change of **g** with respect to i^{th} argument of **g**. For example in (4.3), $a_1 = \max\left|\frac{\partial \mathbf{f}}{\partial \mathbf{x}^{k+1}}\right|$, $a_2 = \max\left|\frac{\partial \mathbf{f}}{\partial \mathbf{x}^{k}}\right|$, …, $a_6 = \max\left|\frac{\partial \mathbf{f}}{\partial \mathbf{z}^{k}}\right|$, and $b_1 = \max\left|\frac{\partial \mathbf{g}}{\partial \mathbf{x}^{k+1}}\right|$, $b_2 = \max\left|\frac{\partial \mathbf{g}}{\partial \mathbf{x}^{k}}\right|$, …, $b_6 = \max\left|\frac{\partial \mathbf{g}}{\partial \mathbf{z}^{k}}\right|$.

Once these coefficients are calculated, the WR operator matrix can be defined as (4.4). Note that only some of the Lipschitz coefficients are required to define the WR operator matrix M.

$$\mathbf{M} = \begin{bmatrix} a_3 + a_4 & a_5 + a_6 \\ b_3 + b_4 & b_5 + b_6 \end{bmatrix} \tag{4.4}$$

As before, the $\rho(\mathbf{M})$ must be less than unity for convergence [18, 21]. Two examples below demonstrate the convergence analysis using a purely resistive circuit (which results in Algebraic Equations) and an RL-Circuit which results in differential-algebraic Equations.

4.2.4 WR convergence for the resistive voltage divider circuit

Consider the voltage divider circuit. A voltage type representation of the Ideal Transformer Model (ITM) [8] is used in Figure 4-1 to split the circuit from the connection point of the two resistors. This results in a purely algebraic model as in (4.5).

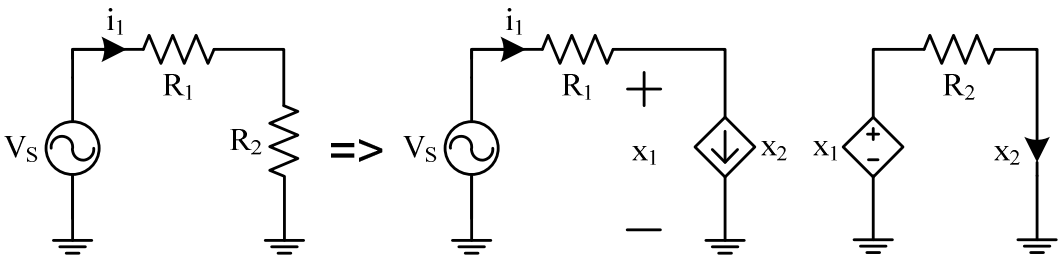


Figure 4-1. The voltage divider circuit splitting using voltage type of ITM interface.

$$\begin{bmatrix} 1 & R_1 \\ 1 & -R_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} V_S \\ 0 \end{bmatrix} \quad \equiv \quad \mathbf{Ax} = \mathbf{b} \tag{4.5}$$

Matrices **L**, **D**, and **U** as defined in (4.1) become (4.6):

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -R_2 \end{bmatrix} + \begin{bmatrix} 0 & R_1 \\ 0 & 0 \end{bmatrix} \tag{4.6}$$

The WR operator matrix can be defined as (4.7).

$$\mathbf{M} = -\left(\mathbf{L} + \mathbf{D}\right)^{-1}\mathbf{U} = \begin{bmatrix} 0 & -R_1 \\ 0 & -\dfrac{R_1}{R_2} \end{bmatrix} \tag{4.7}$$

The eigenvalues of **M** are $\lambda_1 = 0$ and $\lambda_2 = -\dfrac{R_1}{R_2}$. Therefore, $\rho(\mathbf{M}) = \dfrac{R_1}{R_2}$ indicating if WR method is used with the above splitting, R_1 must be smaller than R_2 for convergence. This convergence criterion is the same as that for traditional real-time simulation with the same split [47].

4.2.5 WR convergence for the R-L circuit

Figure 4-2 shows the two subsystems' circuits to be solved by the WR method. The extracted equations in terms of the waveform relaxation interface waveforms are shown in (4.8).

$$\begin{cases} x_1^{k+1} = -R_1 x_2^k - L_1 \dot{x}_2^k + V_S \\ \dot{x}_2^{k+1} = \dfrac{1}{L_2} x_1^k - \dfrac{R_2}{L_2} x_2^{k+1} \end{cases} \tag{4.8}$$

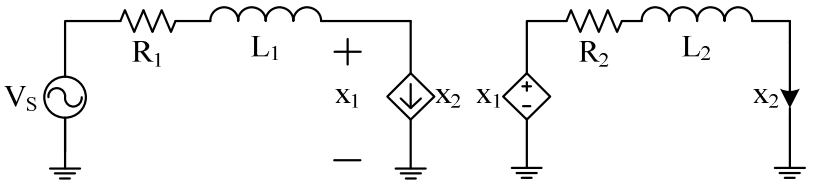


Figure 4-2. The R-L circuit partitioned by voltage type of ITM interface.

These are DAEs of index-I (see (4.3)). The Lipschitz coefficients in (4.8) are calculated and the WR operator matrix is obtained as in (4.9):

$$\mathbf{M} = \begin{bmatrix} a_3 + a_4 & a_5 + a_6 \\ b_3 + b_4 & b_5 + b_6 \end{bmatrix} = \begin{bmatrix} 0 & \dfrac{1}{L_2} \\ L_1 & 0 \end{bmatrix} \tag{4.9}$$

The spectral radius $\rho(\mathbf{M}) = L_1 / L_2$ should be less than unity to ensure WR convergence; i.e. $L_1 < L_2$. This convergence criterion of the same circuit using the direct methods of real-time HIL simulation [47] gives the same result, indicating that the WR appears to converge for the same circuit for which traditional HIL converges.

As seen above, when the WR is applied to solve electric circuits, the solution may or may not converge depending on the structure of the subsystems. This issue is more criti-

cal when there is nonlinearity in one or more subsystems and, therefore, the WR simulation may diverge in some operational regions. The WR-PHIL simulation example in Section 3.4.1 is such example. It is shown in Chapter 5 that depending on the length of the transmission line in the example, the WR-PHIL simulation may diverge. A stabilization technique is proposed in Chapter 5 to solve this issue.

4.3 WR-HIL Convergence Problems

Even if the theoretical stability criteria mentioned above are met, the WR-HIL simulation may still diverge due to several factors such as errors in measurement, playback system (RTPR), amplifier, noise, etc. In the following, a number of the issues that can cause WR to diverge are described.

4.3.1 Noise

Noise can affect analog waveforms generated and sampled by the RTPR. The random nature of noise appearing in the waveforms at the interface can cause divergence of WR-HIL by fluctuating already converged parts of the interface waveforms.

4.3.2 Analog-digital converters and interface amplifiers

The digital to analog (DAC) and analog to digital (ADC) converters use a finite bit length word and as such exhibit quantization errors. Additional errors are possible if additional analog interface amplifiers are present.

If the exchanged waveform contains digital pulses, the rated sampling frequency of the RTPR's A-D converters should be large enough. Otherwise, one or more pulses may appear in some WR iterations and disappear in others.

4.3.3 Initial state of the hardware under test

The HUT must have the same initial state in all WR iterations. In the case of a controller-hardware-in-loop (CHIL) if the controller is resettable, e.g., the controller's memory units reset to any desired initial value (or in some cases, default initial values), a reset signal can be applied to the controller via the RTPR just before playing back the waveforms. Otherwise, if the hardware is not re-initilizable, a sufficiently long initial period T_{init} for simulation must be included in the total time T, i.e. $T=T_{initial}+T_{study}$, where T_{study} represents the interval of interest in the simulation study so that the simulation converges to a steady state. In Power Hardware-in-the-Loop (PHIL) simulation, getting to an initial state is less of a problem as the external hardware can usually be de-energized easily and predictably, for instance by applying zero voltage to the HUT's inputs via the RTPR between the WR iterations.

4.3.4 WR-HIL convergence problem in testing relays

Modern protective relays manufacturers provide automatic relay reset-after-trip in relay settings for easy testing purposes. This can be also used when testing a relay using WR-HIL technique to avoid user interaction; here manually pushing a reset button after a fault is detected and so the relay is tripped, during the simulation. However, in legacy re-

lay hardware, the output contacts of the commercial protective relays remain latched after detecting a fault until a button is manually pressed. In practice, the operator presses this button after the cause of the fault is detected and cleared.

The WR algorithm converges only if the relay hardware is at the same initial conditions at the start of every iteration. During an iteration of the WR method, the simulated fault makes the protection relay trip. At this point, the WR cannot proceed until the relay contact is unlatched and reinitialized. Manual reset of the relay after every WR iteration is a solution but makes the solution dependent on the user interaction, which is not convenient and even impossible in cases where the relay hardware is in remote locations.

4.4 Concluding Remarks

It was shown in this chapter that implementation of the WR method in conducting the HIL simulation is not straightforward. Not only must the theoretical stability criteria be met to ensure the WR-HIL simulation will converge, but also further techniques must be added to the scheme to cope with the hardware noise, analog-digital conversion errors, amplifier errors, etc. Such issues do not exist in the classical WR as all subsystems were numerical computer models. Hence, there is no report of any algorithms that can fix the WR convergence against the above issues. Additionally, in testing the protective relays using the WR method, some security hardware settings such as manual reset of the relay can cause practical issues. Several solutions to these problems are proposed in Chapter 5.

Chapter 5

Stabilization and Accelerated Convergence Techniques for WR-HIL

In this chapter, the convergence problems described in Chapter 4 are solved by extending the classical Waveform Relaxation method to include several new convergence techniques. The resulting WR-HIL approach in this chapter is stable, reliable, and robust compared with the classical form of the WR in conducting the HIL simulation. First, a stabilization technique is proposed to increase the stability regions of the WR-HIL simulations. Then, several convergence techniques are proposed to enable convergence for the WR-HIL simulation in the presence of noise and other interface errors. A parametric analysis is also provided to investigate the efficacy of each technique individually and in combination with other techniques. Further, a solution for testing the protective relays with manual reset buttons using the WR method is presented followed by an experimental example.

5.1 WR-HIL Stabilization Technique

As was shown in Section 4.2, the WR-HIL simulation, like traditional HIL simulation, is prone to instability. Here, a method is proposed to improve the stability of the WR-HIL simulation. This technique is an extension of earlier time-domain simulation stabilization approaches used for interfacing state variable models to EMT simulation [48].

5.1.1 Method

Consider the system in Figure 5-1-a, which shows the original network split into two parts, each simulated separately by WR simulation. The stabilization technique will be presented to stabilize this network.

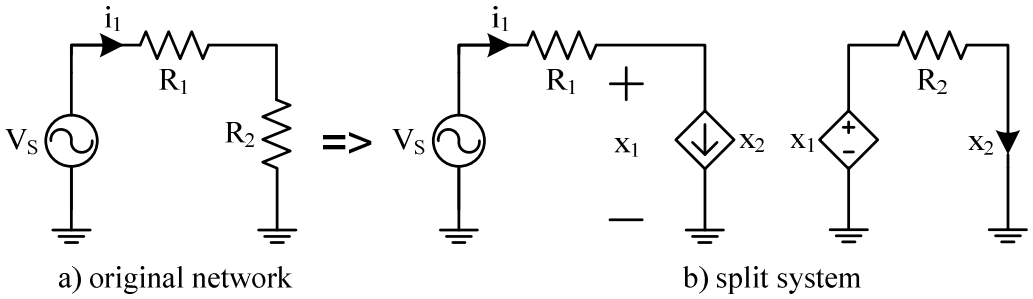


Figure 5-1. The voltage divider circuit splitting using voltage type of ITM interface.

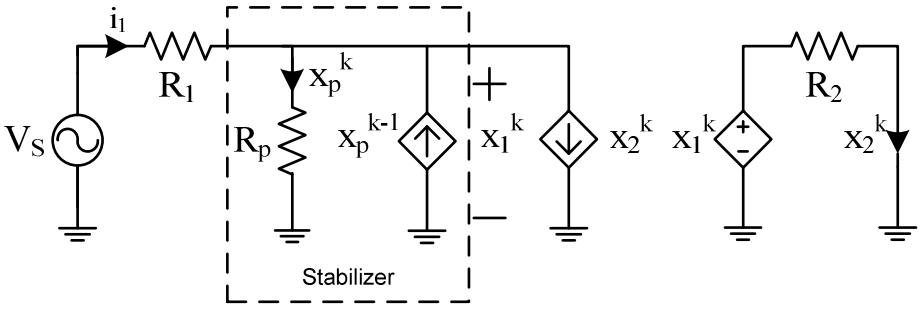


Figure 5-2. The voltage divider circuit with the proposed damping resistor to study the WR stability.

One stabilization solution is to add extra damping using an appropriately selected resistor R_p in parallel with the current source as shown in Figure 5-2; and then to compensate for the error caused by the resistor via the current source $\mathbf{x}_{p,w}^{k-1}$. In Figure 5-2, $\mathbf{x}_{p,w}^{k}$ and $\mathbf{x}_{p,w}^{k-1}$ are the vectors of the waveform samples (at time increments Δt) of the current x_p in the resistor R_p in the kth and (k-1)th iterations respectively (the former is computed in the present iteration and the latter is a recording from the previous iteration. This way the current i_1 is (5.1).

$$\mathbf{i}_{1,w} = \mathbf{x}_{2,w}^{k} + \Delta\mathbf{x}_{p,w}^{k};\ \text{where}\ \Delta\mathbf{x}_{p,w}^{k} = (\mathbf{x}_{p,w}^{k} - \mathbf{x}_{p,w}^{k-1}) \tag{5.1}$$

In general, $\mathbf{x}_{p,w}^{k}$ and $\mathbf{x}_{p,w}^{k-1}$ are waveforms of $\mathbf{x}_{p,w}$ one iteration apart, and if the solution converges, they will be exactly equal resulting in the compensation term $\Delta\mathbf{x}_{p,w}^{k}$ becoming zero, thereby negating any influence the compensation resistor may have had in the converged solution. Therefore, this technique improves simulation stability with minimal accuracy impairment.

In order to mathematically show the impact of adding the parallel resistor on the convergence stability of the WR, the example of Figure 5-2 is analyzed. The problem is classified in an algebraic set of equations, with the equations of the circuit in terms of the WR variables as in (5.2).

$$\begin{bmatrix} 1+\dfrac{R_1}{R_p} & R_1 & -R_1 \\ 1 & -R_2 & 0 \\ 1 & 0 & -R_p \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_p \end{bmatrix} = \begin{bmatrix} V_S \\ 0 \\ 0 \end{bmatrix} \tag{5.2}$$

The **L**, **D**, and **U** matrices can be found in (5.3).

$$\mathbf{L}=\begin{bmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\end{bmatrix},\mathbf{D}=\begin{bmatrix}1+\dfrac{R_1}{R_p} & 0 & 0\\ 0 & -R_2 & 0\\ 0 & 0 & -R_p\end{bmatrix},\mathbf{U}=\begin{bmatrix}0 & R_1 & -R_1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \tag{5.3}$$

The M matrix is calculated in (5.4).

$$\mathbf{M}=-\left(\mathbf{L}+\mathbf{D}\right)^{-1}\mathbf{U}=\begin{bmatrix}0 & -R_{1p} & R_{1p}\\ 0 & -\dfrac{R_{1p}}{R_2} & \dfrac{R_{1p}}{R_2}\\ 0 & -\dfrac{R_{1p}}{R_p} & \dfrac{R_{1p}}{R_p}\end{bmatrix} \tag{5.4}$$

$$\text{or } \rho(\mathbf{M})=R_{1p}\left(\frac{1}{R_2}-\frac{1}{R_p}\right);\ \text{where } R_{1p}=\frac{R_1R_p}{R_1+R_p}$$

Finally, the convergence condition for the WR solution is from (5.5).

$$\rho(\mathbf{M})=R_{1p}\left(\frac{1}{R_2}-\frac{1}{R_p}\right)<1 \ \Rightarrow\ R_p<\frac{2R_1R_2}{R_1-R_2} \tag{5.5}$$

From (5.5), for a given R_1 and the smallest possible R_2 (as R_2 can change through switching action), an appropriate R_p can be selected as in (5.5) to guarantee convergence of the WR for all the operating conditions of the system. Although R_p can be calculated analytically for linear systems, this is generally not possible for nonlinear systems. Hence R_p is usually selected by trial and error.

5.1.2 Example

Consider the PHIL example of Figure 3-3 in Chapter 3, where an actual unloaded transformer is in the loop. The connected power system includes a source and a transmission line. The connected system is simulated in off-line EMT simulation (on

PSCAD/EMTDC). In that example (Section 3.4.1), a short transmission line (L_S=10 mH) was used in the simulation model and the WR-HIL was stable. In this section, it is shown that the system with a 5 times longer line (L_S=50 mH) becomes unstable when the WR is applied without the proposed stabilization technique. Here, the instability issue is described with the consideration of the nonlinear behavior of the HUT (the transformer in the loop has a nonlinear saturation characteristic).

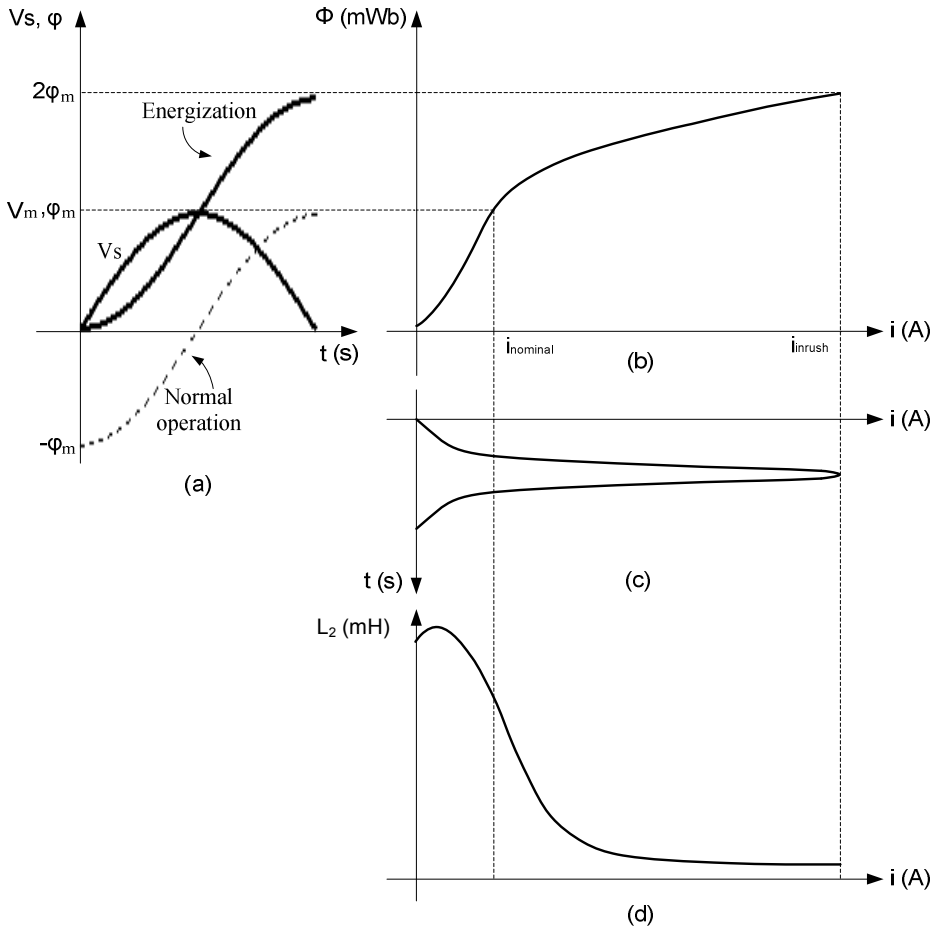


Figure 5-3. Inrush current and magnetizing inductance of a typical power transformer.

Consider Figure 5-3. Energizing a transformer can cause a large inrush current [49], which is a function of the transformer's saturation curve (Figure 5-3). The corresponding

instantaneous incremental inductance [50, 51, 52] $L_2(i) = \frac{d\varphi(i)}{di}$ is plotted in Figure 5-3 (d).

Note that the incremental inductance in Figure 5-3 (d) attains very high as well as very low values. In Section 4.2.5, it was shown that the WR can become unstable if the inductance of the hardware side (L_2) is smaller than that of the simulated side (L_1). Hence, depending on the inductance values, the WR-PHIL simulation may be stable when the incremental inductance is high and unstable when it is low. This is observed from the HIL simulation results presented in Figure 5-4.

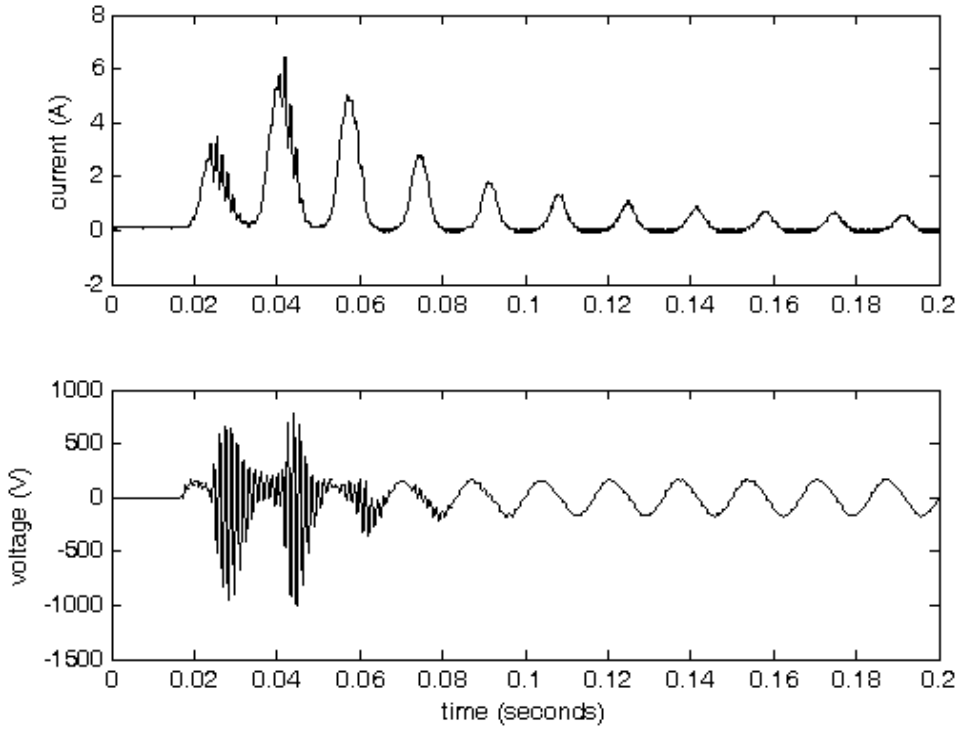


Figure 5-4. WR-HIL nonconvergence for the long model of the transmission line in the transformer energization example (see Figure 3-4).

The proposed stabilization method described in Figure 5-2 is applied for this unstable case (as shown in Figure 5-5) and the WR converges in 7 iterations.

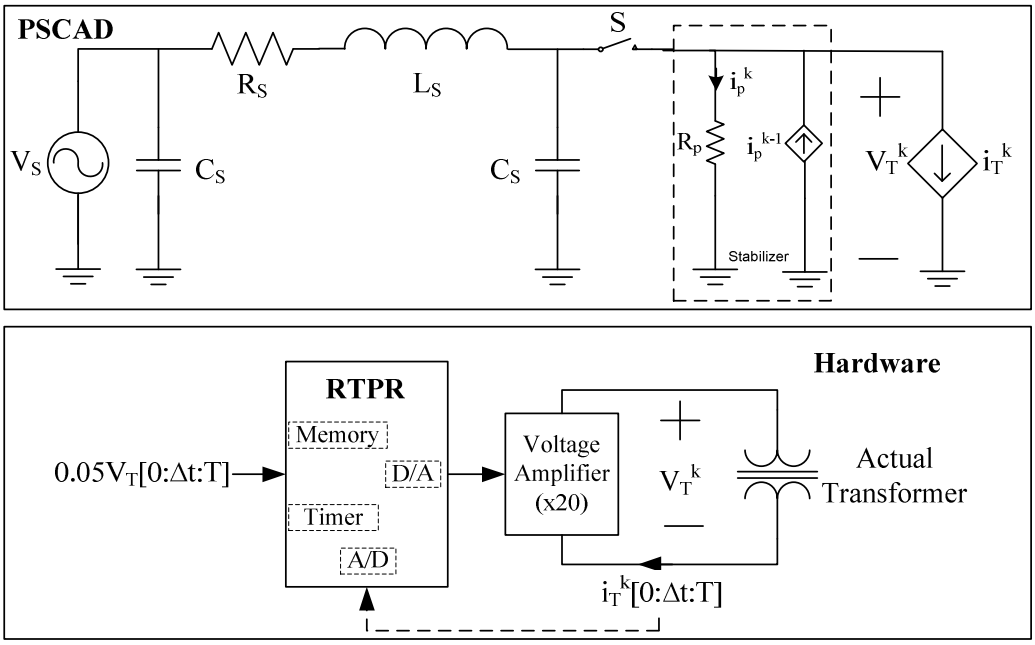


Figure 5-5. Addition of the stabilizing circuit to the PSCAD/EMTDC system model.

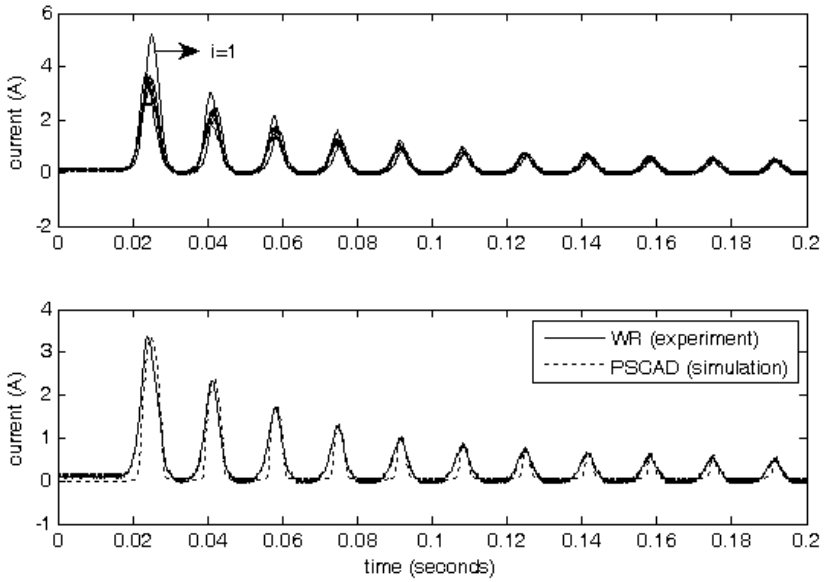


Figure 5-6. Stability of WR-HIL in the transformer energization example with longer transmission line by application of the proposed stabilization technique.

The results are shown in Figure 5-6 and when compared to the PSCAD/EMTDC simulation result for cross-validation, the comparison is good. The PSCAD/EMTDC trans-

former model's parameters for this case were kept the same as those in the short line test case in Section 3.4.1. The value of the parallel damping resistor was selected by trial and error (R_p=50 Ω).

As a result, the addition of the damping resistor to the power system model stabilizes the WR-PHIL simulation for the unstable case of the long transmission line.

5.2 Accelerated Convergence Solutions for WR-HIL

Figure 5-7 shows the HIL simulation block diagram using the classical WR approach with external hardware under test. D_1 and D_2 respectively represent input and output side distortions from A/D and D/A errors and noise.

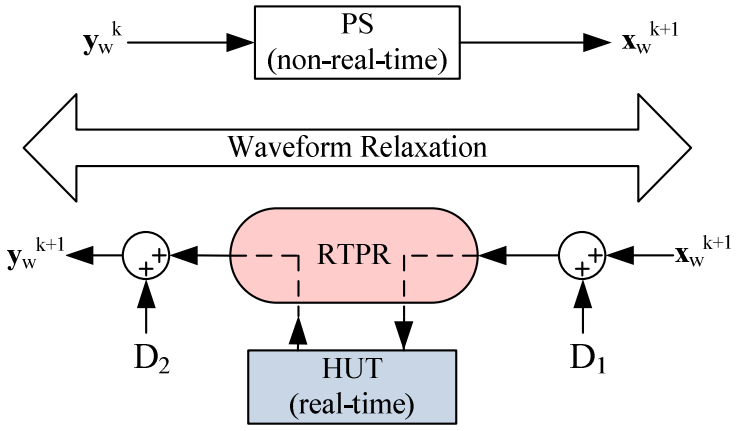


Figure 5-7. WR-HIL simulation affected by hardware distortions D1 and D2.

Assuming Δt is the simulation time-step and T is the total simulation time, the actions performed at classical WR iteration 'k+1' are:

1. Using the HUT's output from the previous WR iteration ($\mathbf{y}_w^k$), the PS is simulated and its response $\mathbf{x}_w^{k+1}$ is obtained;

2. The RTPR makes $\mathbf{x}_w^{k+1}$ available to the HUT in real-time (after suitable resampling with time-step Δt_2) and records the response waveform $\mathbf{y}_w^{k+1}$ of the HUT; and

3. Convergence is checked. If no convergence, the next iteration is called for.

It will be shown in later examples that the classical WR as shown can exhibit poor or no convergence if D_1 and D_2 are present. In following sub-section, new techniques are proposed for enabling and accelerating convergence.

5.2.1 Piecewise Fixed Convergence Method (WR-PWFC)

Algorithm

WR convergence is piecewise, i.e., as iterations continue, the initial portion of the waveform begins to converge first; and the later portions of the waveform converge in later iterations, without affecting the earlier converged part. Noise creates a problem for the convergence, and may cause already converged portions to stray from their converged values in later iterations. The proposed WR-PWFC method identifies the already converged part (window) of a waveform from earlier iterations by comparing a waveform in the current WR iteration with that of the previous iteration. Once the window is identified, that part of the waveform is fixed and not allowed to change in any future iteration. The length of the converged window increases from zero (in the first iteration) towards T, the total simulation time, (in last iteration). Considering the system of Figure 5-7, the following steps are taken in WR iteration k+1:

1. The HUT's response waveform in iteration k, $\mathbf{y}_w^k$, is applied to the simulation of the power system (PS) which gives an output $\mathbf{x}'_w{}^{k+1}$. This is a candidate waveform and is modified to its final value $\mathbf{x}_w^{k+1}$ as indicated below;

2. $\mathbf{x}'_w{}^{k+1}$ is compared with $\mathbf{x}_w{}^k$ from the previous iteration to determine if a larger portion of the waveform has converged. If so, the convergence window size CW^k (in time units) is increased to CW^{k+1} to include the interval of the converged part of the waveform;

3. The waveform $\mathbf{x}_w{}^{k+1}$ is constructed by only updating the non-converged part (i.e. for $t > CW^{k+1}$) as in (5.6):

$$\begin{aligned} &\forall t \in [0, \Delta t, .., CW^{k+1}]: \; x_w{}^{k+1}(t) = x_w{}^k(t); \\ &\forall t \in [CW^{k+1} + \Delta t, .., T]: \; x_w{}^{k+1}(t) = x'_w{}^{k+1}(t) \end{aligned} \tag{5.6}$$

4. The updated waveform $\mathbf{x}_w{}^{k+1}$ is applied to the HUT; and

5. The process iterates until $CW=T$.

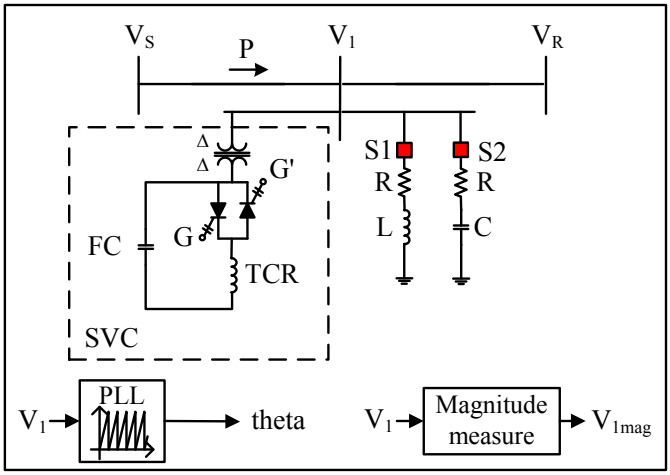
Note this is different from time windowing as was described and found unsuitable in Section 2.2. In that case the entire simulation was a-priori divided into several windows and WR was conducted separately in each window. In contrast, here, each simulation is still conducted in the whole interval [0, T], but the converged part is "locked" and is no longer updated.

Experimental example

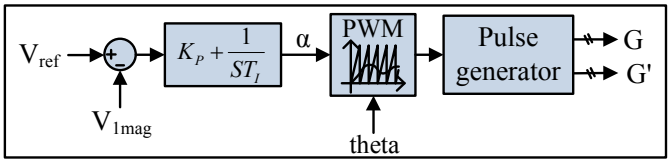
The 3-phase, two bus, 138 kV, 60 Hz power system of Figure 5-8, is modelled in the off-line EMT simulator (PSCAD/EMTDC).

A Static Var Compensator (SVC) with Fixed Capacitors (FC) rated at 140 MVar and Thyristor Controlled Reactors (TCR) rated at 330 MVar regulates the mid-point voltage of the transmission line to 1 pu (138 kV). The Short Circuit MVA at the point of connection is $1000\angle 82.6^{o}$ MVA. A phase locked loop (PLL) generates a sawtooth waveform synchronized to the mid-point ac voltage V_1 and is used for timing. The per-unit magnitude of V_1 (denoted by $V_{1\text{mag}}$ in Figure 5-8) is compared with a set point V_{ref} and the error

is fed to a proportional-integral (PI) regulator, which determines the firing angle α of the TCR. The PI gains are K_P=1 and T_I=0.01. The generated pulses from the controller, G and G′, are applied to six thyristors of the SVC. An R-L load rated at 110 MVA with the power factor of 0.6 is connected via breaker S1 to the mid-point at t=0.3 Sec. The R-L load is disconnected and, concurrently, an R-C load rated at 170 MVA with the power factor 0.6 is connected to the bus via S2 at t=0.6 Sec.



Power network simulated in non-real time PSCAD



Control System implemented in real-time hardware

V_S	138kV ∠ 10°
V_R	138kV ∠ 0°
RL load	110 MVA ∠ 53°
RC load	170 MVA ∠-53°
K_P	1
T_I	0.01

System specifications

Figure 5-8. WR-PWFC example for testing an SVC controller hardware.

The classical WR did not converge, due to the existence of noise and interface errors in the control hardware. The diverged WR results for V_{1mag} in 1st, 15th, 30th, 45th, and 60th

iterations are shown in Figure 5-9 (a). It can be seen that the convergence did not progress beyond (approximately) the first 0.1 seconds.

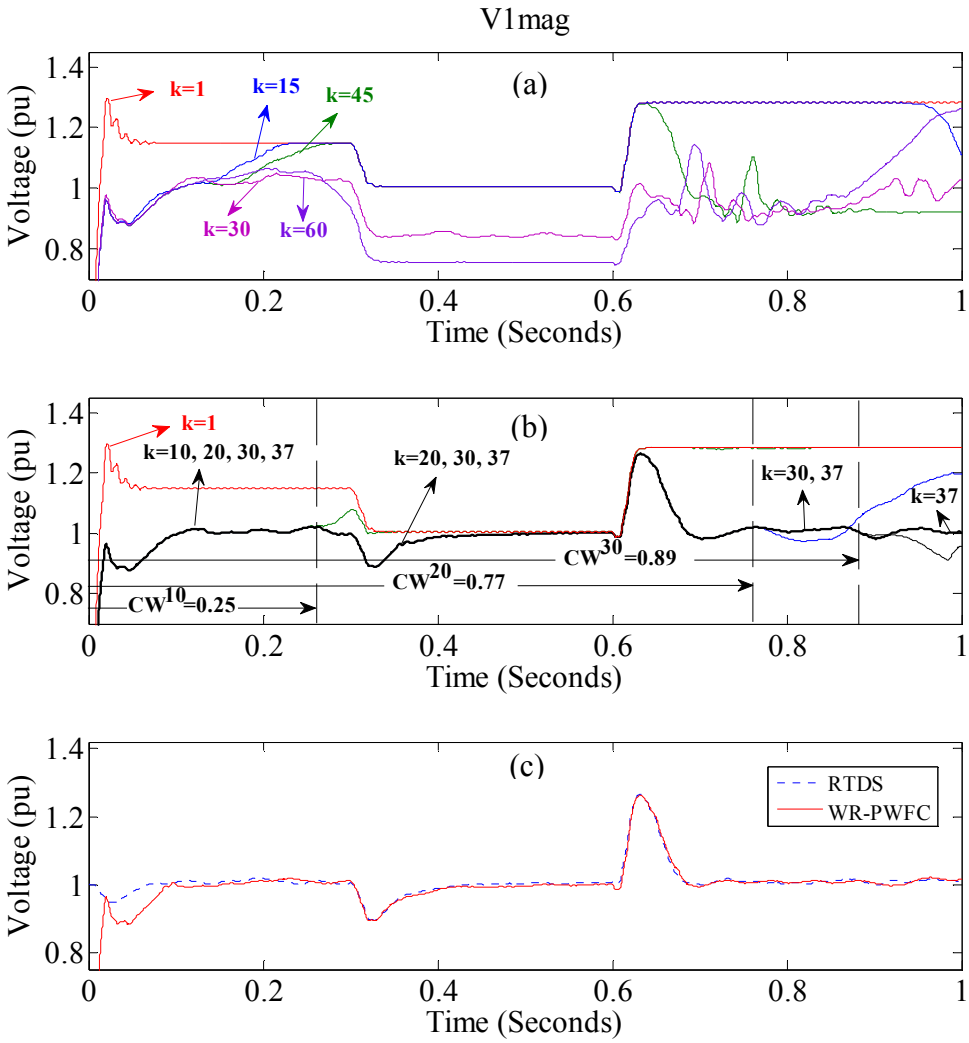


Figure 5-9. (a) Classical WR-HIL divergence for the system of Figure 5-8, (b) WR-PWFC based HIL convergence shown for V1mag, (c) Comparison of V1mag waveform obtained from RTDS and WR-PWFC.

When the WR-PWFC technique was applied, the simulation converged in 37 iterations. Figure 5-9 (b) plots V_{1mag} in the 1st, 10th, 20th, 30th, and 37th (last) WR iterations.

The power system was also modelled in a Real-Time Digital Simulator (RTDS) platform using the RSCAD software to find the closed loop response of the SVC controller hardware for cross-validation. Figure 5-9 (c) compares the RTDS closed loop response

and the WR-PWFC converged response for $V_{1\text{mag}}$. The close agreement between the waveforms shows that the proposed approach is an effective alternative to real-time simulation. The slight difference at the beginning is because of slightly different start up states for the two simulation platforms.

WR acceleration: The WR-PWFC acceleration approach discussed at the Section 5.2.1 is demonstrated below. Figure 5-9 (b) shows the $V_{1\text{mag}}$ waveform for the time period of [0s, 1s] in the 10th, 20th, 30th, and 37th WR iterations. The convergence window size is determined by the convergence limit 'ε' set here as 0.0005. For example, in the 10th iteration, the new convergence window is $CW^{k=10}$=0.25 seconds because the difference between the two waveforms ($\mathbf{V}^{9}_{1mag,w}$ and $\mathbf{V}^{10}_{1mag,w}$) is less than the threshold ε=0.0005 for the time period of [0s 0.25s]. As can be seen, as the iteration count grows, the converged window size also grows, i.e. $CW^{37} > CW^{30} > CW^{20} > CW^{10}$. As discussed in the WR-PWFC algorithm above, speedup is achieved by maintaining the controller hardware's input waveforms, *theta* and $V_{1\text{mag}}$, in the time period [0s, 0.25s] at the 10th iteration values for the entire WR simulation (next 27 iterations). Similarly for all iterations after the 20th, the waveforms applied to the hardware are maintained as simulated in [0s, 0.77s], and so on. The process converges in the range of interest, i.e. [0s, 1s] in 37 iterations, when CW^{37} = 1 second.

5.2.2 WR with internal control model (WR-ICM)

Another approach developed in this thesis to speed up the WR-HIL simulation is to include an approximate model of the HUT in the power system model (PS). The feature

of including a model in the simulator is analogous to the use of model based controllers in control theory referred to as "Internal Model Control (IMC)" [53]. In IMC, a model of the plant being controlled is included in the controller implementation to make the controller more open-loop and hence faster as feedback from the plant is not as critical if the model is accurate enough. The actual feedback is used only to correct the model's feedback. In this thesis, a model of the controller is included in the power system in addition to the external controller hardware being tested.

Two approaches have been designed. Approach 1 is used when algebraic operations of addition and subtraction are applicable, i.e., the HUT's output response does not include digital signals such as firing pulses to power electronic switches. Approach 2 is designed to enable the technique when one or more of the HUT's output waveforms contain digital signals.

Approach 1 (WR-ICM1)

Approach 1 includes a model of the HUT, referred to as the *"Internal Control Model"* (ICM) within the power system model as shown in Figure 5-10. The ICM may be an approximate model as the HUT's precise workings may be unknown, which is often the case with proprietary control hardware from manufacturers. In Figure 5-10, SubSys1 is the simulation subsystem containing the PS and HUT models. Since the HUT's model is included within the EMT simulation, and largely mimics the effect of the HUT, only a small correction is required from actual HUT. This minimizes the impact of noise and amplifier error, etc. However, the HUT still sees the actual waveforms from the system and so its response is true. However the HUT's response is only used to make corrections

to the response of the model of the HUT in the simulation. The WR-ICM1 procedure in iteration 'k+1' is as follows:

1. The model approximation error waveform $\Delta\mathbf{y}_w^{\,k}$ for iteration k is the mathematical difference between the approximate HUT output response waveform $\mathbf{y}_{\mathbf{1},w}^{k}$ from the model included in the EMT simulation, and the actual waveform $\mathbf{y}_w{}^{k}$ from the actual HUT as in (5.7).

$$\Delta\mathbf{y}_w^{\,k} = \mathbf{y}_w^{\,k} - \mathbf{y}_{\mathbf{1},w}^{k} \tag{5.7}$$

2. The correction $\Delta\mathbf{y}_w^{\,k}$ is added to the feedback $\mathbf{y}_{1,w}{}^{k+1}$, from the simulated HUT model for more accuracy to provide the refined HUT feedback waveform $\mathbf{y}_w''{}^{k+1}$;

3. The RTPR plays back $\mathbf{x}_w{}^{k+1}$ in real-time to the HUT and records the HUT's response waveform $\mathbf{y}_w{}^{k+1}$.

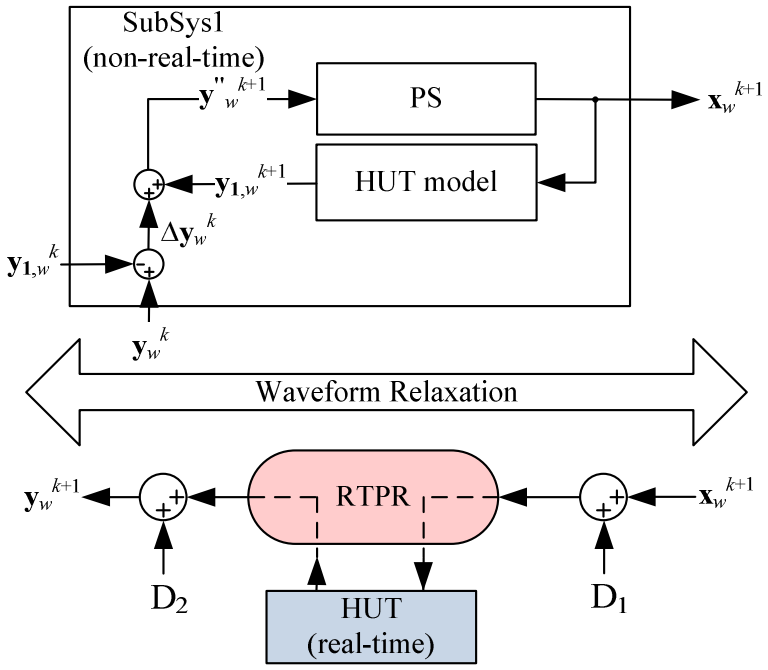


Figure 5-10. Approach 1 of the WR with Internal Control model (WR-ICM1).

It is easy to show that on convergence, the approach gives the correct answer. From Figure 5-10, supposing that the WR converges in *N*+1 iterations, (with $k = N$):

$$\mathbf{y}''^{N+1}_{w} = \mathbf{y}^{N+1}_{\mathbf{1},w} + \Delta\mathbf{y}^{N}_{w} = \mathbf{y}^{N}_{w} + (\mathbf{y}^{N+1}_{\mathbf{1},w} - \mathbf{y}^{N}_{\mathbf{1},w}) = \mathbf{y}^{N}_{w} \tag{5.8}$$

From (5.8), when all waveforms converge, $(\mathbf{y}^{N+1}_{\mathbf{1},w} - \mathbf{y}^{N}_{\mathbf{1},w})$ becomes zero. This shows that the simulated model (with response $\mathbf{y_1}$) does not affect the final result; although it does affect the convergence rate. The more accurate the model of the HUT, the faster the convergence of the WR. For example, if the simulated HUT's model is exact, y^{k}_{w} and $y^{k}_{1,w}$ are exactly the same in the first WR iteration (k=1) and from (5.8) the model converges in the first iteration (although the algorithm will not know this till iteration 2 as it has not yet had the opportunity to compare the waveform with the previous iteration).

Approach 2 (WR-ICM2)

As before, a model of the HUT is used in WR-ICM2 to accelerate WR-HIL convergence. This method is only applied when the HUT outputs are digital signals, e.g. firing pulses, relay trip signals, etc. For these signals, subtracting waveforms as was done to generate the correcting signal $\Delta\mathbf{y}$ in Figure 5-10 is meaningless. Figure 5-11 shows the basic principle.

Here, the correction to the simulation is made at the input of the simulated HUT model, rather than at output to the PS as was the case in WR-ICM1. To overcome this problem, the output waveform of the HUT, $\mathbf{y}_w^{k+1}$, is fed back into another non-real-time EMT simulation of the PS, (block labelled SubSys2) in order to generate another PS output waveform $\mathbf{x}_w^{k+1}$, which is the most recent estimate of the output of PS corresponding to the new HUT output $\mathbf{y}_w^{k+1}$. The difference $\Delta\mathbf{x}^{k}_{w} = \mathbf{x}^{k}_{w} - \mathbf{x}''^{k}_{w}$ between the PS output (from

SubSys2) corresponding to the most recent HUT response and the PS output with the HUT model included (SubSys1) can now be applied as a correction on the output side of the PS model in Subsys1 as shown in Figure 5-11.

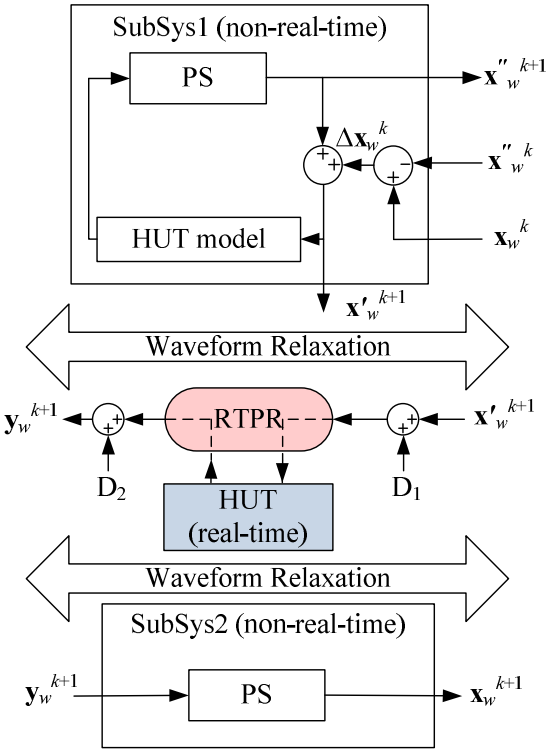


Figure 5-11. Approach 2 of the WR with Internal Control model (WR-ICM2).

Experimental example

This example considers the First Cigre Benchmark Model for High Voltage Direct Current (HVDC) [54]. The HVDC system of Figure 5-12 is a monopolar, 12-pulse, 500 kV, 1 GW system with fixed capacitors at both sides, each rated at 125 MVar, and including damped filters at both sides, each with a total of 500 MVar of reactive power compensation. The DC line resistance is 5Ω and Short Circuit Ratios (SCR) at both rectifier and inverter sides are 2.5.

The HVDC power system is modelled in the off-line PSCAD/EMTDC simulation program and the rectifier and inverter controllers are implemented in physical hardware as the HUT. The exact structure of the controllers (Figures 6-9 and 6-10) is as in [55]. Two RTPR devices are used, one each for the rectifier and inverter side controllers.

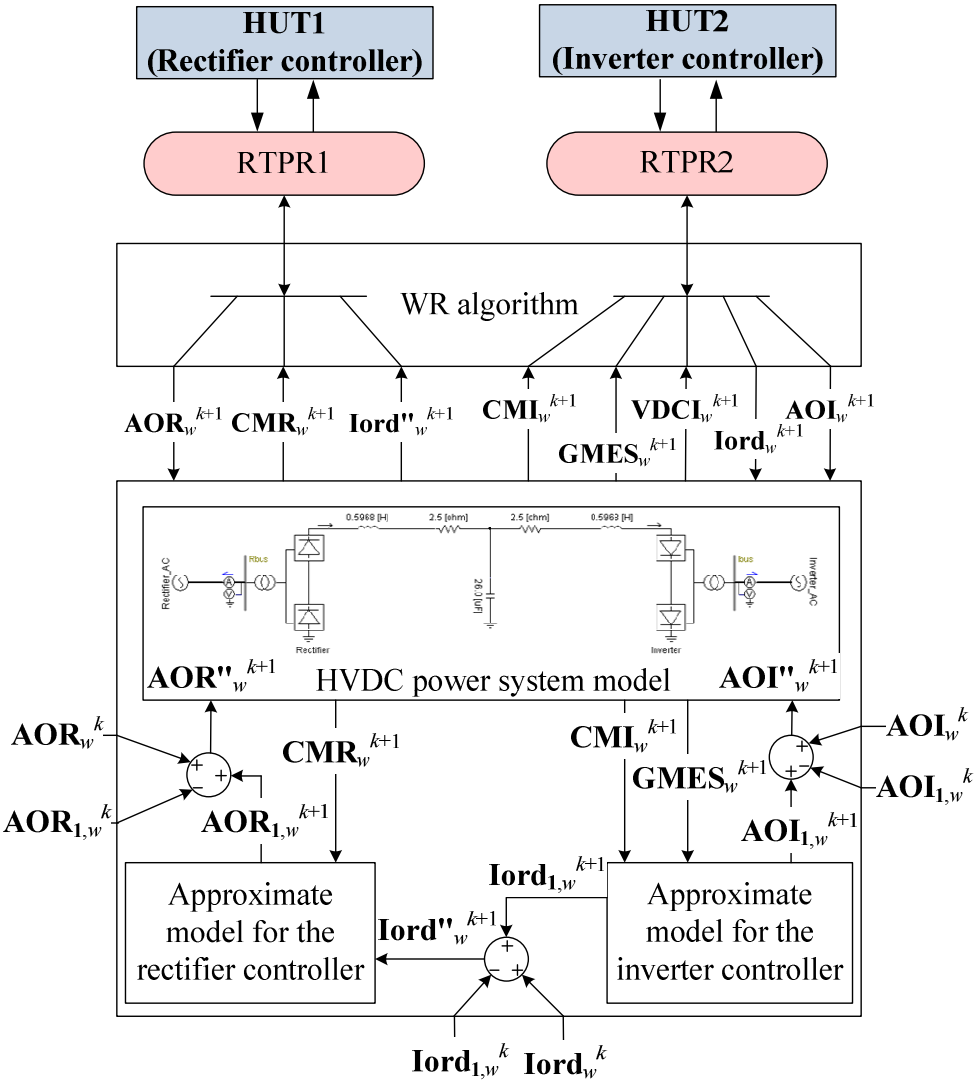


Figure 5-12. WR-ICM1 for testing HVDC controllers: *AOI*: inverter angle order, *AOR*: rectifier angle order, *VDCI*: inverter DC voltage, *CMI*: dc current measured at inverter side, *CMR*: dc current measured at rectifier side, *GMES*: Gamma angle measured at inverter, *Iord*: rectifier current order from inverter.

In Figure 5-12, for the inverter controller, the measured waveforms of the inverter extinction angle (*GMES*), inverter side dc voltage and current (*VDCI* and *CMI*) are inputs and the response waveforms are inverter angle order (*AOI*) and rectifier current order (I_{ord}). The PI gains/time constants for the current controller are $K_{P,C}$=0.63, $T_{I,C}$=0.015, and for the extinction angle controller are $K_{P,E}$=0.75, $T_{I,E}$=0.054 respectively [55].

WR-ICM1 is used with the assumed internal models of the rectifier and inverter side controllers as shown in Figure 5-13 and 5-14. The rectifier side PI controller gains are purposely set to K_P=1 and T_I=0.02, which are clearly different from those of the hardware. Similarly, the inverter controller gains are K_P=1 and T_I=0.02 for both the current and extinction angle controllers.

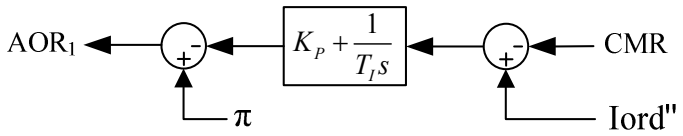


Figure 5-13. A PSCAD/EMTDC approximate model for the rectifier controller.

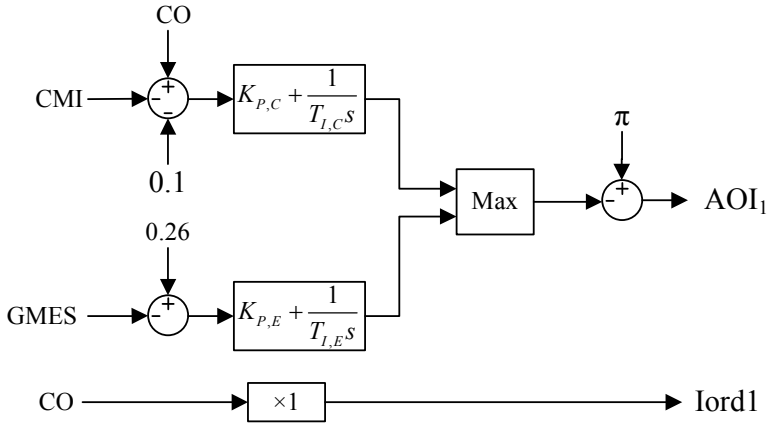


Figure 5-14. Approximate model for the inverter controller. *CO*: reference current.

Classical WR (where no internal models are included) did not converge, probably due to interface noise. With WR-ICM1, convergence was obtained in 26 iterations. Figure 5-

15 shows the simulated inverter firing angle order (*AOI*) waveform in different WR iterations from iteration 1 to 26.

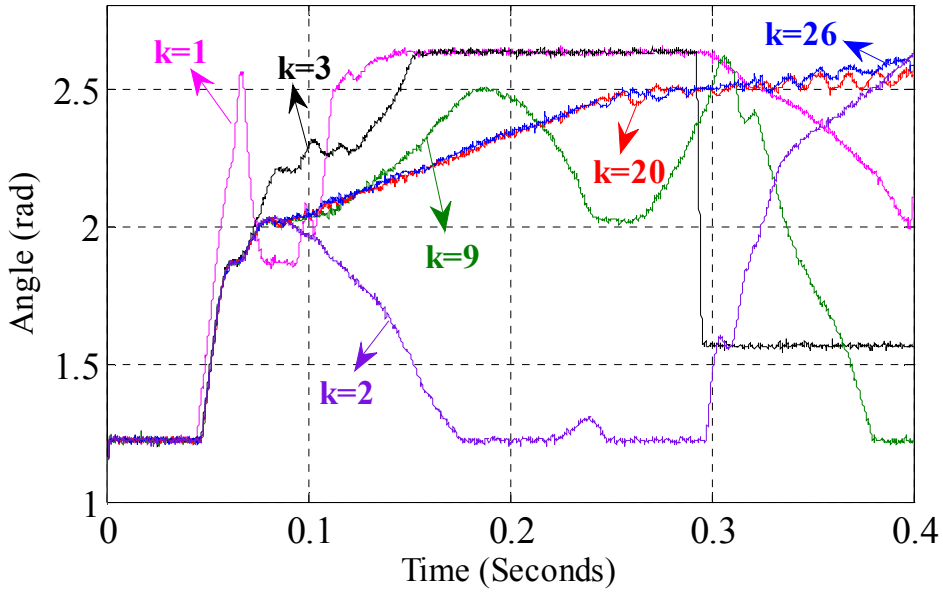


Figure 5-15. Inverter angle order waveform in WR-ICM1 iteration.

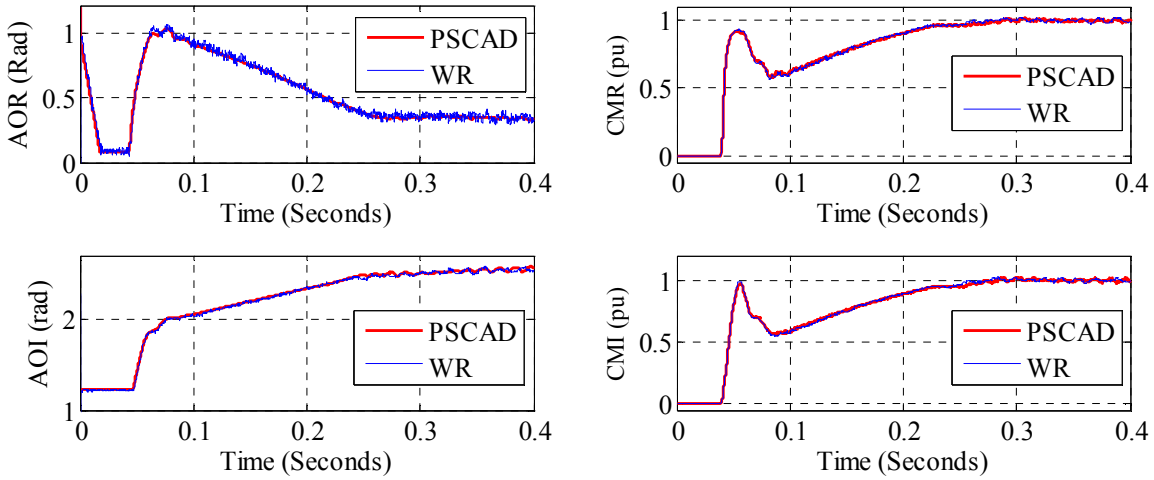


Figure 5-16. Comparison of WR-ICM1 converged waveforms with direct PSCAD/EMTDC simulation in testing HVDC rectifier and inverter controllers hardware.

The converged WR-ICM1 results for *AOR*, *AOI*, *CMR*, and *CMI* are shown in Figure 5-16. To ensure that the results from WR are acceptable, overlaid on these curves are results obtained by conducting a pure off-line PSCAD/EMTDC simulation with the HUT

and all controls modelled in simulation. In this case the HUT controller's structure was known so a detailed model could be developed all in software. The results are essentially the same (apart from some ripple due to the fact the HUT is physical hardware in the WR simulation and has noise and analog-digital conversion errors). The above discussion confirms the ability of the WR-ICM1 method to conduct HIL simulations.

5.2.3 Waveform Relaxation averaging method (WR-AM)

Algorithm

The WR-AM method proposed here is an alternative convergence method to the WR-PWFC and WR-ICM approaches discussed above. Due to noise or A-D converters errors, there may be slight variations in the converged parts of the waveform from iteration to iteration. To compensate, WR-AM proposes to use a weighted average of the HUT waveforms from the last few (L) consecutive WR iterations. L is selected by the user and L=5 to 10 is observed to work well in most of cases. It works as follows; in iteration k:

The average of last L iterations for the sampled hardware response, $\mathbf{y}_w$, is (5.9).

$$\overline{\mathbf{y}}_{\mathbf{L},w}^{k} = \frac{\sum_{i=k-L+1}^{k} (\mathbf{y}_w{}^{i})}{L} \tag{5.9}$$

The error waveform $\mathbf{d}_w{}^k$ between the present iteration waveform ($\mathbf{y}_w{}^k$) and the average ($\overline{\mathbf{y}}_{\mathbf{L},w}^{k}$) is (5.10).

$$\mathbf{d}_w{}^k(t) = \left| \mathbf{y}_w{}^k(t) - \overline{\mathbf{y}}_{\mathbf{L},w}^{k}(t) \right|,\ t = 0, \Delta t, ..., T \tag{5.10}$$

A counter vector $\mathbf{N}_w{}^k$ records how well a sample point in the waveform has converged. For example, if the error $\mathbf{d}_w{}^k(m\Delta t)$ is less than a convergence threshold β, then the

element $\mathbf{N}_w^k(m\Delta t)$ is incremented by one (initially $\mathbf{N}_w^0$=[1,1,..,1]). Thus regions in which the waveform is closer to the average are given a higher weight.

Weighting waveform $\mathbf{W}_w^k$ is found from (5.11) with a constant K to control WR convergence rate; and with a maximum allowable weight of unity. Note that as N_w^k is in the denominator, higher weight is given to unconverged portions of the HUT output.

$$\mathbf{W}_w^k(t) = \min\left(\frac{K\mathbf{d}_w^k(t)}{\mathbf{N}_w^k(t)}, 1\right), t = 0, \Delta t, ..., T \tag{5.11}$$

The HUT output waveform $\mathbf{y}_{\mathbf{f},w}^k$ to be fed back to the EMT simulator for iteration k+1 is constructed as a weighted sum of the latest HUT output $\mathbf{y}_w^k$ and the average waveform $\overline{\mathbf{y}}_{\mathbf{L},w}^k$ as in (5.12), where the multiplication is done for corresponding samples of the waveforms being multiplied:

$$\mathbf{y}_{\mathbf{f},w}^k(t) = \mathbf{W}_w^k(t)\cdot\mathbf{y}_w^k(t) + \left(1-\mathbf{W}_w^k(t)\right)\cdot\overline{\mathbf{y}}_{\mathbf{L},w}^k(t), t = 0, \Delta t, ..., T \tag{5.12}$$

This way converged parts of the waveform (where $\mathbf{W}_w^k$ is small and so (1-$\mathbf{W}_w^k$) is close to unity) are given a larger weighting and the latest contribution sample $\mathbf{y}_w^k$ is given a smaller weighting $\mathbf{W}_w^k$. As the converged portion $\overline{\mathbf{y}}_{\mathbf{L},w}^k$ is an average waveform, new noise in the present iteration does not significantly affect it. On the other hand, unconverged parts, where $\mathbf{W}_w^k$ is large, correspond to the changes resulting from the present iteration and are given more weight. This approach thus prevents noise and other errors from upsetting the already converged waveform portion.

The WR convergence can be controlled by K in (5.11). For example, if the noise is small, a large value of K makes WR converge faster. With small K, the WR convergence is slow but less distorted by the noise. In this work, K was selected by trial and error.

Experimental example

This case demonstrates the HIL simulation of a current controller for a single phase thyristor controlled rectifier circuit with an R-L load as shown in Figure 5-17. The aim of the circuit and controller is to regulate the dc current in the R-L load to its desired reference.

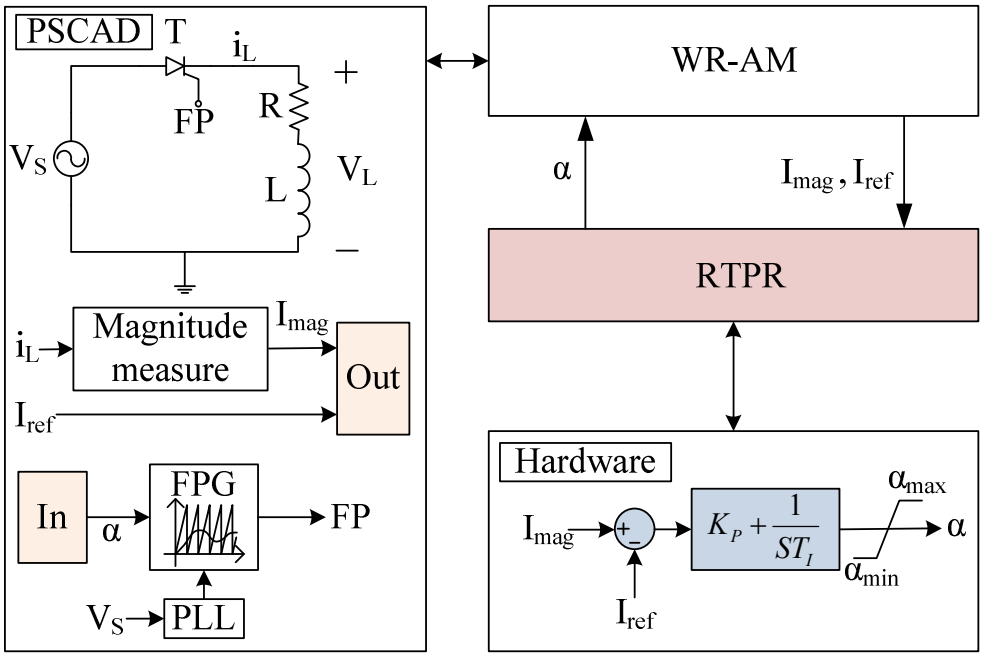


Figure 5-17. PI controller in-the-loop with simulated rectifier using WR-AM.

The current controller is implemented in physical hardware and the remainder of the circuit is simulated in PSCAD/EMTDC. The controller outputs firing angle order (α) in response to the current error, i.e. the difference between ordered (I_{ref}) and measured (I_{mag}) currents. The output α only stabilizes when the error goes to zero, i.e., when the ordered and measured currents are the same. A Phase Lock Loop (PLL) is used for synchronizing

with the ac waveform and hence giving the primary timing reference. A firing pulse generator issues firing pulses corresponding to the firing angle order, i.e. at the instant when the angle (ωt) reaches α, where ω is the angular frequency of the supply. The circuit parameters are: V_S=240 kV, 60 Hz, R=0.1Ω, L=0.1H, ω = 2∗π∗60 Hz = 376.991 r/s). The controller gains are K_P=1.4, T_I=0.07.

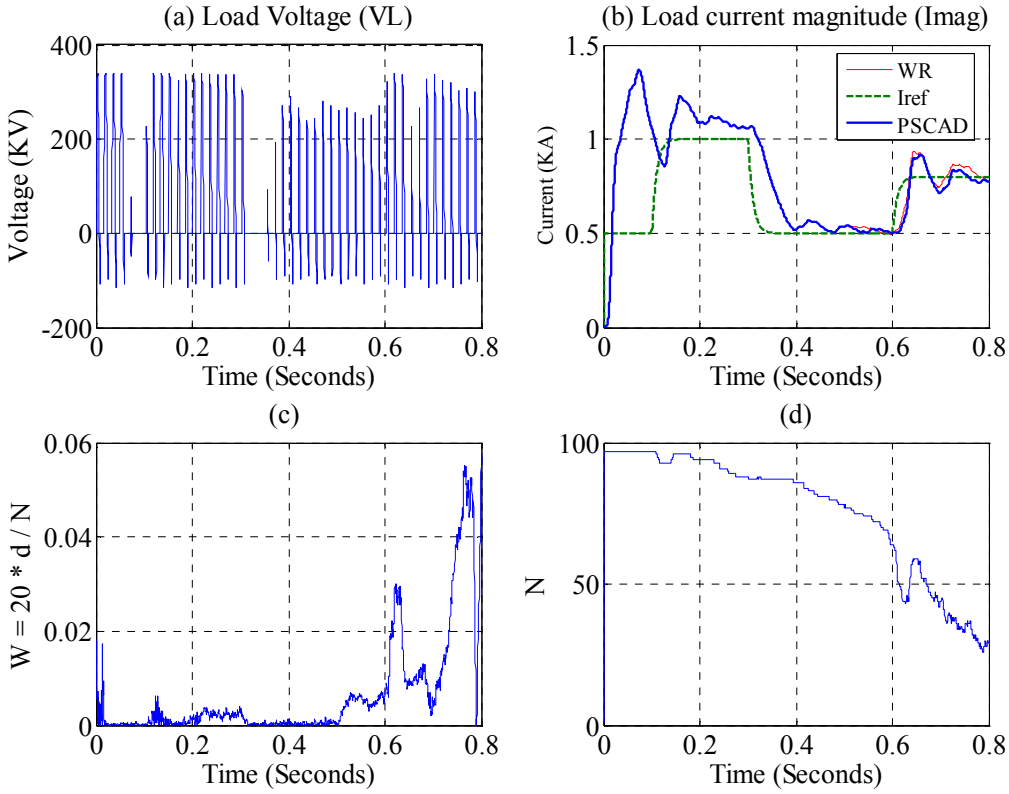


Figure 5-18. WR-AM converged results for load voltage and current, W, and N.

It was found that the classical WR did not converge, probably due to noise and other errors. With WR-AM using performance parameters L=10, K=20, and β=0.1 (see (5.9)-(5.12)), it converged in 85 iterations. The converged results for the load voltage and current as well as the WR-AM performance waveforms of W and N are displayed in Figure 5-18. From Figure 5-18 (c), a small value of W in the interval [0s, 0.6s] (i.e. a large value

for N in Figure 5-18 (d)) ensures that the waveforms in this interval will not be affected by noise while the remaining part of the waveform, i.e. [0.6s, 0.8s], is converging.

The PSCAD/EMTDC simulation result for I_{mag} is compared with that of WR-AM in Figure 5-18 (b) and the close results indicate that the WR-AM method improves WR convergence in the presence of noise. Note that no attempt has been made here to optimize the circuit's performance, and the tracking of current could be improved with better selection of controller parameters.

5.3 Parametric Analysis of WR Convergence Techniques

Here, a simulation of the WR-HIL convergence methods discussed above is conducted. The purpose of "simulating a simulation" is that one can add controlled amounts of noise, etc. to the simulation and conduct parametric studies. This allows for a more precise comparison. If the WR-HIL simulation had been conducted, there was no way to add precise amounts of noise and to make sure that the methods being investigated each experienced the same external conditions. The various acceleration techniques discussed above are now compared using the simple closed loop test system of Figure 5-19 (a). Figure 5-19 (b) shows the output response (z) of this system when K_P=100, T_I=0.01, A=7, and B=10.

The system of Figure 5-19 (a) is simulated using WR technique by the two blocks in Figure 5-20. Here, the HUT is also simulated so that noise may be introduced in a con-

trolled manner. Simulated noise (D_1 and D_2) is injected into the exchanged waveforms at input and output (x and y) ends respectively.

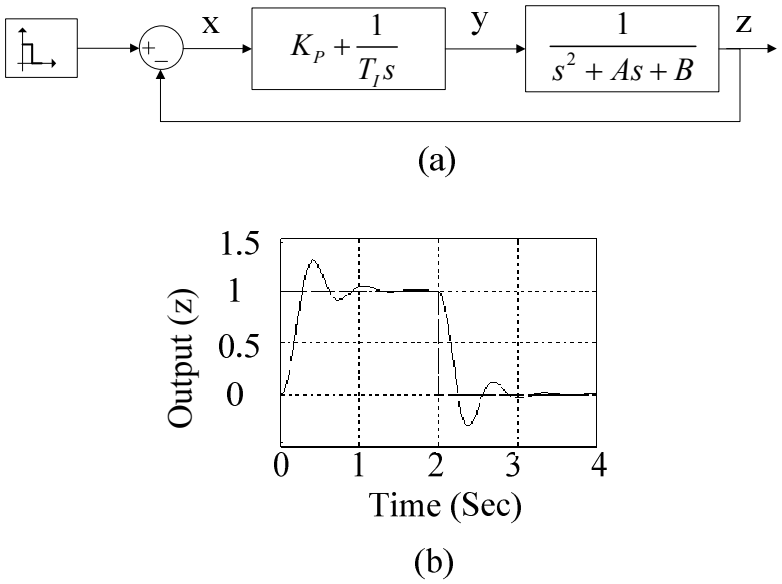


Figure 5-19. A simple closed loop test system.

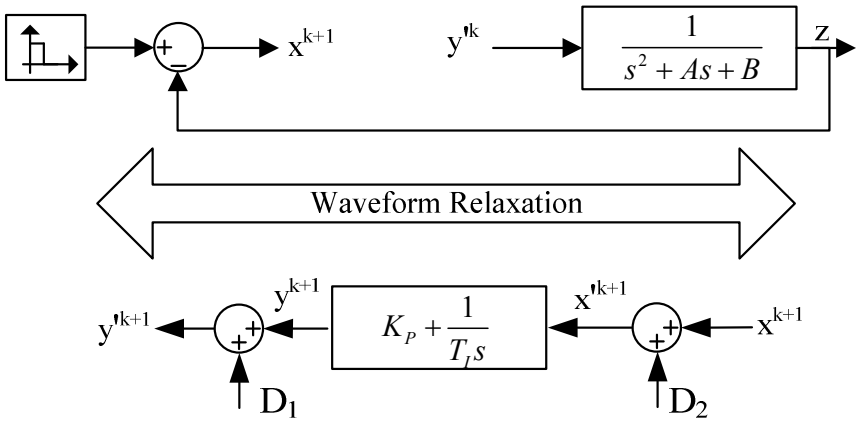


Figure 5-20. Splitting system of Figure 5-19(a) in two subsystems for application of WR.

In the parametric analysis below, WR simulations are conducted with noise D_1 (injected as a Gaussian noise with the Matlab random number generation function) varying from 5e-6 to 1.0 and with D_2 =0.2 D_1. The parametric analysis is carried out for: a) each acceleration method (WR-PWFC, WR-ICM1, WR-AM), applied individually, b) two

methods are combined (e.g. WR-PWFC & WR-AM); and c) all three methods applied simultaneously.

Figure 5-21 shows the number of WR iterations required for each case as a function of the noise magnitude (D_1). The classical WR converged in 48 iterations only when no noise was injected, and diverged with even the smallest amount of noise, e.g. 5e-6.

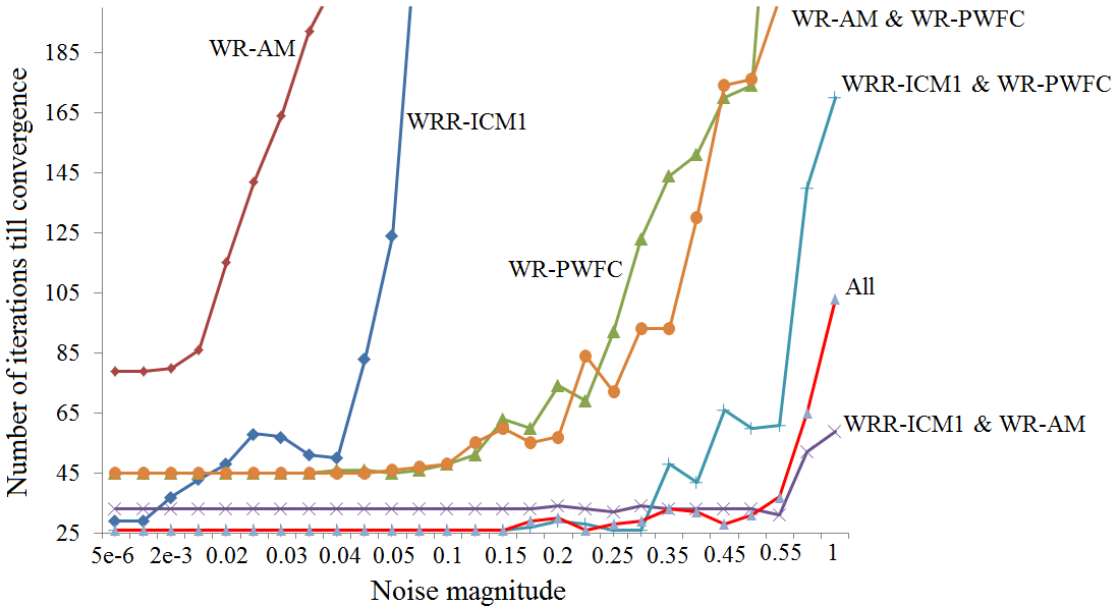


Figure 5-21. Number of iterations required for convergence as a function of noise for different WR convergence techniques, individually and in combination.

Based on the results shown in Figure 5-21, the following conclusions can be made:

With the methods applied individually, WR-AM did improve convergence in the presence of noise, but showed the poorest performance (largest number of iterations) as noise increased. WR-ICM1 had better performance, but WR-PWFC had the best performance, with iteration count of under 50 with noise of 0.1 pu.

Combination of any two techniques proved to be more effective than individual application of each technique.

However, the most superior performance was with all three methods applied simultaneously, where the iteration count remained under 30 for noise up to 0.5 pu.

One may argue that although the use of all three methods simultaneously takes fewer iterations for the WR convergence, the computational burden may still be high when all three methods are simultaneously applied as each iteration may take longer to simulate. However communication of the waveforms over the internet is the slowest bottleneck with the technology being used here, and so reducing the iteration count was the most effective way to speed up the process.

5.4 WR-HIL Technique for Relay Testing

As was explained in Section 4.3.4, the relays remain latched after a fault is detected. This interferes with the WR iterations as the relay must be reset to the same initial conditions at the start of every WR iteration. Fortunately, most modern protective relay manufacturers allow the user to configure the relay to unlatch either manually or automatically. For instance, the SEL-421 relay from Schweitzer Engineering Laboratories has a relay logic bit called "*ULTR* (Unlatch Trip)", which sets the relay unlatch to manual if set to "0" and automatic if set to "1". If the latter is chosen, then the relay logic word "*TDUR* (Trip Duration)" can be set to the desired time duration that the relay remains tripped. For example, *ULTR*=1 and *TDUR*=20 specifies that the relay trips when a fault is detected and remains tripped for 20 cycles (20/60=0.33 seconds for a 60 Hz system) and eventually automatically unlatches to normal conditions. This setting of the relay is used to facilitate the relay hardware to be tested using the WR method without the user interactions.

5.4.1 Experimental example

Figure 5-22 shows the example system [56] to investigate the WR based test of a commercially available protective relay (SEL-421). A typical 60 Hz power system is modeled in the off-line PSCAD/EMTDC simulation program. A three phase to ground fault is applied to the transmission line as shown in Figure 5-22.

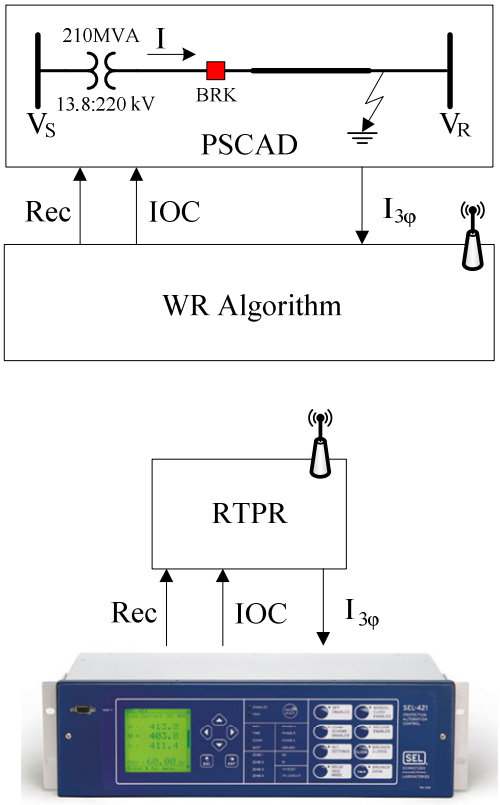


Figure 5-22. Testing an instantaneous overcurrent (IOC) relay on SEL-421 platform with the reclosing (Rec) function using the WR-HIL method [56].

The line three-phase currents ($I_{3\varphi}$) are sampled and recorded to be played back in real-time by the RTPR device connected to the SEL-421. The RTPR device samples and records two output waveforms of the relay. As shown in Figure 5-22, the first output is the instantaneous overcurrent trip signal (*IOC*), and the other waveform is the reclosing pulse

(*Rec*) issued by the relay. These two pulses are applied to the logic circuit of the PSCAD/EMTDC simulation breaker (Figure 5-22).

The SEL-421's phase instantaneous overcurrent bit (50P1) is used as the output *IOC* of Figure 5-22. The pickup current level, i.e. the current level at which the relay issues a trip signal, is set to 1.47 A. In the experiment, current amplifiers were not explicitly used and their function was included in the simulated power system model. Hence, the three phase line currents ($I_{3\varphi}$) measured from the PSCAD/EMTDC simulation were multiplied by the relays' Current Transformer (CT) ratio and applied to the relay on the secondary side of the CT. An example of the *IOC* trip signal is shown in Figure 5-23 with a fault occurring at t=0.1 seconds. The *IOC* pulse is high for normal conditions and low when a fault occurs. A high-to-low transition of the *IOC* signal opens the breaker of the system example of Figure 5-22, by setting the value of the breaker open/close signal (*BRK*) to 0.

The SEL-421 reclosing function (*Rec* in Figure 5-22) is set up to issue three closing attempts with a 4 cycle time interval, i.e. 4/60=0.066 seconds, between each attempt (see Figure 5-23). As can be seen in Figure 5-23, the relay waits *TDUR*=0.33 seconds after the fault occurrence and goes back to the normal conditions (as explained, the relay setting *ULTR*=1 and *TDUR*=20 cycles, avoids the user manually resetting the relay). These settings can be clearly seen in the example of Figure 5-23. A low-to-high transition of the *Rec* signal closes the breaker of the system example of Figure 5-22.

The relay's two output pulses of *Rec* and *IOC* are used to control the breaker of the system in Figure 5-22. For example, in Figure 5-23, the breaker is closed and a fault occurs at t=0.1 seconds. The *IOC* signal changes from high to low and therefore *BRK* signal is set to "high" (which opens the breaker). The first reclosing attempt is issued 4 cycles

after the fault time (t=0.1 seconds). A low-to-high transition of the *Rec* signal resets the *BRK* trip signal to zero (which closes the breaker). As it can be seen from Figure 5-23, the *IOC* remains high for the rest of the simulation which indicates the fault has been cleared before the breaker is closed and hence, *BRK* remains low (breaker closed).

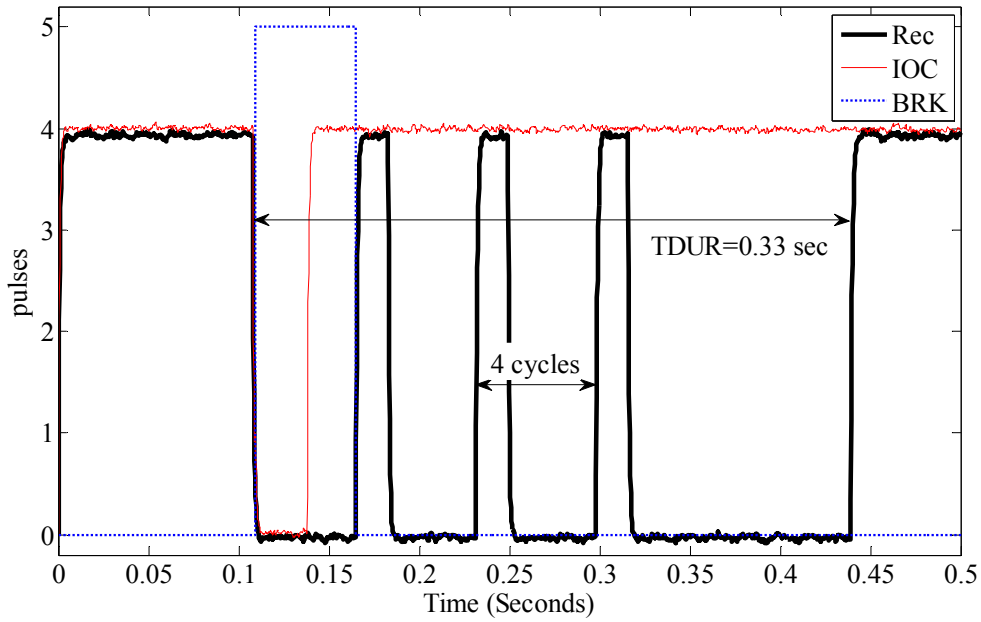


Figure 5-23. An example of different trip signals used for testing the reclosing function of SEL-421 [56].

The Waveform Relaxation algorithm for testing the relay hardware is used below to find the closed loop response of the relay hardware and the PSCAD/EMTDC power system simulation of Figure 5-22 for different durations of the fault ($T_{f\text{-}dur}$). In order to validate the obtained closed loop results, the power system model of Figure 5-22 is also simulated in real-time using an RTDS platform and the relay is directly connected to the RTDS simulation and tested for each case for comparison, as shown later.

It is noted that the breaker controlled by *BRK* signal opens only when the instantaneous line current is zero (same for each phase) and also the fault clears only when the in-

stantaneous fault current is zero for each phase. The total simulation time is T=0.5 seconds and the simulation time-step is 50 microseconds.

In the next sections, the WR-HIL simulation is conducted for different fault durations.

A. Fault duration $T_{f\text{-}dur}$=0.05 seconds

As in the waveform relaxation algorithm, the PSCAD/EMTDC power system model is simulated for 0.5 seconds with all trip signals (*IOC* and *Rec*) set to zero. Therefore, the breaker remains closed for the entire simulation time. Figure 5-24 shows the three phase currents in the transmission line (sending end). The three phase to ground fault occurs at t=0.1 seconds and therefore the magnitudes of the three phase currents increase. The fault is removed at t=0.15 seconds and the line currents' magnitudes recede to the rated values.

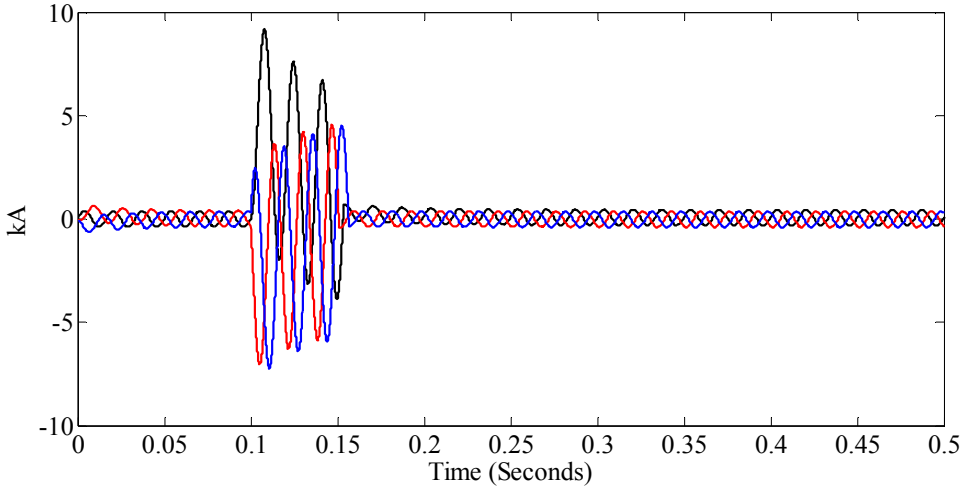


Figure 5-24. Line's three phase currents: the initialization of the WR-HIL with simulating PSCAD/EMTDC power system with relay pulses *IOC* and *Rec* set to zero [56].

In the first iteration of the WR, the recorded line currents of Figure 5-24 are sent to the RTPR. The RTPR plays back the currents samples in real-time to the SEL-421 relay and records the two relay output response waveforms *IOC* and Rec for T=0.5 seconds with a sampling time of 50 microseconds. The recorded relay output waveforms are made

available to the PSCAD/EMTDC simulation and the power system is once again simulated for T=0.5 seconds. The waveforms of the first iteration of the WR are shown in Figure 5-25. From Figure 5-25, both *IOC* and *Rec* signals reset to zero when the fault occurs (t=0.1 seconds). High-to-low transition of the *IOC* pulse sets *BRK* to high and therefore the breaker opens. The *IOC* signal remains low for a relatively long time because the currents waveforms of Figure 5-24 are off-line and do not change on-line even though the relay issues a trip signal. The PSCAD/EMTDC breaker is closed when the *Rec* signal shows a low-to-high transition. After the breaker is closed, it can be seen from Figure 5-25 that the currents have rated magnitudes which shows that the fault was cleared before the breaker was closed. This is because the fault duration ($T_{f\text{-}dur}$=0.05 seconds) is shorter than the reclosing delay of 4 cycles (T_{rec}=0.066 seconds).

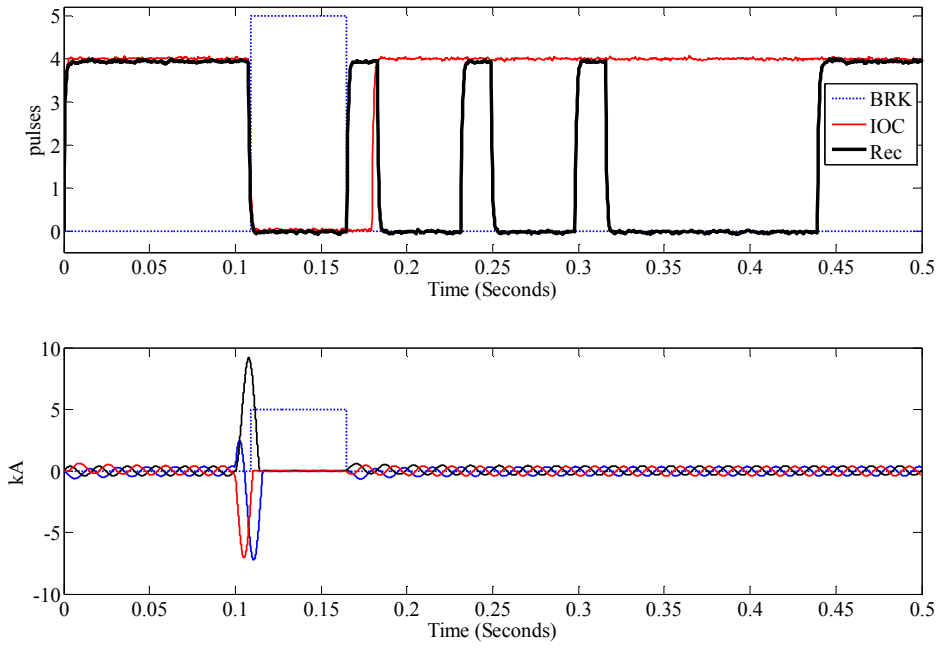


Figure 5-25. Iteration 1 of the WR-HIL results for the line currents and SEL-421 output signals [56].

The WR convergence criterion is checked by comparing the resulting transmission line currents of the initial and the first iterations of the WR. Because the waveforms are dif-

ferent, the second iteration of WR is started with the playback of the line current waveforms of Figure 5-25 to the relay. Again, the relay's output signals are recorded and used to simulate the PSCAD/EMTDC model of the power system as in the second WR iteration. The results of the second WR iteration are shown in Figure 5-26.

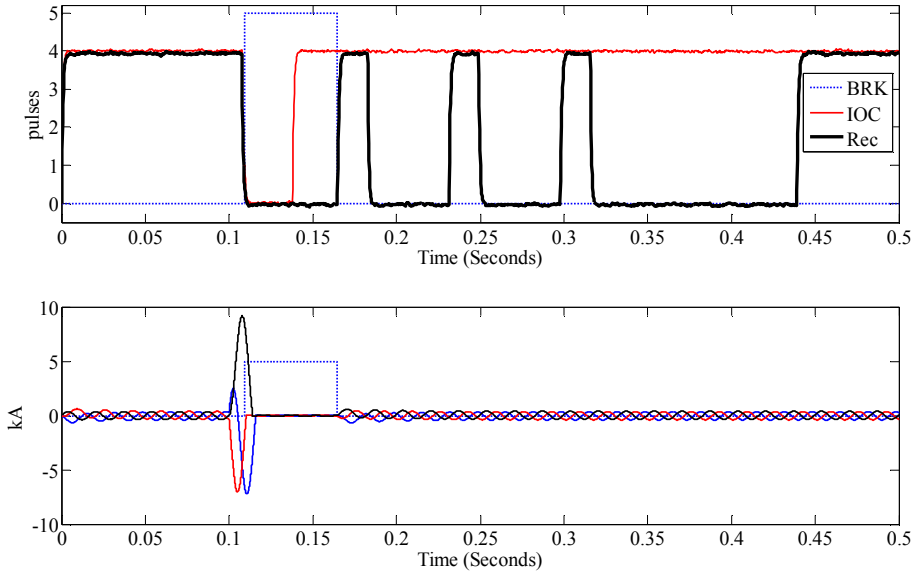


Figure 5-26. Iteration 2 of the WR-HIL results for the line currents and SEL-421 output signals [56].

Comparing Figure 5-25 and Figure 5-26, the only difference is the low-to-high transition time of the *IOC* waveform which does not affect the breaker operation in the PSCAD/EMTDC simulation. The reason for the faster changing of the *IOC* signal from low to high is that the current magnitudes of Figure 5-25 are changed to zero due to the breaker opening in the first WR iteration. Therefore, the PSCAD/EMTDC simulation results remain the same as the low-to-high transition of the *IOC* does not affect the breaker open/close signal (*BRK*). It was found that none of the relay output signals and the PSCAD/EMTDC simulation results change any more with further exchange of the waveforms. As a result, the WR is deemed to have converged to the closed loop responses of Figure 5-26. The converged results are compared with those obtained from a convention-

al RTDS platform in Figure 5-27. The results compare well, which shows the high accuracy of the approach in finding the closed loop response of the two subsystems (PSCAD/EMTDC simulation of the power system model and the SEL-421 relay hardware). Due to some internal logic in RTDS, there is a delay between the breaker close signal (*BRK* being low) and the actual closing of the breaker; hence, there is a delay in the resumption of the current in the RTDS simulation.

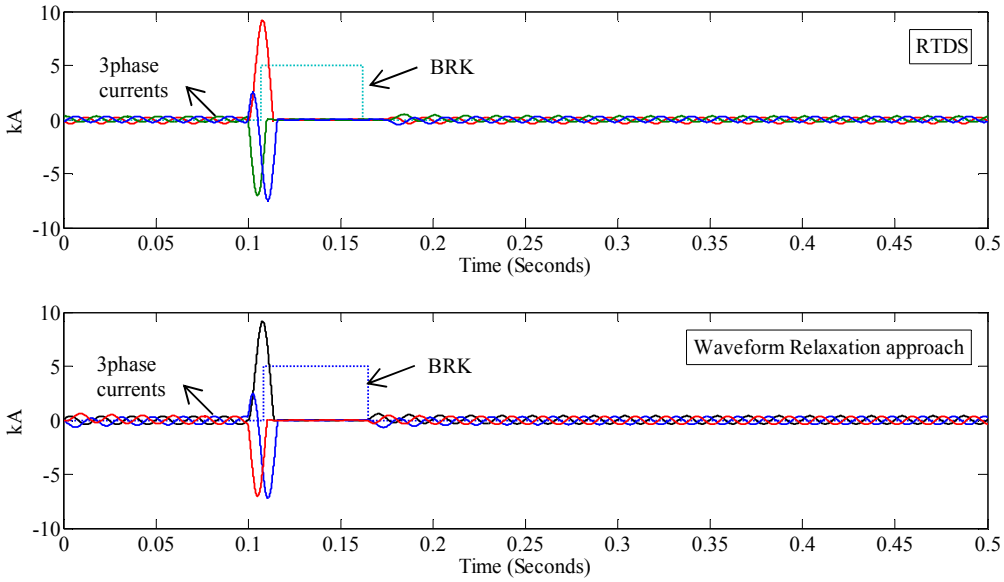


Figure 5-27. Comparison of the WR-HIL converged results and the RTDS closed loop response for the example of $T_{f\text{-}dur}$=0.05 seconds [56].

B. Fault duration $T_{f\text{-}dur}$=0.1 seconds

Figure 5-28 displays the initial PSCAD/EMTDC simulation result for the three phase line currents with an initial zero waveform as the relay response pulses for a fault duration of 0.1 seconds. In the first WR iteration, the current waveforms of Figure 5-28 resulting from the initial PSCAD/EMTDC simulation of the power system is played back in real-time to the relay hardware. The relay detects the fault at t=0.1 seconds and issues a trip signal with setting the *IOC* output from high to low and generating three reclosing

pulses at the Rec output as explained before. The *IOC* remains low until the current magnitudes of the waveforms of Figure 5-28 reduce to less than the pickup current. The first reclosing attempt occurs at T_{rec}=0.066 seconds after the fault occurrence which precedes the low-to-high transition of the *IOC* signal (more than 0.1 seconds after the fault occurrence). This is why the PSCAD/EMTDC simulation result in the first WR iteration for the transmission line currents in Figure 5-29-a shows that the breaker is closed before the fault is cleared. Of course this is not the converged result and would be fixed in next WR iterations as can be seen in Figure 5-29.

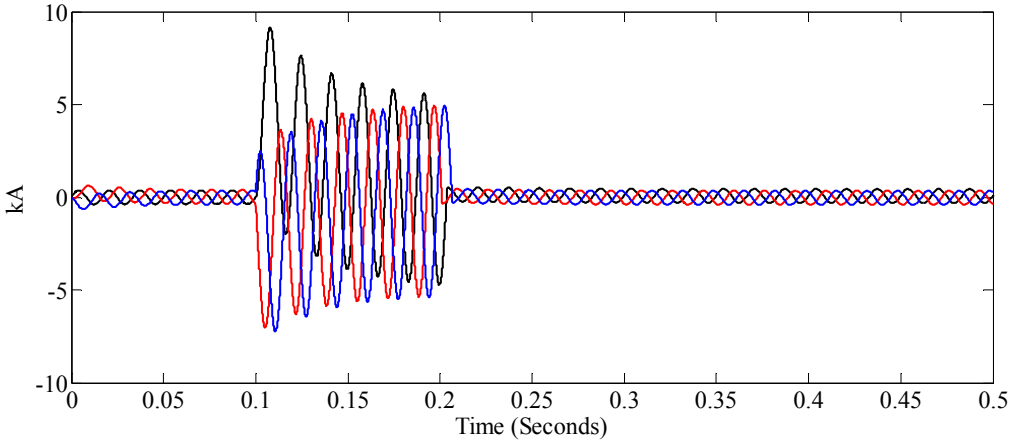


Figure 5-28. Line's three phase currents: the initialization of the WR-HIL with simulating PSCAD/EMTDC power system with relay pulses *IOC* and *Rec* set to zero [56].

In the second WR iteration, the PSCAD/EMTDC simulation result for the line currents from the first WR iteration are played back to the relay. In this iteration, the first low-to-high transition of the *IOC* signal occurs sooner than that of the first iteration because of the breaker operation in the first iteration. The phase overcurrent function of the relay detects that the fault still remains after the first reclosing attempt and issues a high-to-low transition in the *IOC* signal as seen in Figure 5-29-b. From Figure 5-29-b, the relay outputs the same *Rec* pulse in all WR iterations because the fault time remains the

same. The relay, however, changes the *IOC* signal according to the breaker operation in the simulation and hence changes in the measured line currents in the previous WR iteration. This process continues until convergence. On convergence, exchange of the waveforms and further simulation of the PSCAD/EMTDC power system and playback of the waveforms to the relay result in the same waveforms at the outputs. The converged waveforms are identical to the closed loop response of the two simulation and hardware subsystems. This can be found from the comparison of the WR converged results with the RTDS real-time closed loop simulation results in Figure 5-29-d.

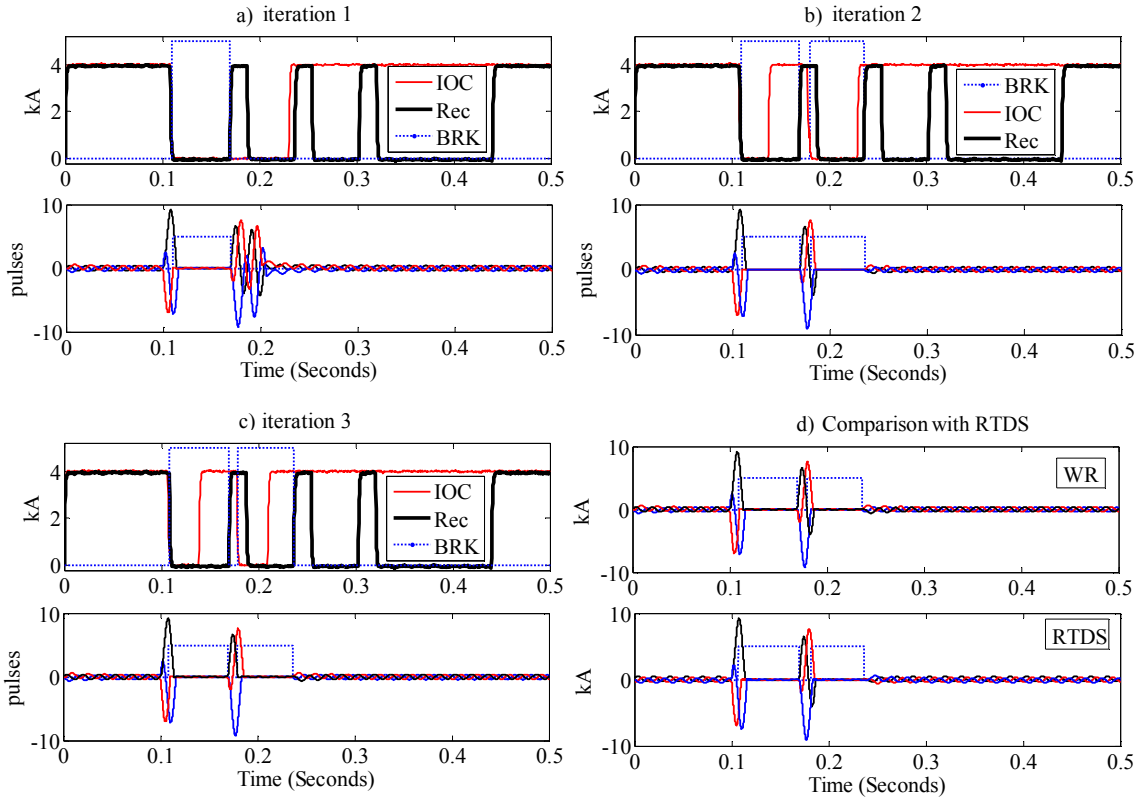


Figure 5-29. The PSCAD/EMTDC simulation and SEL-421 relay responses for all WR-HIL iterations and comparison with RTDS closed loop simulation results [56].

C. Fault duration $T_{f\text{-}dur}$=0.25 seconds

In this experiment, the simulated fault is applied for 0.25 seconds and the WR algorithm is used to iteratively find the closed loop response of the SEL-421 relay hardware and the PSCAD/EMTDC simulation of the power system. Initially, the PSCAD/EMTDC simulation result for the line currents with the zero waveforms as the relay pulses to the simulation is shown in Figure 5-30.

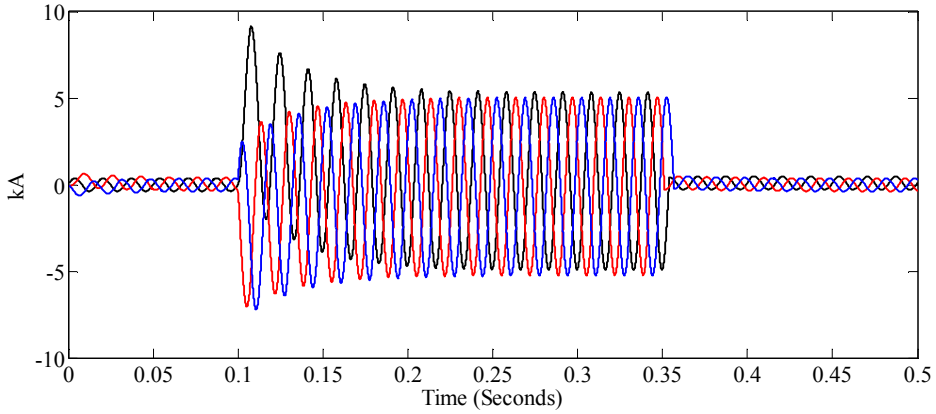


Figure 5-30. Line's three phase currents: the initialization of the WR-HIL with simulating PSCAD/EMTDC power system with relay pulses *IOC* and *Rec* set to zero [56].

Figure 5-31 presents the WR results for the PSCAD/EMTDC simulation and the relay output waveforms for every WR iteration. Similar to the other cases described earlier, the SEL-421 relay changes its *IOC* output signal according to the PSCAD/EMTDC simulation response for the line currents in every WR iteration. In this example, the fault duration is longer than all three reclosing attempts of the relay. In other words, the relay eventually issues a permanent trip signal by resetting the *IOC* signal to zero after the third unsuccessful reclosing attempt as can be seen in the last iteration of the WR in Figure 5-31-e. The relay sets the *IOC* output to high at t=0.433 seconds. This is not because of the failure of the relay to remain tripped for the detected fault but, as described earlier, the re-

lay automatically unlatches the trip because *ULTR*=1 and *TDUR*=20 cycles (0.333 seconds). Hence, the relay unlatches the trip (*IOC* signal) at $t=T_{fault}+TDUR$=0.1+0.333=0.433 seconds.

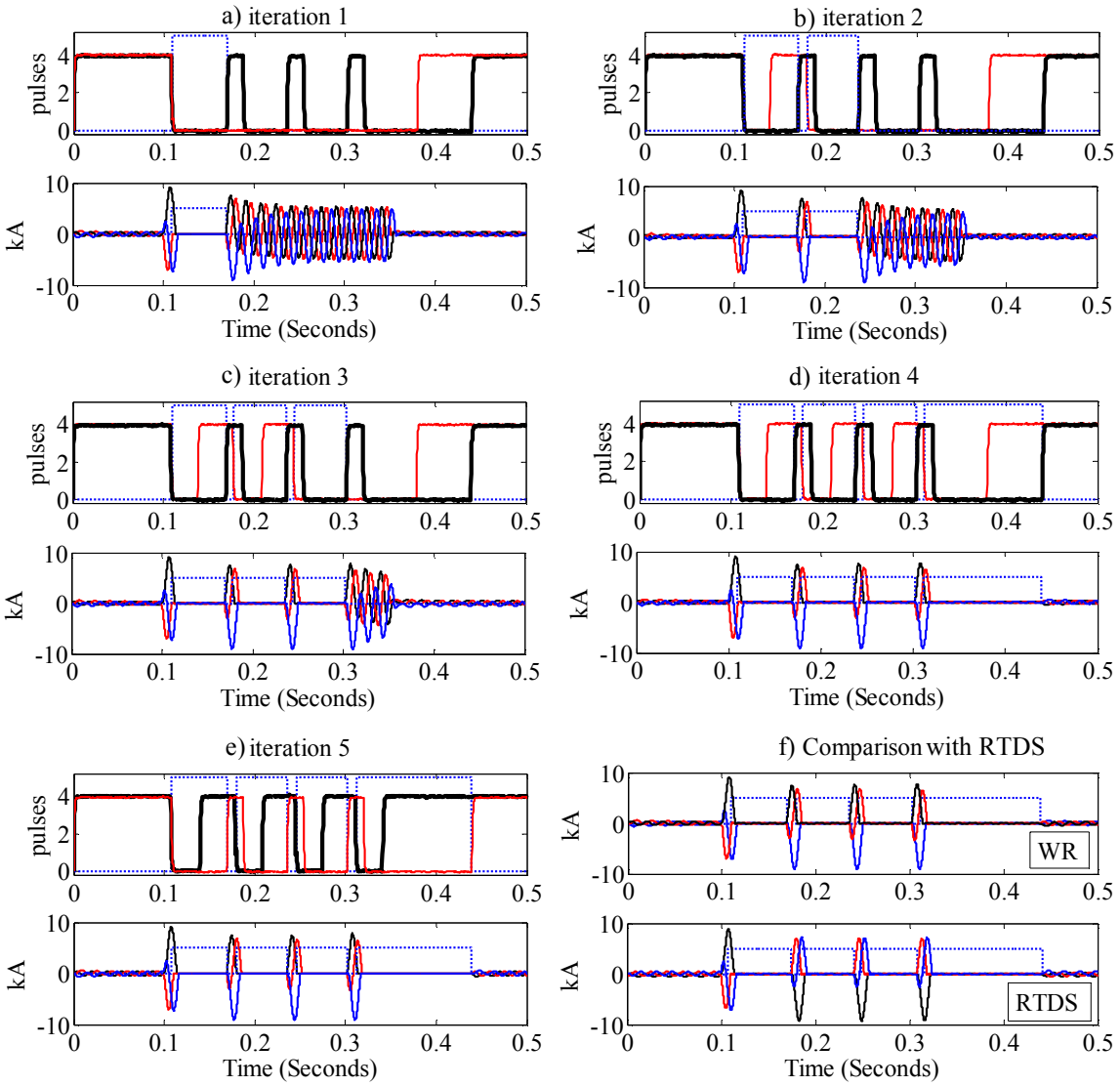


Figure 5-31. The PSCAD/EMTDC simulation and SEL-421 relay responses for all WR-HIL iterations and comparison with RTDS closed loop simulation results [56].

The WR converged results are identical to the closed loop simulation of the power system simulated in real-time using the RTDS platform and the SEL-421 relay as shown in Figure 5-31-f.

5.5 Concluding Remarks

It is shown that, like the classical HIL simulators, WR can also face instability when applied to HIL simulation with controllers and power systems. In WR-PHIL, the impedance ratio between the simulation subsystem and the HUT mainly determines whether the simulation is stable. A stabilization technique was proposed in this chapter, where a parallel resistor (usually small) is connected at the interface node of the simulation subsystem so that the equivalent impedance of the subsystem is small in early iterations (assuming the voltage type representation of the Ideal Transformer Model of the interface method is used). The current passing through the resistor is recorded in every iteration and the difference with that of a current iteration is fed back into the system. Therefore, as the WR simulation iteration count increases, i.e., close to convergence, the injected damping current waveform becomes smaller. The method showed success in stabilizing the WR simulation of the transformer in the loop example.

It was also shown that although classical WR works well with purely software modules, it cannot converge for HIL applications due to noise and other interfacing errors that accrue when one module is an actual hardware component. This chapter introduced three new modifications to classical WR (WR-PWFC, WR-ICM, and WR-AM) for accelerating convergence. These are respectively based on limiting the convergence window size,

accelerating convergence by including an approximate model of the HUT in the EMT simulation, and waveform averaging. Several illustrative examples of FACTS and HVDC systems with controllers implemented in hardware demonstrate the efficacy of the approach.

Using a simple test system modelled in software, where noise could be added in a controlled manner, it was shown that the highest convergence rate was achieved by applying all three techniques simultaneously.

The output contacts of the commercially available protective relays remain latched until a button is manually pressed by the operator after detection of a fault. This feature of the relay does not allow the iterative solution of the WR to be used in testing the relay hardware. However, in most modern commercial relays, the relay has a feature that permits automatic unlatching after a defined number of cycles. This option in the relays settings can be used to facilitate test of the relay using the WR method. In this chapter, an experimental example was presented where a SEL-421 overcurrent relay from Schweitzer with the reclosing feature was set to automatically unlatch after detection of a trip. The WR based simulation results for testing the SEL-421 relay within the PSCAD/EMTDC power system simulation showed a good comparison with the RTDS closed loop simulation results, and confirmed the accuracy of the approach.

Chapter 6

Controller Optimization with WR-HIL

In this chapter, the WR-HIL simulation technique is extended to include optimization algorithms for tuning the parameters of actual power system controllers. In the following sections, the algorithm is described and two experimental examples are given. In one example, the proportional and integral gains (PI) of the rectifier and inverter controllers, implemented in hardware, are optimized within the HVDC system model simulated in the off-line PSCAD/EMTDC. In the second example, the proportional-integral (PI) gains of a typical Static VAR Compensator (STATCOM) controller hardware are optimized for a defined objective function.

6.1 Power System Controller Optimization

One of the principal reasons for using HIL simulation is that selecting the parameters of an actual controller based on the conceptual model of the controller is often inaccurate [55]. The discrepancies between a purely software model and the actually implemented final controller design may result in different optimization results. The interface gain and

phase errors, amplifier errors, limited interface hardware bandwidth, interface filters, quantization errors in the interface's analog-digital converters, noise, interface DC off-sets, minimum and maximum voltage and current levels in the interface hardware, etc. can sometimes significantly change the optimal controller parameters depending on the system under study compared to the possible optimal parameters found with the computer software model of the controller [57]. On the other hand, there is possibility that a controller manufacturer may not disclose certain hidden internal features of the controller [57]. The optimization of actual power system controllers with the traditional real-time (digital) simulators has been studied in [57] where both the controller and the power system model are conducted in real-time.

In this section, the proposed WR-HIL approach is augmented with an optimization algorithm, so that the process of selecting the optimal controller parameters can be automated. In this scheme, a non-negative objective function (OF) is developed that penalizes any deviation from the desired objective for the system, with a penalty that increases with the difference. Hence an OF close to zero means that the objectives are largely met, and a large OF means that the performance is poor. A non-linear optimization method such as the Genetic Algorithm (GA) [58] or the Nelder-Mead downhill Simplex algorithm [59] then inspects the objective function results from earlier simulation runs and suggests new candidate parameters for the controller that are likely to improve the performance. Convergence is reached when the OFs from successive runs differ only by less than a predefined exit threshold.

The power system models are simulated in off-line EMT programs and the optimization toolbox from a general purpose software program such as Matlab is used for optimizing the parameters of the HIL controller.

6.2 WR-HIL Based Controllers Optimization

In this section the standard controller optimization algorithm is briefly described and then the proposed solution of WR based controllers optimization method is presented.

6.2.1 Basic optimization algorithm with offline (non-WR) simulation

Figure 6-1 displays the classical algorithm for the optimization of parameters of a controller modelled within an offline simulation. In this scheme, both the controller and the power system models are simulated in PSCAD/EMTDC [60]. The "Initialization" block in Figure 6-1 consists of required initial data for starting the optimization algorithm, e.g., the initial candidate points (if required), the algorithm stopping criteria, the OF definition, etc.

The OF corresponding to a set of candidate points, **p**, is calculated using the waveforms generated from the PSCAD/EMTDC simulation of the system model. If the optimization algorithm has not converged, another candidate point is generated and the power system is simulated. This process is iterated until the optimization algorithm is converged. In the next section, this algorithm is extended with the WR-HIL approach for the optimization of an actual controller hardware.

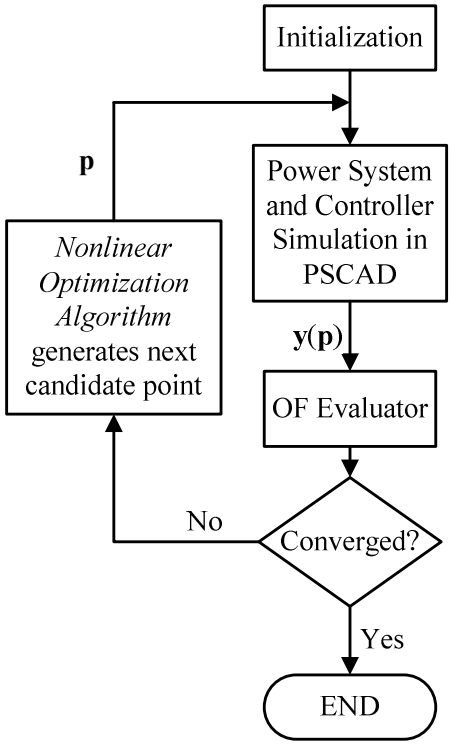


Figure 6-1. Basic optimization algorithm of a controller model within a power system modeled in PSCAD/EMTDC.

6.2.2 Proposed solution

The WR-HIL algorithm described is equipped with the classical optimization algorithm of Figure 6-1 in order to enable the optimization of an actual controller. Here, it is assumed that the parameters of the controller ($\mathbf{p}$) are tunable through the RTPR device. The proposed WR based optimization algorithm is shown in Figure 6-2. The RTPR sets the parameters of the actual controller to the new candidate point, $\mathbf{p}^{n+1}$, generated from the optimization algorithm. The WR-HIL iteratively finds the closed loop response of the simulated power system (PS) model and the actual controller corresponding to the candidate point $\mathbf{p}^{n+1}$. The optimization algorithm calculates the Objective Function,

OF($\mathbf{p}^{n+1}$), based on the waveforms resulted from WR-HIL simulation and generates the next candidate point. This process is continued until the optimization algorithm converges to the best candidate point.

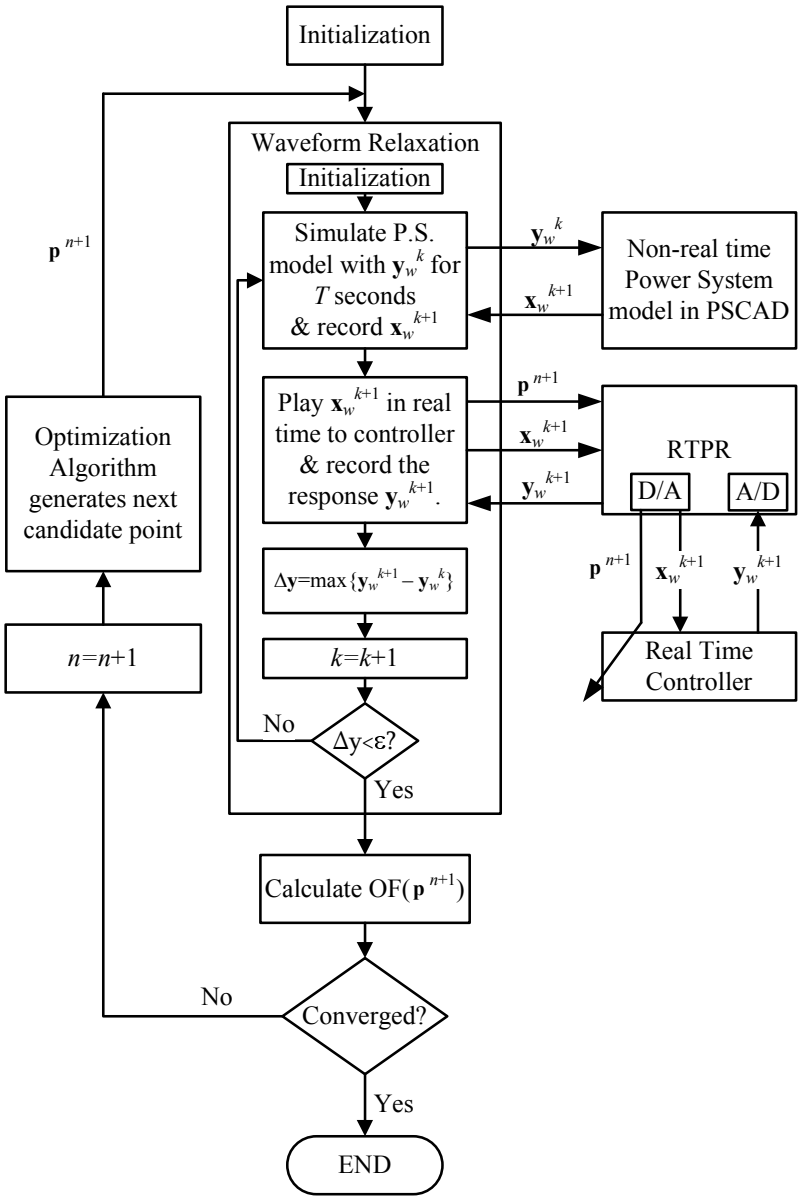


Figure 6-2. WR-HIL based optimization of an actual controller's parameters.

In Figure 6-2, the WR-HIL iterations are denoted by k, and n is used as the optimization iteration counter. Not only the optimization algorithm itself, but also the WR-HIL

algorithm must converge for every candidate point to ensure the WR based controller optimization algorithm converges. The accelerated convergence techniques proposed in Chapter 5 must be used in the WR-HIL scheme to ensure that the WR converges for every candidate point with the presence of the noise and other interface errors.

Figure 6-2 shows the optimization procedure for one controller hardware. The algorithm is obviously extendable to include simultaneous optimization of multiple geographically distributed controllers, e.g., the rectifier and inverter controllers of a HVDC system. The optimization method used in this work was the Nonlinear Simplex method of Nelder and Mead [59]. The method was selected as it converges rapidly for variables less than about 10 and does not require explicit derivatives [60]. It does not guarantee the global minimum solution, but always leads to an improved minimum which is sufficient in most cases.

6.3 Nonlinear Simplex Optimization Method

The Nelder-Mead downhill Simplex is a popular nonlinear local search method for minimization of a given OF based on fundamental geometric considerations [59, 60, 61]. The Simplex optimization method has been previously used in power systems simulation [60]. Supposing N is the number of optimization parameters, the simplex is defined as an N-dimensional geometric Figure formed by N+1 points called vertices. Therefore, for N parameters to be optimized, a Simplex algorithm should be provided with N+1 initial points to begin. The algorithm is briefly described below [61, 62].

The optimization algorithm evaluates the OF values corresponding to the initial vertices and orders the N+1 points from the best (the point with smallest OF) to worst (the point with largest OF). Consider a two variable optimization problem that gives a triangular simplex as shown in Figure 6.3. In the original simplex, one vertex (H) will have the highest OF, one (L) the lowest and one (M) an in between value (if there is a degenerate case, i.e. M and L yield exactly the same value, it does not matter which vertex is selected as M or L). In order to move towards the minimum value, the algorithm discards the highest value vertex by reflecting it through the midpoint of the remaining two vertices as shown in Figure 6.3.a. The OF at the reflected point (R) is evaluated. It is possible that the new vertex (R) has a value smaller than the best so far (i.e. L). This calls for an expansion as the reflected direction looks favourable. Thus a larger step is taken as shown in Figure 6.3, by multiplying the reflected length by a factor $\alpha > 1$. Occasionally, the OF at R may be in between L and M, so an outside contraction is indicated as shown in Figure 6.3.c, with the reflected length multiplied by a factor $\beta < 1$. If the vertex R has an OF even larger than at M, then a more severe (inside) contraction is carried out as in Figure 6.3.d.

If the value at the contracted point (IC or OC) is less than at the worst point (H), then the simplex dimension is shrunk by multiplying the lengths with the shrinking factor $\gamma < 1$ in the direction of the smallest point (L) as shown in Figure 6.3.e.

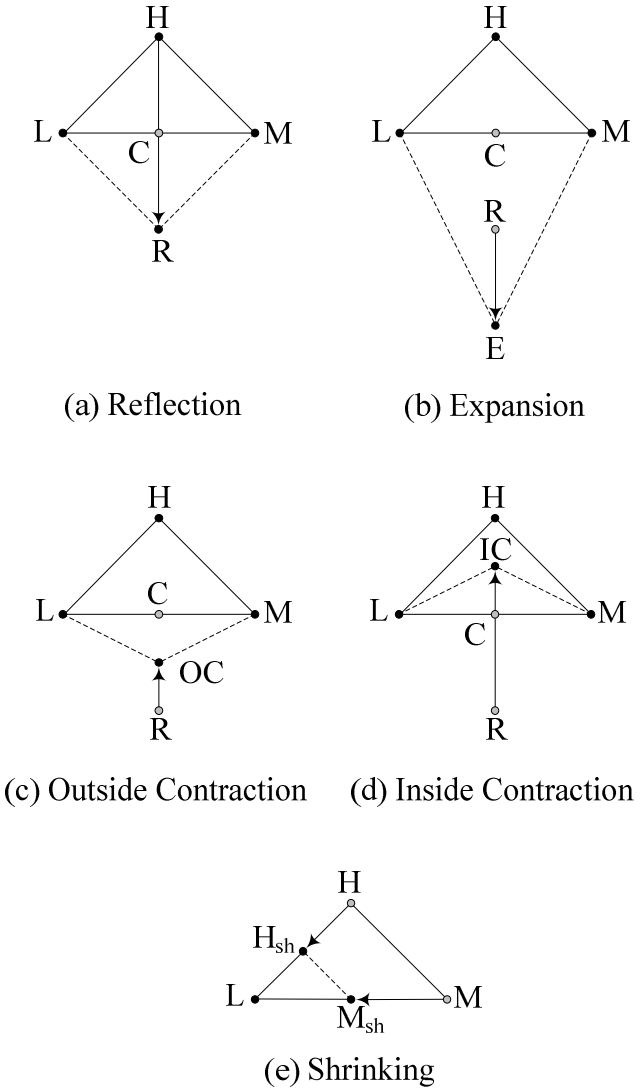


Figure 6-3. Geometric operations used in the Simplex optimization algorithm (OF(L) < OF (M) < OF(H)). a) Reflection: moving the worst vertex (H) towards the minimum point, b) Expansion: if OF(R) < OF(L), c) Outside contraction: if OF(L) < OF(R) < OF(M), d) Inside contraction: if OF(R) > OF(M), e) Shrinking: if OF (OC or IC) > OF(H).

The procedure can be generalized to any simplex dimension *N* by replacing the "mid-point" of the two smaller value vertices in the above discussion with the "centroid" of the smaller *N*-1 vertices.

6.4 Experimental Examples

Two WR based controller optimization examples are presented. In both examples, the nonlinear Nelder-Mead downhill Simplex optimization algorithm [59] written in Matlab in an m-file is used.

6.4.1 STATCOM PI controller tuning

The power system of Figure 6-4 with a Static Synchronous Compensator (STAT-COM) is modeled in PSCAD/EMTDC. The STATCOM controller [63], shown in Figure 6-5, is the HIL here. WR-based controller optimization algorithm proposed in Figure 6-2 is used to optimize the parameters of the PI controller so that the injected reactive current to the system follows the reference current as closely as possible.

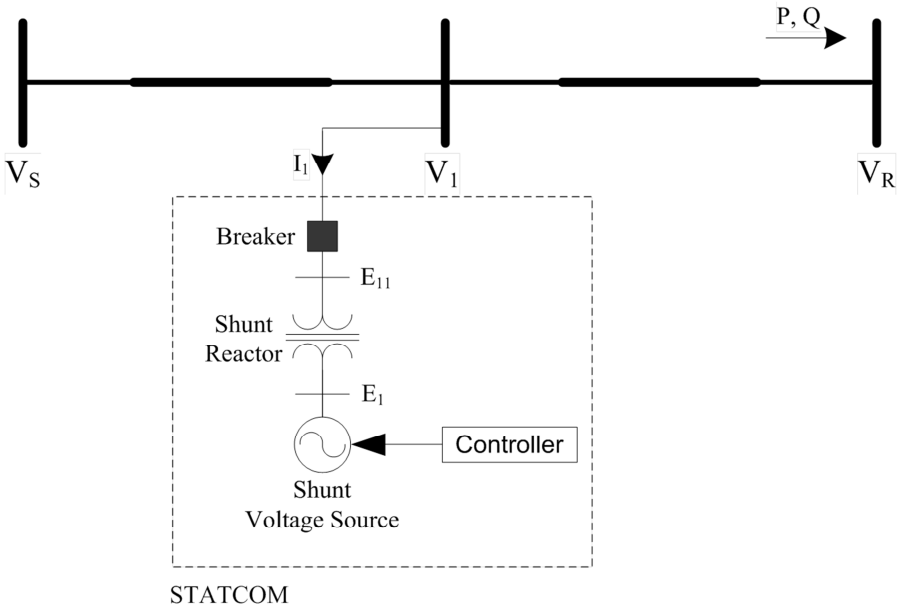


Figure 6-4. Power system and STATCOM modelled in PSCAD/EMTDC.

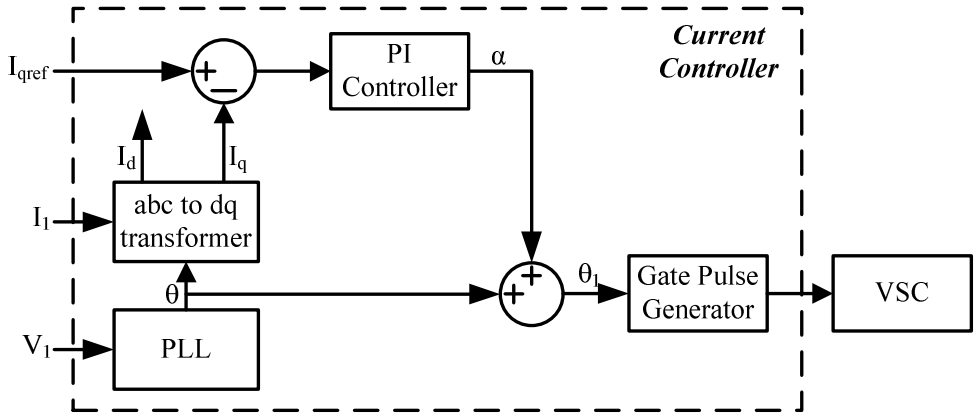


Figure 6-5. STATCOM current controller implemented in hardware.

The STATCOM breaker is closed at 0.05 seconds. The STATCOM reference current is set to be initially zero and then changed to +1 pu (capacitive compensation) at t=0.125 seconds and then changed to -1 pu (inductive compensation) at t=0.2 seconds. The total simulation time is T=0.3 seconds with the simulation time-step Δt=50 microseconds.

In order to ensure the convergence of WR-HIL, the WRR-M1 acceleration method proposed in Section 5.2.2 is used. In this method, an approximate model of the hardware under test is connected to the PSCAD/EMTDC power system model. A STATCOM controller with the proportional gain of K_P=1 and the integral time constant of T_I=0.04 is modelled in PSCAD/EMTDC as the WRR-M1's approximate model of the actual controller. The K_p and T_I gains of the actual STATCOM controller hardware are tunable by the RTPR device.

The optimization problem is to find the best PI controller parameters, e.g., K_p and T_I, so that the quadratic component (I_q) of the injected current from the STATCOM tracks the reference current (I_{qref}) changes closely with smallest overshoot and minimum steady state error and oscillations. The OF is formulated as an integral square error (ISE) as in

(6.1). ISE is a well suited function, often used for such minimization problems where a measured waveform is compared to a reference waveform [60].

For the ISE of (6.1), T_O and T_S define the optimization time interval, and $\mathbf{p}$ is the optimization parameter. Note that (6.1) penalizes any deviation over the interval between I_q and I_{qref}. Note that for a tracking problem I_{qref} can be a function of time.

$$ISE(\mathbf{p}) = \int_{T_o}^{T_S} \left(1 - \frac{I_q(\mathbf{p},t)}{I_{qref}(t)}\right)^2 dt; \quad \mathbf{p} = [K_P \;\; T_I]^T \tag{6.1}$$

Because the reference current waveform crosses zero at t=0.2 seconds, the OF is split into two integral functions before and after the zero crossing time as in (6.2) in order to avoid divide-by-zero error in the numerical calculation of the ISE. In this example, To=0.135 and Ts=0.275 seconds are used.

$$\begin{aligned} OF(\mathbf{p}) &= ISE(\mathbf{p}) = ISE_1(\mathbf{p}) + ISE_2(\mathbf{p}) \\ &= \int_{0.135}^{0.195} \left(1 - \frac{I_q(\mathbf{p},t)}{I_{qref}(t)}\right)^2 dt + \int_{0.21}^{0.275} \left(1 - \frac{I_q(\mathbf{p},t)}{I_{qref}(t)}\right)^2 dt \end{aligned} \tag{6.2}$$

The Simplex is started with three initial vertices as shown in Table 6-1. The stopping criterion for the Simplex algorithm is a tolerance of 0.01 in the OF. The Simplex converges in 33 iterations (each of which requires several WR-HIL iterations - about 25) with the results given in Table 6-1. The Simplex iteration number in this example is defined as the number of times that a new candidate point is generated. The OF at every Simplex iteration is shown in Figure 6-6. From Table 6-1, the OF is reduced from 21.84 (original case [63]) to 3.08 with the optimal parameters found by the Simplex algorithm.

The improvement in response can also be observed by comparing the pre-optimization (Original) and post-optimization (Best) waveforms in Figure 6-7.

Table 6-1. Initial guess parameters for Simplex algorithm and the optimization results.

Specification	K_P	T_I	OF
Initial guess	0.5	0.02	21.84
	1	0.08	5.66
	0.4	0.05	13.4519
Worst	2.1	0.17	30.43
Optimum	**1.72468**	**0.1566**	**3.08**

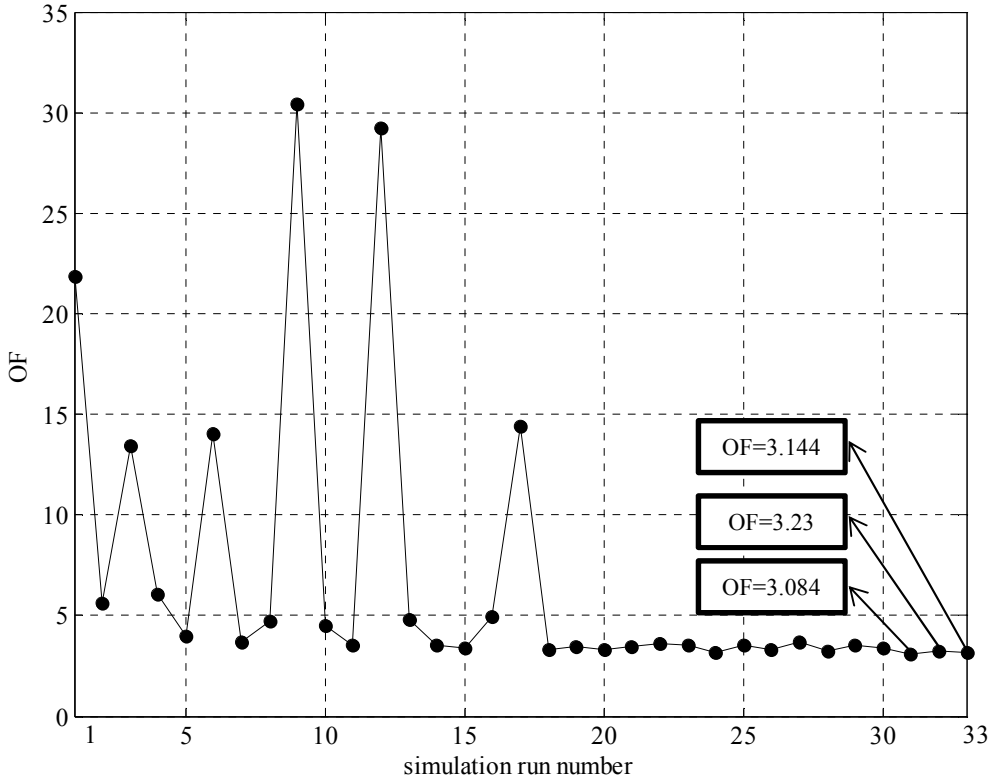


Figure 6-6. Variation of OF in every Simplex iteration.

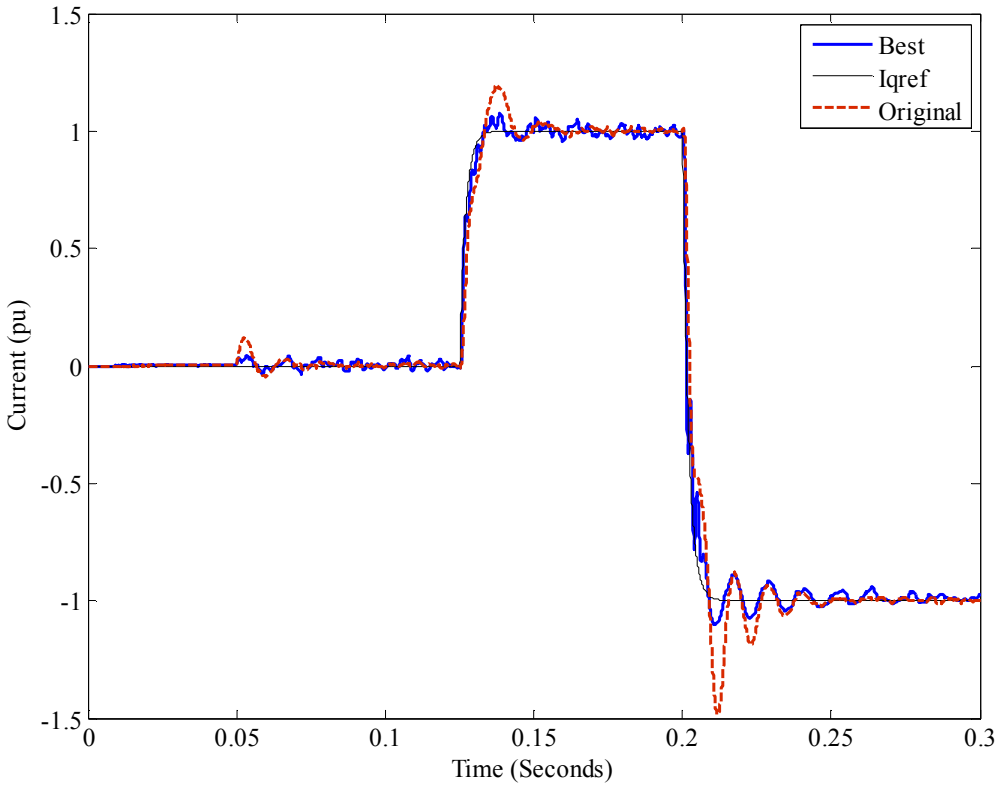


Figure 6-7. STATCOM injected reactive current for the optimum point and the original point [63].

From Figure 6-7, the overshoot in the current is reduced significantly with the cost of a few more oscillations with the optimum PI controller parameters. The elapsed time for the WR based optimization algorithm to converge was about 135 minutes.

6.4.2 HVDC controllers optimization

In this example, the performance of the proposed controller optimization method in Figure 6-2 is investigated for simultaneous optimization of the parameters of multiple controller components. The power system is an HVDC transmission system presented in Figure 6-8 with the data provided in Table 6-2.

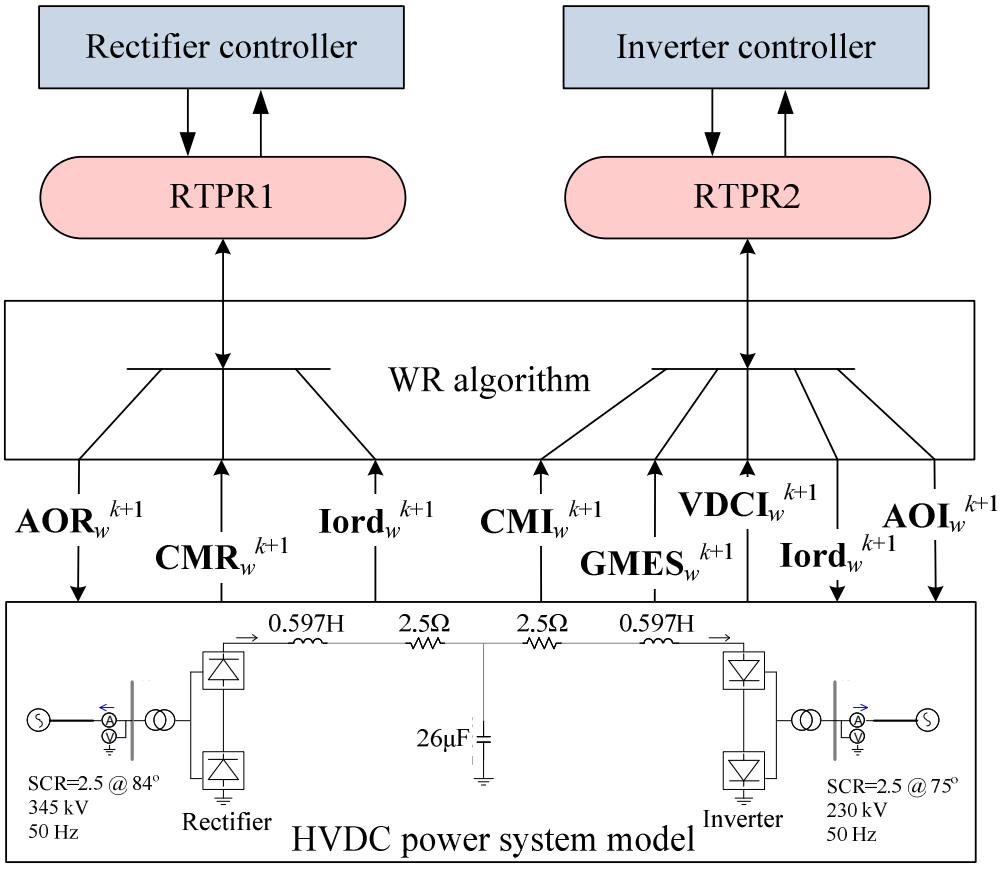


Figure 6-8. WR-HIL based optimization of HVDC controllers: AOI: inverter angle order, AOR: rectifier angle order, VDCI: inverter dc voltage, CMI: dc current measured at inverter side, CMR: dc current measured at rectifier side, GMES: Gamma angle measured at inverter, Iord: rectifier current order from inverter.

The HVDC system is modelled in the non-real-time PSCAD/EMTDC simulation program. The rectifier and inverter controller structures are from the first CIGRE HVDC benchmark [54] and are implemented in real-time hardware as shown in Figures 6-9 and 6-10. One RTPR device is connected to each controller and both RTPRs are connected via the internet to the simulator computer. Note that, as the data exchange is over the internet, it would be possible to have the rectifier and inverter controllers in remote locations. This would be the case if one were optimizing real controllers in an HVDC scheme, but did not want to transport the controllers outside the control room.

Table 6-2. 12 pulse HVDC CIGRE Benchmark [54] Model data.

Rectifier Side	
AC system	Transformers
373.3 kV, 50 Hz SCR=2.5 $\angle 84^o$	603.73 MVA 345/213.4557, 18%
Inverter Side	
AC system	Transformers
215.05 kV, 50 Hz SCR=2.5 $\angle 75^o$	591.79 MVA 209.2288/230, 18%

Filters and fixed capacitors (MVar) for both sides		
11-th harmonic	13-th harmonic	Fixed capacitors
250	250	125

DC link		
DC line resistance	Rated DC voltage on rectifier side	Rated DC current
5Ω	500 kV	2 kA

The rectifier and inverter controller diagrams are shown in Figure 6-9 and Figure 6-10, respectively. The measured dc current at the rectifier side and the inverter current order are the rectifier controller input waveforms and the response of the controller is the rectifier angle order. The measured waveforms of the inverter extinction angle as well as the dc voltage and current at inverter side are the inverter controller input waveforms.

The inverter angle order and the current order to the rectifier are the response waveforms of the inverter controller [54, 55].

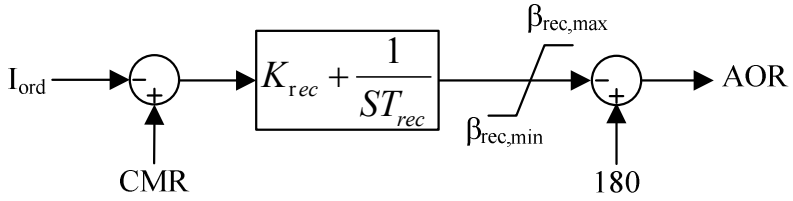


Figure 6-9. HVDC rectifier controller (implemented in real-time hardware).

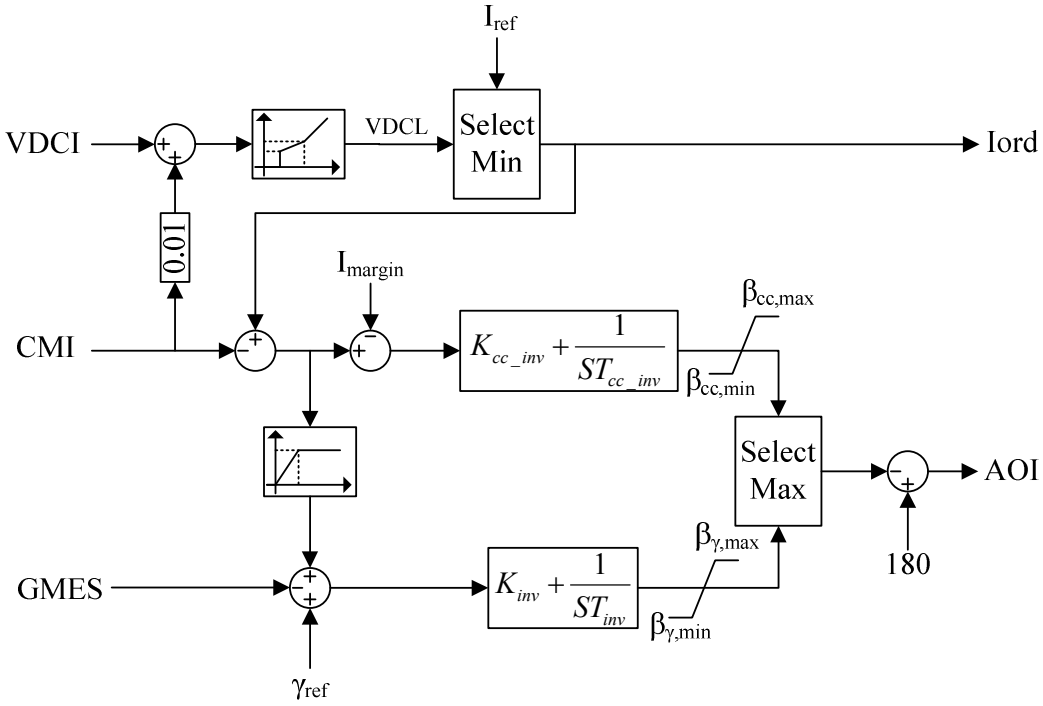


Figure 6-10. HVDC inverter controller (implemented in real-time hardware).

The proportional gain (K_{rec}) and integral time constant (T_{rec}) of the rectifier's current controller, as well as the proportional gain (K_{inv}) and integral time constant (T_{inv}) of the inverter extinction angle (γ) controller form the optimization parameter vector (**p**) in this example. The optimization objective here is to minimize the difference between the measured dc current at the rectifier side (*CMR* in Figure 6-9) and the dc reference current (I_{ref} in Figure 6-10) for step changes in I_{ref}. In this example, I_{ref} is reduced from 1 pu to

0.78 pu and then increased to 1 pu as two step changes at t=0.3 and t=0.5 seconds, respectively. The OF formulated in (6.3) satisfies the mentioned requirement. The total simulation time is 0.7 seconds with the simulation time-step of 50 microseconds. The WR Reinforcement technique of method 1 (WRR-M1) described in Section 5.2.2 is used to accelerate the WR convergence.

$$ISE(\mathbf{p}) = \int_{0.3}^{0.7} \left(1 - \frac{I_{rec}(\mathbf{p},t)}{I_{ref}(t)}\right)^2 dt \tag{6.3}$$

The Simplex in this example is 4-dimensional, and therefore formed by 5 vertices. The Simplex algorithm is initialized with the five initial points shown in Table 6-3. Once a new vector of the optimization parameters ($\mathbf{p}$) is generated by the Simplex algorithm, the first two values (K_{rec} and T_{rec}) are sent to the RTPR connected to the rectifier controller (RTPR1) and the other two (K_{inv} and T_{inv}) are sent to the RTPR connected to the inverter controller (RTPR2).

The Simplex stopping criterion was set as a tolerance of 0.02 in OF value. The Simplex converged in 40 iterations in about 200 minutes. The OF at each Simplex iteration is shown in Figure 6-11.

The resulting OF values corresponding to the initial, optimum, as well as a randomly selected starting point are presented in Table 6-3. From Table 6-3, the optimized OF is 4.68 compared to relatively high pre-optimized values for the initial guess points. The *CMR* waveforms for the two cases, one with the optimized controllers parameters and the other with the randomly chosen controller parameters shown in Table 6-3 are compared in Figure 6-12.

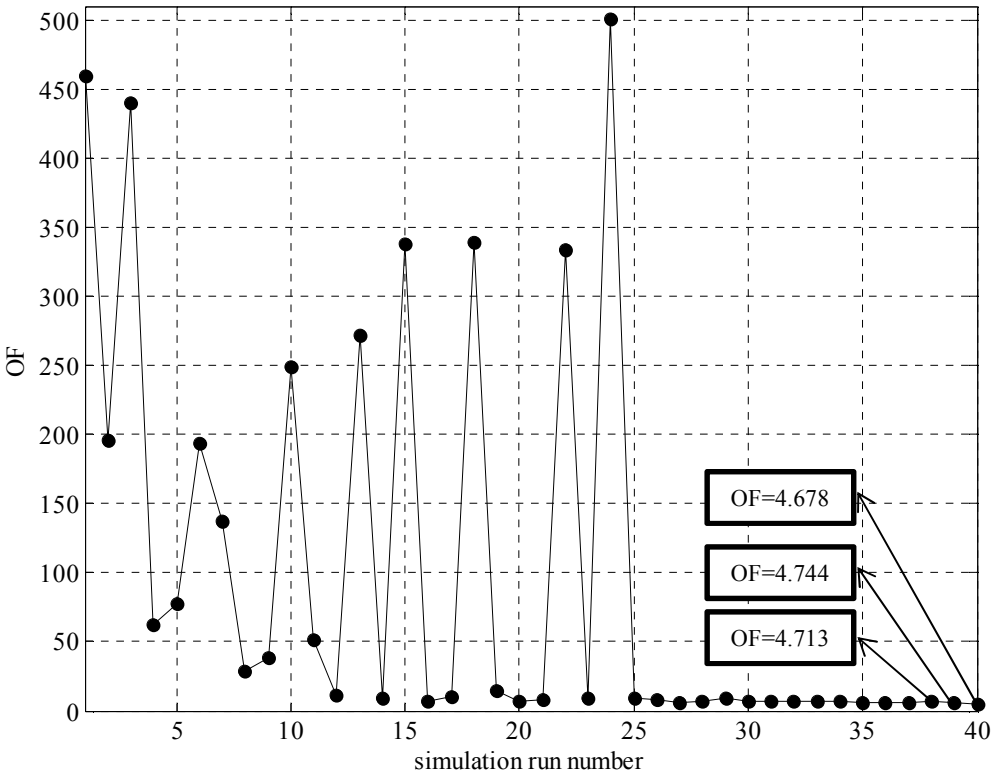


Figure 6-11. Variation of OF at each Simplex iteration.

Table 6-3. OF values for the initial vertices, a random point, and the optimization result.

Specification	K_{rec}	T_{rec}	K_{inv}	T_{inv}	OF
Initial guess	0.5	0.02	2	0.05	338.07
	1	0.08	0.3	0.04	353.40
	0.4	0.05	1	0.08	87.95
	1.2	0.03	0.2	0.08	24.28
	0.3	0.1	0.8	0.03	684.44
Random	1	0.03	1	0.01	49.26
Optimum	**1.4256**	**0.0064**	**0.3308**	**0.1501**	**4.6779**

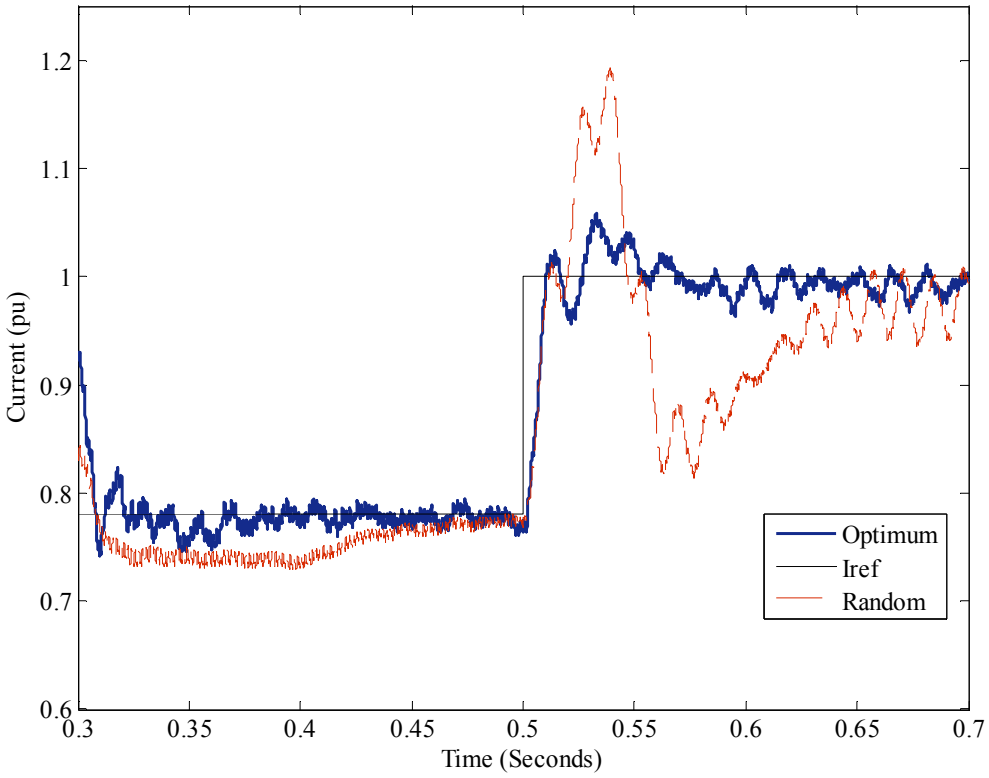


Figure 6-12. Rectifier DC current comparison for the optimized and random controllers parameters.

6.5 Concluding Remarks

It was shown that the proposed controller optimization algorithm based on the WR-HIL simulation technique is capable of optimizing the parameters of single or multiple geographically distributed controller components simultaneously. As the data exchange was over the internet, it would be possible to have multiple controllers optimized each in remote locations. In contrast to optimization with a conventional real-time HIL simulator, this method not only reduces the costs by allowing the power system model simulation on ordinary personal computers, but also saves significant time by allowing the controllers to remain at their distant locations while being optimized. If accurate models of the actual

controllers are available, then the total optimization time can be significantly reduced by applying the WR-ICM acceleration technique.

Chapter 7

Potential WR-HIL applications

In this chapter, several applications are suggested in which the WR-HIL simulation could be used. These include: synchronous testing of multiple wide area HIL, simplified test of multiple hardware with the same hardware structure, and rapid detection of faulty controllers. These are described below.

7.1 Testing of Coupled Wide Area Controllers

Often the controllers of a power system are located in different areas, for example, the sending and receiving end controllers in a dc system. Sometimes it is not convenient to ship these to the WR-HIL testing facility. The approach described in Section 3.3 makes this possible. This is because it uses an internet-based server that communicates with local or remote HUT and WR simulation sites to exchange the waveforms. If the two sites are remote, a GPS-based time stamp is used in the synchronization. This feature has been implemented in the designed WR-HIL platform; however, due to the unavailability (lack of permission from any utility) of a system that spans a large geographic area, actual test-

ing of the complete system was not possible. Another advantage of the approach is that any distributed communication between the remote units, which is part of the control system, can be left in situ and the two can be tested as a unit - something not possible with conventional real-time simulator based HIL.

Another example is the wide area measurement-based control of power systems [65, 66, 67], where the Phasor Measurement Units (PMU), installed in different points of a wide area power system, send the time stamped (by using the Global Positioning System (GPS) technology) voltage and current values to a centralized controller. The centralized controller then sends appropriate signals to multiple wide area controllers. The power system control is more robust with this technique because a controller's response depends on not only the local voltage and current values, but also those of other nodes in the power network. Wide area measurement based Power System Stabilizers [66] are based on this technique.

In order to test and/or tune the wide area controllers, which communicate signals via fiber optics or LAN simultaneously, all actual controllers should be in a closed loop with the simulated model of the power system. As of today, there is no method of HIL simulation to conduct such testing unless the controllers are disconnected from the system and all located in the same room with the real-time simulator. The WR-HIL simulation technique can be used to conduct such HIL simulations while the controllers can remain at their original places. In these cases, two possibilities can be considered. First, the purpose of doing HIL simulation is only testing the controllers. Second, the communication links between the controllers are also part of the hardware under test.

In the first case, any physical connection between the controllers is disconnected and the communication signals are added to the WR signals. For example, the demanding current from the HVDC's inverter controller to the rectifier controller is added as an output waveform of the inverter controller captured by the RTPR and as an input to the rectifier which is played back in real-time by the RTPR at every iteration of the WR. In this scheme, there is no direct link between the controllers and therefore the communication system is not part of the test. This is shown in Figure 7-1 when the switches "S" connect the ADC and DAC of the RTPRs to the simulated link "L". In this scheme, no GPS signal is required.

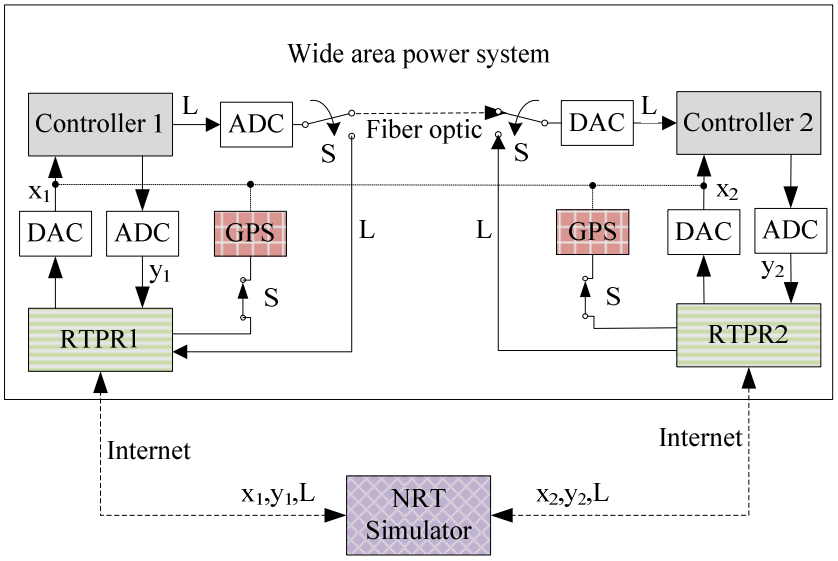


Figure 7-1. Closed loop test of wide area controllers excluding or including the communication link.

In the second case, the physical communication link between the controllers is left intact in the testing process, rather than simulating it. In this case, both switches "S" shown in Figure 7-1 connect the ADC and DAC of the RTPRs to the physical link (e.g. a fiber optic cable) between the controllers.

In regular WR-HIL simulation, i.e. simulated communication links and without use of a GPS, each controller is considered as an independent subsystem. However, when the physical links between the controllers are preserved in the simulation, all the RTPR units must be equipped with GPS signals so that all the waveforms for all the controllers are played back in real-time simultaneously. In this case, all the controllers with the interconnecting links are considered as a single subsystem in the WR-HIL simulation algorithm. The application of the accelerated convergence techniques proposed in Chapter 5 can greatly improve the convergence of the WR algorithm with the presence of random communication delays in the interconnecting physical links between the controllers.

7.2 Simplified Test of Multiple Controllers with Identical Structure

There are a number of examples where multiple hardware, such as controllers and protection relays, with identical hardware structure (with the same or different parameters) are installed in the power system. An example is a case with multiple PI controllers in a system with different proportional/integral constants, where each PI controller has the same hardware platform as the others, e.g., all provided by the same manufacturer. Assume that the response of each controller is related to the response of other controllers and therefore all the controllers need to be tested or tuned simultaneously. In the traditional approaches using a real-time simulator system, all the controllers are wired to the simulator. With WR-HIL, the test setup can be significantly simplified as shown in Figure 7-2.

It is assumed that the controller is a modern design and its parameters are electronically adjustable, either with an analog voltage or with a digital word. A solution is proposed here to simplify such type of simulation and to reduce the hardware setup and wiring complexity in Figure 7-2.

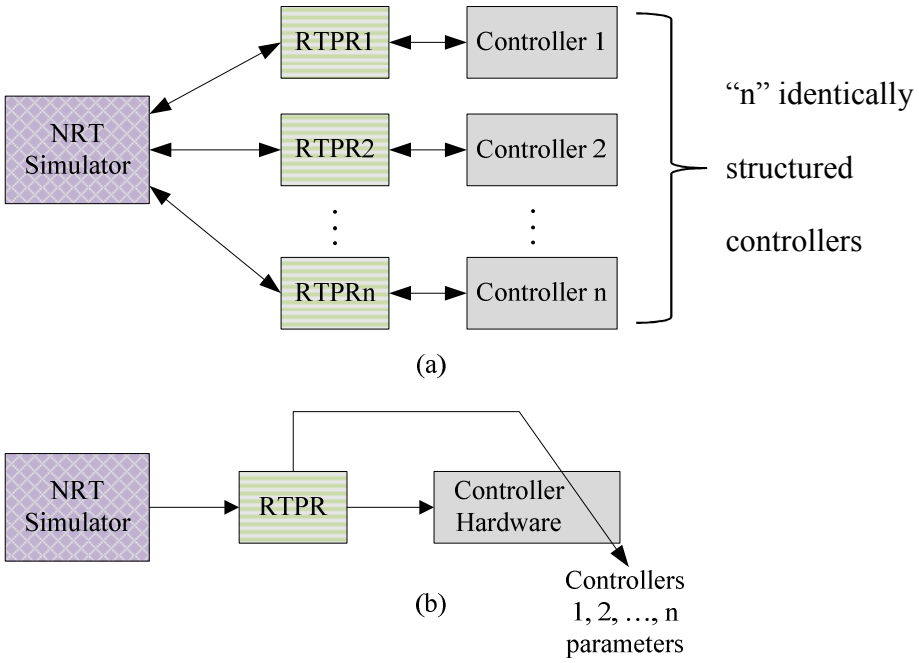


Figure 7-2. Simplified test of multiple hardware with the same structure.

Figure 7-2 (b) displays the proposed setup where only one controller hardware is connected to one RTPR device. In this figure, it is assumed that the hardware parameters can be set by the RTPR. The simulation starts with setting the parameters for "Controller 1". Once the Non-Real-Time (NRT) simulation results are available from the first iteration (say, iteration 1) of the WR algorithm, they are sent to the controller under test, which is loaded with the parameters of "Controller 1". Its response is acquired and stored. No new simulation cycle is initiated at this stage. Then, the controller hardware is loaded with the parameters of "Controller 2" and the response is captured once again, with re-

sponses from iteration 1 of the off-line simulation. This is done for all "n" parameter sets of all "n" controllers. Once the responses of all n controllers are obtained, they are made available to the simulator which conducts its next simulation iteration (iteration 2). This process is continued till convergence. Therefore, the "Controller Hardware" presented in Figure 7-2 (b) is one piece of hardware but represents "n" distinct hardware subsystems.

Also as described in Section 1.2 the playback to the non-real-time simulator could include the latest simulation results from the current iteration (Gauss-Seidel) or waveforms recorded from the previous iteration (Gauss-Jacobi).

This simplification is only possible with the WR-HIL technique because it is an iterative method and therefore the other subsystems are "on-hold" when one subsystem is being operated. This is an advantage over conventional real-time simulators which demand that the response of all subsystems must be available at every simulation time-step when using the real-time simulators.

Additionally, the method can be extended to the situation when the controllers do not all have the same structure but can be partitioned into a small number of sets, each with a different structure. For instance, these could be four PI controllers with the same hardware platform and five protection relays with the same hardware structure. In this example, one controller hardware can represent four PI controllers and one relay hardware represents five. Therefore, the problem of four controllers, five relays, nine RTPRs is reduced to one controller, one relay, and two RTPRs which significantly reduces the size, complexity, and the cost of the HIL simulation.

7.3 Rapid Detection of Faulty Hardware

The RTPR device can be used as a tool to detect a faulty controller. The proposed method is shown in Figure 7-3. Assume that there are "n" controllers being operated in a section of the power system (either local or wide area). In addition to the faults occurring in transmission and distribution lines, a failure in the performance of a controller can also cause malfunction of the system. In this situation, it is important to quickly detect which controller among many installed controllers has failed. Assuming that the section of the power system is isolated due to the fault, or backup controllers are switched on, testing each disconnected controller sequentially is time consuming and testing the controllers all at the same time by using the real-time digital simulators is expensive.

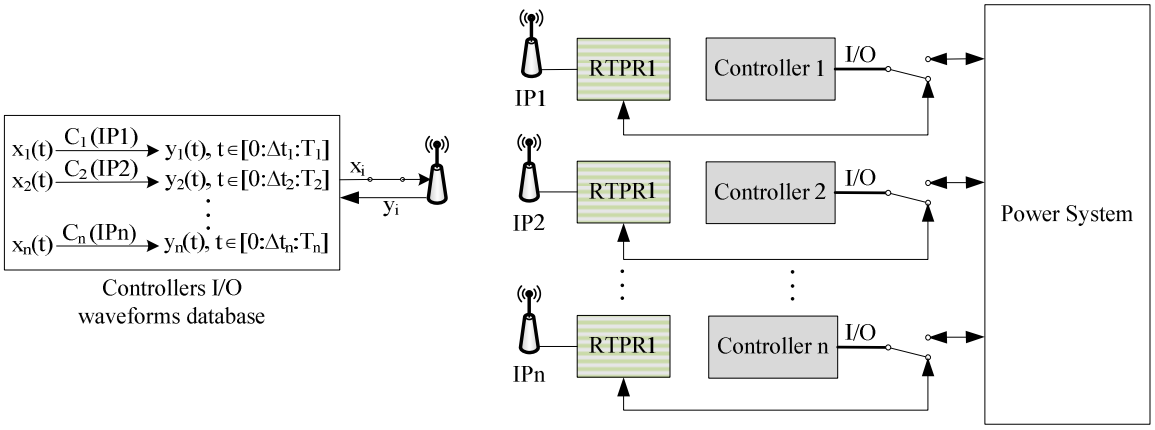


Figure 7-3. Using RTPRs for rapid detection of faulty controller(s).

Suppose that all the input and output signals of the "n" controllers in Figure 7-3 resulting from the HIL simulation of the controllers within the model of the power system have been recorded for a time window of T seconds at some earlier time when the

equipment was healthy. At a later time, while the controllers are connected to the actual power system, one or more of these controllers may become faulty.

The switches positioned as shown in Figure 7-3 represent the situation when the controllers are disconnected from the power system after a fault is detected and connected to "n" RTPRs. Each RTPR is equipped with a WiFi module with a predefined IP address, which is also stored in the controllers I/O waveforms database shown in Figure 7-3. Each controller's input signal, x_i, is sent to the RTPR, which is connected to the corresponding controller. The RTPRs playback the waveforms x_i in real-time and sample and record the response waveforms of the controllers, y_i. The controllers' response waveforms are then sent back to the database to be compared with the expected waveforms (generated from earlier saved HIL simulation results when all the controllers were healthy). The controllers whose post-fault response waveforms are different from the pre-fault waveforms can thus be detected. The WiFi module eliminates the need for moving the equipment off0site for doing this testing approach.

7.4 Concluding Remarks

Some potential practical applications for the proposed method were suggested. The method is suitable for development of a server based hardware-in-the-loop simulation. In this scheme, a server implements the waveform relaxation method with the proposed algorithms and a network infrastructure is used to automatize the exchange of the waveforms between the simulation and the hardware subsystems. The proposed method can be effectively used to test the coupled wide area measurement based controllers in power

systems using GPS timing if necessary. Further, simplified test of multiple hardware with the same hardware platforms and rapid detection of faulty controllers in a power system are the other suggested applications enabled by this approach.

Chapter 8

Contributions, Conclusions, and Future Work

This thesis has contributed to advances in hardware-in-the-loop simulation methods. The new algorithms and tools developed in this thesis expand the traditional waveform relaxation method to a new cost-effective hardware-in-the-loop simulation technique. The proposed method allows the simulation of power system models on an off-line electromagnetic transient simulation program using an ordinary computer, which makes the hardware-in-the-loop simulation highly cost-effective. In addition, the simulation system(s) and the hardware under test units can be situated at locations remote from each other. The latter is achieved from the inherent properties of the waveform relaxation method, which decouples the subsystems in the closed loop simulation. Also the proposed method adds the ability of conducting the hardware-in-the-loop simulation using any desired electromagnetic transient simulation program, or even a combination of programs for different sections of the system, because only the off-line simulation results are

used in the proposed method. The proposed algorithms and methods are verified by comparing the obtained experimental results with those obtained from the hardware-in-the-loop simulation results using a commercial real-time digital simulator. A variety of experimental examples including testing of the controllers, protective relays, and the power hardware equipment are given to prove the generality of the approach.

This chapter summarizes the main contributions of this thesis and further recommends some other related work that can be carried out in future.

8.1 Main Contributions of the Thesis

8.1.1 Hardware implementation of the method

- To allow the use of the waveform relaxation method to conduct hardware-in-the-loop simulations, a hardware interface is required. A special hardware platform was designed in this thesis to play back the waveforms resulting from the off-line simulation of the power system models and sample and record the response waveforms of the hardware under test. The designed hardware can communicate with the simulator via internet to enable testing of internet distributed hardware, i.e., geographically located far from the simulator. It was shown that the number of input/output channels used in an experiment as well as the size of the memory of the designed real-time player/recorder determines the maximum simulation time.

- The proposed waveform relaxation based hardware-in-the-loop simulation method is efficient and cost-effective in the hardware side. The designed real-time player/recorder device is simple, low cost, scalable, and portable (internet based RTPR).

- The designed real-time player/recorder was successfully tested to perform the hardware-in-the-loop simulation for protective relays, controllers, and power hardware equipment. In the latter, a power amplifier was used to interface the real-time player/recorder device with the power hardware under test.

8.1.2 Identifying WR-HIL challenges and proposing solutions

- It was shown that the implementation of the waveform relaxation method in conducting the hardware-in-the-loop simulation is not straight forward. The main limitations and practical challenges for the WR-HIL simulation technique were identified and several solutions were proposed followed by an experimental example for each solution.

- A theoretical analysis was conducted to show that the WR-HIL simulation is prone to instability if certain conditions are not satisfied. A stabilizing technique was proposed to improve the stability of the technique when applied to the circuit simulation. It was shown that addition of a damping resistor in parallel with the simulated power system can greatly increase the stable region of the simulation without affecting the accuracy of the converged results. The current drawn by the parallel resistor is compensated through the waveform relaxation iterations and therefore does not affect the converged waveforms.

- In many cases, even with the theoretical stability criteria met, the hardware-in-the-loop simulation still showed nonconvergence when the classical waveform relaxation was used. This was deemed to be a result of factors such as noise and errors in the analog-

digital conversion and the amplifier. Several solutions were proposed to improve the convergence of the WR-HIL simulation technique.

- The waveform relaxation piecewise fixed convergence (WR-PWFC) method was proposed to identify and limit the convergence window size. Using this method, the converged parts of the waveforms are identified and protected from any further changes possibly caused by the hardware noise and other related errors. The method was shown to be highly efficient by comparing the obtained results for an experimental example with those obtained from the real-time digital simulator.

- Inclusion of an internal controller model (WR-ICM) was proposed to accelerate the waveform relaxation based hardware-in-the-loop simulation. Two approaches had to be designed. WR-ICM1 is usable when the hardware under test accepts only analog waveforms in the input channels. WR-ICM2 is used when WR-ICM1 cannot be applied, e.g. one of the inputs to the hardware under test is a digital signal, and the hardware generates all analog waveforms in the output channels. In both methods, an approximate model of the hardware under test is added in the simulation part. The response of the model of the controller is compensated through the waveform relaxation iterations to ensure the high accuracy of the converged results. The WR-ICM method showed to be highly efficient in accelerating the waveform relaxation simulation convergence.

- The waveform relaxation averaging method (WR-AM) was proposed as a solution to decrease the effect of noise and other hardware errors. A special weighted averaging method was used to identify the converged parts of the waveforms and use the average of the converged parts over the last few iterations. For the under-convergence parts, less weight is given to the averaged waveform and more weight is given to the waveforms re-

sulting from the current iteration to allow the faster convergence of the waveforms. The proposed method was shown to be effective in reducing the effect of the noise on the converged parts of the waveforms and overall improvement of the waveform relaxation convergence with the presence of the noise.

- A parametric analysis was conducted to evaluate and compare each of the proposed accelerated convergence techniques applied individually and in combination. The simulation results showed that the best performance is achieved when all the proposed methods are applied simultaneously.

- It was shown that using the waveform relaxation method for testing the closed loop performance of the protective relays also encounters some practical challenges. The main issue was identified as the requirement for the manual reset of the commercially available relays after detection of a fault. As a result, the user may have to press a button manually before a new waveform relaxation iteration is performed. This is not favorable because firstly, the relay may be in remote locations, and secondly, it needs the user direct interaction with the simulation. Fortunately, several modern relays provide a logic bit in the relay settings to allow the automatic reset of the relay after a certain period of time defined by the user. This function is suggested in this thesis to be effectively used when testing a relay using the waveform relaxation technique.

8.1.3 Optimization of controller parameters using WR-HIL

- The proposed method was equipped with an optimization algorithm. It was shown that the approach could be used to optimize the parameters of a controller so that the best closed loop performance of the controller in the loop with a simulated power system is

achieved. The proposed algorithm was tested to optimize the parameters of actual HVDC and STATCOM controllers using the Simplex optimization method.

8.1.4 Suggested potential applications for WR-HIL

- Some practical applications for the proposed method were suggested. The method is suitable for development of a server based hardware-in-the-loop simulation. In this scheme, a server implements the waveform relaxation method with the proposed algorithms and a network infrastructure is used to automatize the exchange of the waveforms between the simulation and the hardware subsystems. The proposed method can be effectively used to test the coupled wide area measurement based controllers in power systems. Further, simplified test of multiple hardware with the same hardware platforms and rapid detection of faulty controllers in a power system are the other suggested applications for the proposed approach.

8.2 Recommended Future Work

The methods and algorithms proposed in this thesis verified by the experimental results show to be very promising in the development of a new hardware-in-the-loop simulation. Further research and investigation can improve the performance of the proposed approach. Some suggestions for the further research are proposed in following.

8.2.1 Implementation of suggested applications for WR-HIL simulation in Chapter 7

- Testing of coupled wide area controllers (refer to Section 7.1);

- Simplified test of multiple controllers hardware with identical structures (refer to Section 7.2);

- Rapid detection of faulty hardware (refer to Section 7.3).

8.2.2 Accelerating WR-HIL convergence for testing black box controllers

One of the efficient acceleration techniques for the WR-HIL simulation proposed in Chapter 5 of this thesis is the WR-ICM method. In this method, an approximate model of the hardware under test is included in the power system model simulation to accelerate the WR convergence. If the actual controller is black box, i.e., no approximate model can be provided, then appropriate model approximation techniques may be employed. Such techniques only rely on the input and output waveforms of the black box controller to give a model which approximately represents the transient behavior of the controller within the power system model under study. Karhunen-Loeve decomposition [64] is an example of such methods which relies on the experimental response waveforms of the controller to generate an approximate model.

8.2.3 Parallel simulation of the power system models in WR-HIL approach

As mentioned in the Section 1.2.2 of the thesis, the Gauss-Jacobi waveform relaxation (WR-GJ) method is traditionally used for the parallel simulation of the power system models. The WR-HIL simulation technique proposed in this thesis could be extended to include this type of simulation in order to decrease the overall simulation time. In this scheme, a proper partitioning method shall be used to partition the original power system model and distribute the generated subsystems on multiple processing units and perform the WR-GJ algorithm. This method will be efficient in reducing the simulation time especially when the size of the power system model is large and/or when different parts of the power system are modelled in different simulation programs. Currently, inclusion of multiple subsystems modelled in the same or different simulation programs is possible with the methods presented in the thesis, but they are solved by the Gauss-Seidel waveform relaxation (WR-GS) method in the designed platform which can be replaced by the WR-GJ method if multiple processors are available.

References

[1] Xi Lin, "System equivalent for real time digital simulator," PhD dissertation, Dept. of Electrical and Computer Engineering, University of Manitoba, Canada, 2010.

[2] J. Grainger, W. Stevenson, *Power System Analysis*, McGraw-Hill, New York, 1994.

[3] D. A. Woodford, A. M. Gole, R. W. Menzies, "Digital simulation of DC links and AC machines," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No. 6, June 1983.

[4] C. Dufour, J. Belanger, S. Abourida, "Real-Time and Hardware-in-the-Loop Simulation of Electric Drives and Power Electronics: Process, Problems and Solutions," in Proc. 2005 International Power Electronics Conf., pp. 1908-1913.

[5] P.M. Menghal, A. J. Laxmi, "Real Time Simulation: Recent Progress & Challenges," in Proc. 2012 IEEE EPSCICON Conf., pp. 1-6.

[6] C. Dufour, S. Abourida, J. Belanger, V. Lapointe, "Real-Time Simulation of Permanent Magnet Motor Drive on FPGA Chip for High-Bandwidth Controller Tests and Validation," in Proc. 2006 IEEE IECON Conf., pp. 2591-2596.

[7] F. Adler, H. Stagge, R. W. De Doncker, "A Versatile Modular Hardware Platform for Distributed Large-Scale Real-Time Simulation," in Proc. 2012 International Power Electronics and Motion Control Conf., pp. LS3b.1-1 – LS3b.1-6.

[8] W. Ren, M. Steurer, T. L. Baldwin, "Improve the Stability and the Accuracy of Power Hardware-in-the-Loop Simulation by Selecting Appropriate Interface Algorithms," *IEEE Trans. Ind. Appl.*, vol. 44, pp. 1286-1294, July 2008.

[9] B. Luitel, G. K. Venayagamoorthy, G. Oliveira, "Developing Neural Networks Library in RSCAD for Real-Time Power System Simulation," in Proc. 2013 IEEE Computational Intelligence Applications in Smart Grid, pp. 130-137.

[10] W. Ren, M. Sloderbeck, M. Steurer, V. Dinavahi, T. Noda, S. Filizadeh, A. R. Chevrels, M. Matar, R. Iravani, C. Dufour, J. Belanger, M. O. Faruque, K. Strunz, J. A. Martinez, "Interfacing Issues in Real-Time Digital Simulators," *IEEE Trans. Power Del.*, vol. 26, pp. 1221-1230, April 2011.

[11] C. Choi, W. Lee, "Analysis and Compensation of Time Delay Effects in Hardware-in-the-Loop Simulation for Automotive PMSM Drive System," *IEEE Trans. Ind. Electron*, vol. 59, pp. 3403-3410, September 2012.

[12] I. D. Yoo, A. M. Gole, "Compensating for Interface Equipment Limitations to Improve Simulation Accuracy of Real-Time Power Hardware In Loop Simulation," *IEEE Trans. Power Del.*, vol. 27, pp. 1284-1291, July 2012.

[13] M. Dargahi, A. Ghosh, G. Ledwich, F. Zare, "Studies in power hardware in the loop (PHIL) simulation using real-time digital simulator (RTDS)," IEEE international conference on power electronics, drives and energy systems, pp. 1-6, Bengaluru, India, December 2012.

[14] A. Viehweider, G. Lauss, F. Lehfuss, "Interface and stability issues for SISO and MIMO power hardware in the loop simulation of distributed networks with photovoltaic

generation," *International journal of renewable energy research*, Vol. 2, No. 4, pp. 631-639, 2012.

[15] C. Schacherer, J. Langston, M. Steurer, M. Noe, "Power Hardware-in-the-Loop Testing of a YBCO Coated Conductor Fault Current Limiting Module," *IEEE Trans. On Applied Superconductivity*, vol. 19, no. 3, part 2, pp.1801–1805, 2009.

[16] G. Lauss, F. Lehfuss, A. Viehweider, T. Strasser, "Power hardware in the loop simulation with feedback current filtering for electric systems," IECON 2011 - 37th Annual Conf. IEEE Ind. Electron. Society, pp.3725–3730, Melbourne, Australia, Nov. 2011.

[17] A. D. Jalnapurkar, "Circuit simulation using distributed waveform relaxation techniques," PhD dissertation, Electrical Engineering Dept., University of Saskatchewan, Saskatoon, Saskatchewan, Canada, 1998.

[18] Y. L. Jiang, O. Wing, "A note on convergence conditions of waveform relaxation algorithms for nonlinear differential-algebraic equations," *Applied Numerical Mathematics*, vol. 36, no. 2-3, pp. 281-297, 2001.

[19] J. Janssen, "Acceleration of waveform relaxation methods for linear ordinary and partial differential equations," PhD dissertation, Computer Science Dept., Katholieke University of Leuven, Heverlee, Belgium, 1997.

[20] M. L. Crow, M. D. Ilic, "The waveform relaxation method for systems of differential/algebraic equations," *mathl. comput. Modeling,* vol. 19, no. 12, pp. 67-84, 1994.

[21] Y. L. Jiang, O. Wing, "Convergence conditions of waveform relaxation methods for circuit simulation," Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, vol. 6, pp. 232-235, Monterey, CA, 1998.

[22] M. L. Crow, M. D. Ilic, "The waveform relaxation method for systems of differential/algebraic equations," Proceedings of the 29th conference on decision and control, vol. 2, pp. 453-458, Honolulu, HI, 1990.

[23] M. L. Crow, M. D. Ilic, J. K. White, "Convergence properties of the waveform relaxation method as applied to electric power systems," IEEE International Symposium on Circuits and Systems, vol. 3, pp. 1863-1866, Portland, OR, 1989.

[24] J. Y. Pan, Z. Z. Bai, "On the convergence of waveform relaxation methods for linear initial value problems," *Journal of Computational Mathematics*, vol. 22, no. 5, pp. 681-698, 2004.

[25] J. White, A. Sangiovanni-Vincentelli, F. Odeh and A. Ruehli, "Waveform relaxation: Theory and practice," *Trans. Soc. Comput. Simul*. 2 (1985), pp. 95–133.

[26] Z. Jackiewicz, M. Kwapisz, "Convergence of Waveform Relaxation Methods for Differential-Algebraic Systems," *SIAM Journal*, Vol. 33, No. 6, pp. 2303-2317, August 2006.

[27] E. Lelarasmee, A.E. Ruehli, D. Karlson, A.L. Sangiovanni-Vincentelli, "The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits," *IEEE Transactions on computer-aided design of integrated circuits and systems*, Vol. CAD-I, No. 3, pp. 131-145, July 1982.

[28] J. Sun and H. Grotstollen, "Fast Time-Domain Simulation by Waveform Relaxation Methods," *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 44, (8), 660-666, (August 1997).

[29] S. Skelboe, "Adaptive partitioning techniques for ordinary differential equations," *BIT Numerical Mathematics*, Vol. 46, No. 3, pp. 617-629, August 2006.

[30] N. M. Nakhla, A.E. Ruehli, M. S. Nakhla, R. Achar, "Simulation of Coupled Interconnects Using Waveform Relaxation and Transverse Partitioning," *IEEE Transactions on Advanced Packaging*, Vol. 29, No. 1, pp. 78-87, February 2006.

[31] M. Ossowski, A. Kuczynski, "Speeding up waveform relaxation algorithms," International Workshop Computational Problems of Electrical Engineering, pp. 62-65, Zakopane, Poland, 2004.

[32] S. Roy, A. Dounavis, "Waveform relaxation with overlapping based partitioning for fast transient simulation of package/board power distribution networks," IEEE conference on Electric Performance of Electronic Packaging and Systems, pp. 252-255, Tempe, AZ, 2004.

[33] J. Marti, L. Linares, J. Calvino, H. Dommel, J. Lin, "OVNI: an object approach to real-time power system simulators," International Conference on POWERCON, vol. 2, pp. 977-981, August 1998.

[34] M. L. Crow and M. Ilic, "The parallel implementation of the waveform relaxation method for transient stability simulations," *IEEE Trans. Power Syst.,* vol. 5, pp.922 - 932, 1990.

[35] M. Ilic-Spong, M. L. Crow and M. A. Pai, "Transient stability simulation by wave-form relaxation methods," *IEEE Trans. Power Syst*., vol. 2, pp.943 - 952, 1987.

[36] B. Leimkuhler, "Time-step acceleration of waveform relaxation," *SIAM JOURNAL ON NUMERICAL ANALYSIS*, vol. 35, 1998.

[37] O. Nevanlinna, "Remarks on Picard Lindelof iteration," *BIT 29*, pp. 328-346, 1989.

[38] M. W. Reichelt, J. K. White, J. Allen, "Optimal convolution sor acceleration of waveform relaxation with application to parallel simulation of semiconductor devices," *SIAM J. Sci. Comput*., Vol. 16, No. 5, pp. 1137-1158, September 1995.

[39] M. Hu, K. Jackson, J. Janssen, S. Vandewalle, "Remarks on the optimal convolution kernel for CSOR waveform relaxation," *Advances in computational mathematics*, Vol. 7, No. 1-2, pp. 135-156, 1997.

[40] F. Cingoz, H. Behjati, A. Davoudi, "Accelerated simulation of ultracapacitors using waveform relaxation method," IEEE Transportation Electrification Conference and Expo, pp. 1-6, Dearborn, MI, 2012.

[41] F. Pruvost, P. Laurent-Gengoux, F. Magoules, B. Haut, "Accelerated waveform relaxation methods for power systems," IEEE international conference on electrical and control engineering, pp. 2877-2880, Yichang, China, September 2011.

[42] R. Achar, M. S. Nakhla, H. S. Dhindsa, A. R. Sridhar, "Parallel and Scalable Transient Simulator for Power Grids via Waveform Relaxation (PTS-PWR)," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*, vol. 19, pp. 319-332, February 2011.

[43] A. M. Kulkarni, K. Salunkhe, M. C. Chandorkar, S. P. Panda and N. Sankaranarayanan, "Waveform Relaxation based Iterative Real-time Playback Schemes for Testing of Wide-area Power System Controllers," conf. rec., IEEE Power and Energy Society General Meeting, 25-29 July 2010, Minneapolis, USA.

[44] "DIGILENT®," Internet: www.digilentinc.com, [Oct. 09, 2015].

[45] M. Goulkhah, A. M. Gole, J. Fuith, A. M. Kulkarni, "Comparison of Waveform Relaxation with Traditional Real Time Digital Simulation for Closed Loop Performance Testing of Protective Relays," in Proc. 2014 Cigre Canada Conference.

[46] J. L. Blackburn and T. J. Domin, "Protective Relaying - Principles and Applications," CRC Press, third Edition, 2006.

[47] M. Hong, S. Horie, Y. Miura, T. Ise, C. Dufour, "A Method to Stabilize a Power Hardware-in-the-loop Simulation of Inductor Coupled Systems," in Proc. 2009 International Conference on Power Systems Transients.

[48] A. M. Gole, R. W. Menzies, H. M. Turanli, D. A. Woodford, "Improved interfacing of electrical machine models to electromagnetic transients programs," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-103, No. 9, pp. 2446-2451, Sep. 1984.

[49] A. K. Al-Khalifah, E. F. El-Saadany, "Investigation of magnetizing inrush current in a single-phase transformer," in Proc. 2006 Large Engineering Systems Conference on Power Engineering, pp. 165-171.

[50] R. A. Salas, "Nonlinear Inductance Calculations of a Ferrite Inductor with a 2D Finite Element Model," in Proc. 2011 International Conference on Electromagnetics in Advanced Applications, pp. 986-989.

[51] S. G. Abdulsalam, W. Xu, L. A. Neves, X. Liu, "Estimation of transformer saturation characteristics from inrush current waveforms," *IEEE Transactions on Power Delivery*, Vol. 21, No. 1, pp. 170-177, January 2006.

[52] S. G. Abdulsalam, W. Xu, "A sequential phase energization method for transformer inrush current reduction-transient performance and practical considerations," *IEEE Transactions on Power Delivery*, Vol. 22, No. 1, pp. 208-216, January 2007.

[53] Garcia CE, Prett DM, Morari M, "Model Predictive Control: Theory and Practice – a Survey," *Automatica*, vol. 25, pp. 335-348, May 1989.

[54] M. Szechtman, T. Wess, C. V. Thio, "First Benchmark Model for HVDC Control Studies," *Electra*, no. 135, April 1991.

[55] C. Karawita, U. Annakkage, "Multi-Infeed HVDC Interaction Studies Using Small-Signal Stability Assessment," *IEEE Trans. Power Del*, vol. 24, pp. 910-918, April 2009.

[56] M. Goulkhah, A. M. Gole, "Practical Application of Waveform Relaxation Method for Testing Remote Protective Relays," in Proc. 2015 IEEE Int. Conf. on Industrial Technology (ICIT15), Seville, Spain, 2015.

[57] In-Kwon Park, A. M. Gole, Paul Forsyth, Rick Kuffel, "Hardware in the Loop Optimization Using a Real Time Simulation Environment," in Proc. International Conference on Power Systems Transients (IPST), Delft, Netherlands, 2011.

[58] A. S. Fraser, "Simulation of genetic systems," *Journal of Theoretical Biology*, vol. 2, no. 3, pp. 329-346, 1962.

[59] J. A. Nelder, and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308-313, 1965.

[60] A. M. Gole, S. Filizadeh, "Inclusion of Robustness into Design Using Optimization-Enabled Transient Simulation," *IEEE Transactions on Power Delivery*. vol. 20, no. 3, pp. 1991-1997, July 2005.

[61] A. Hedar, M. Fukushima, "Minimizing multimodal functions by simplex coding genetic algorithm," *Optimization Methods and Software*, vol. 18, no. 3, pp. 265–282, 2003.

[62] T. Weise, "Global Optimization Algorithms – Theory and Application," Germany: it-weise.de (self-published), 2009. [Online]. Available: http://www.it-weise.de/.

[63] Kalyan K. Sen, Mey Ling Sen, "Introduction to FACTS Controllers: Theory, Modeling, and Applications," 1st edition, Wiley-IEEE Press, Sep. 2009.

[64] M. Rathinam, L. R. Petzold, "Dynamic iteration using reduced order models: a method for simulation of large scale modular systems," *SIAM JOURNAL ON NUMERICAL ANALYSIS*, vol. 40, no. 4, pp. 1446-1474, 2002.

[65] Hui Ni, G.T. Heydt, L. Mili, "Power System Stability Agents Using Robust Wide Area Control," *IEEE Transactions on Power Systems*, Vol. 17, No. 4, pp. 1123-1131, November 2002.

[66] F. Okou, L.A. Dessaint, O. Akhrif, "Power Systems Stability Enhancement Using a Wide-Area Signals Based Hierarchical Controller," *IEEE Transactions on Power Systems*, Vol. 20, No. 3, pp. 1465-1477, August 2005.

[67] J. Bertsch, C. Camal, D. Karlson, J. McDaniel, "Wide-Area Protection and Power System Utilization," Proceedings of the IEEE, Vol. 93, No. 5, pp. 997-1003, May 2005.

Appendix 1

Gauss-Seidel WR Operator Matrix

This appendix extracts the Gauss-Seidel WR operator matrix for a set of algebraic equations.

Consider the algebraic equation in (A.1).

$$\mathbf{Ax} = \mathbf{b} \tag{A.1}$$

For simplicity, a set of two algebraic equations in considered as in (A.2).

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{A.2}$$

The matrix **A** can be written as the summation of its strictly lower triangular matrix (**L**), diagonal matrix (**D**), and strictly upper triangular matrix (**U**) as in (A.3).

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U} = \begin{bmatrix} 0 & 0 \\ a_{21} & 0 \end{bmatrix} + \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} \\ 0 & 0 \end{bmatrix} \tag{A.3}$$

(A.1) can be rewritten in the form of (A.4) as below.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases} \tag{A.4}$$

Application of the Gauss-Seidel WR method to solve the set of algebraic equations of (A.4) results in the iteration equations of (A.5).

$$\begin{cases} a_{11}x_1^{k+1} + a_{12}x_2^{k} = b_1 \\ a_{21}x_1^{k+1} + a_{22}x_2^{k+1} = b_2 \end{cases} \tag{A.5}$$

(A.5) can be written in the matrix form of (A.6).

$$\begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^{k} \\ x_2^{k} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{A.6}$$

The general form of (A.6) is (A.7) with **L**, **U**, and **D** matrices defined in (A.3).

$$(\mathbf{L}+\mathbf{D})\mathbf{x}^{k+1} + \mathbf{U}\mathbf{x}^{k} = \mathbf{b} \tag{A.7}$$

From (A.7), we have:

$$\mathbf{x}^{k+1} = -(\mathbf{L}+\mathbf{D})^{-1}\mathbf{U}\mathbf{x}^{k} + (\mathbf{L}+\mathbf{D})^{-1}\mathbf{b} \tag{A.8}$$

Therefore, the Gauss-Seidel WR relates the set of variables **x** from a current iteration of k+1 and a previous iteration of k with the operator matrix of (A.9).

$$\mathbf{M} = -(\mathbf{L}+\mathbf{D})^{-1}\mathbf{U} \tag{A.9}$$