CHAPTER 5

REAL TIME IMPLEMENTATION USING DSP 28335

A real time controller of a Single Machine Infinite Bus & multi-machine power system has been developed in the Mat lab/Simulink environment with code composer studio v3.3 for the Improvement of the stability. Stability has been given much attention, and is being regarded as one of the major sources of power system insecurity. The modelling and simulation of the developed controller has been demonstrated through the real time Emulator available with ezdsp28335 to investigate the stabilization control performance for a study SMIB & multi-machine power system. The robust design of multi-machine power system stabilizers (PSSs) using multi-objective particle swarm optimization (MOPSO) is also discussed. The problem of selecting the stabilizer parameters is converted to an optimization problem with integral square error (ISE) and integral of time multiplied absolute value of the error (ITAE)-based objective functions. The MOPSO is employed to search for optimal PSS parameters for a wide range of operating conditions. The effectiveness of the proposed approach in enhancing the dynamic stability of power systems is confirmed through eigenvalue analysis and nonlinear simulation results.

5.1 Introduction

The TMS320F28335 is highly integrated, high-performance solutions for demanding controlapplications. The TMS320F28335 is a standalone development platform that enables user to evaluate and develop applications. It has a wide range of application environments. The TMS320F28335 is designed to work with Code Composer Studio. Code Composer communicates with the board through an On Board JTAG emulator [1].

5.2 Code generation using CCS v3.3 for TMS320F28335

When the Simulink schematics are done, converted to fixed point precision and tested, the last step is to generate C code downloadable on the microcontroller. This step is done using two toolboxes of Mat lab:

- The Real Time Workshop
- The Real Time Workshop Embedded Coder

The Real Time Workshop is an essential tool used in rapid prototyping with Simulink. Automatic program building allows you to make design changes directly to the block diagram, putting algorithm development (including coding, compiling, linking, and downloading to target hardware) under control of a single process [1].

5.2.1 Steps to generate optimized C code

- Design a model in Simulink

The rapid prototyping process begins with the development of a model in Simulink. Using principles of control engineering, it’s possible to model plant dynamics and other dynamic components that constitute a controller and/or an observer [2].

- Simulate the Model in Simulink

Using MATLAB-Simulink, and toolboxes it’s possible to develop algorithms and analyse theresults.If the results are not satisfactory, it’s possible to iterate the modelling and analysis process until results are acceptable [2].

- Generate Source Code with Real-Time Workshop

Once simulation results are acceptable, it’s possible to generate downloadable C code that Implements the appropriate portions of the model. Simulink could be used in external mode to monitor signals, tune parameters, and further validate and refine the model, quickly iterating through solutions [2].

- Implement a Production Prototype

At this stage, the rapid prototyping process is complete.

5.2.2 Real-Time Workshop

The Real-Time Workshop Embedded Coder is a separate, add-on product for use with Real-Time Workshop. It is intended for use in embedded systems development to generate code that is easy to read, trace, and customize for all production environment. The Real-Time Workshop Embedded Coder provides a framework for the development of production code that is optimized for speed, memory usage, and simplicity. It generates optimized ANSI-C or ISO-Ccode for fixed point and floating point microprocessors. It extends the capabilities provided by the Real-Time Workshop to support specification, integration, deployment, and testing of addresses targeting considerations such as RAM, ROM, and CPU constraints, code configuration, and code verification. The Embedded Real-Time target, provided by the Real

Time Workshop Embedded Coder, is designed for customization. In our applications ERT target is used with optimization for fixed point systems. Correct specification of target-specific characteristics of generated code can be critical in embedded systems development. The Hardware Implementation category of options in the settings menu provides a simple and flexible way to control such characteristics in both simulation and code generation [3].

5.2.3 Additional key features of TMS320F28335

The controller having the key features, A Texas Instruments TMS320F28335 device with a Digital Signal Controller and can be operated up to 150 MHz frequency, single voltage power supply (+5V) and Configurable boot load options.

Table 5.1 Key features of TMS320F28335

Sr No	Features
1	Fast, 150 MHz clock/instruction cycle.
2	High speed A/D converter, 12.5 MHz max sample rate, 16 channels.
3	12-bits.
4	A/D includes two parallel sample and hold circuits.
5	Floating point
6	Nominally a 32-bit machine
7	34 K words (16-bit) of on-chip static random access memory (RAM)
8	256 K words (16-bit) of flash read only memory (ROM).
9	6 high resolution (150 picoseconds) pulse width modulators.
10	Serial port peripherals
11	88 configurable general purpose I/O (GPIO) pins

One of Real-Time Workshop build options builds a Code Composer Studio project from the C code generated and, therefore, all features provided by Code Composer Studio work to help develop the algorithm or application [3].

5.3 Generation of PIL configuration

In these work, the Simulink is used as a friendly graphical tool for systems development, design and generation of executable code for various devices and applications. The following section presents more details of a control system design procedure using the PIL approach, Simulink tools and a DSP processor. Today there are many co-simulation tools and many are the possibilities of combination between these tools and the several processors hardware platforms. Such tools are in continuous development and making possible the coupling with several hardware platforms from different manufacturers, including additional functional facilities to become easier the task of development. Fig 5.1 shows the PIL configuration for Single Machine Infinite bus system [3].

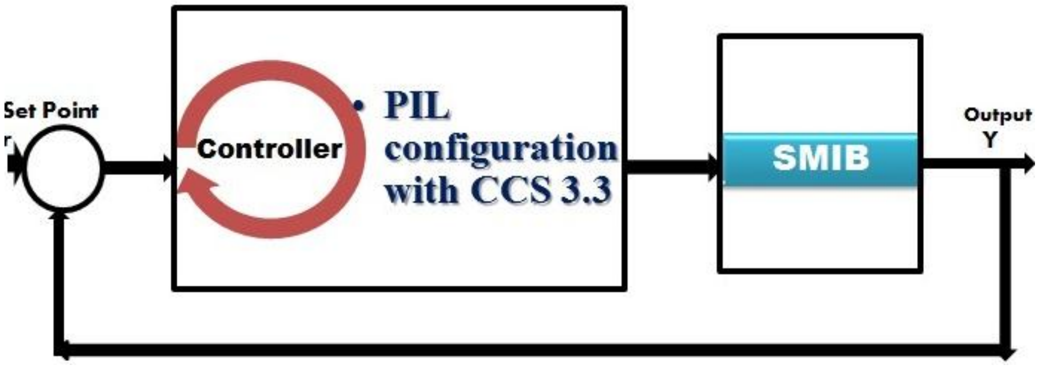

Fig. 5.1 PIL configuration for Single Machine Infinite bus system

In the PIL which uses MATLAB/Simulink environment both for a design of controller using CCS v3.3using the code generation and SMIB system to perform a PIL co-simulation together with a device. In simulations SIL and PIL were performed to obtain an attitude determination and control system (ADCS) for the microsatellite CKUTEX from Cheng Kung University. The SIL simulation is made using the MATLAB software. MATLAB software is used with the purpose of generate the code of a system. The Simulink is used together with another MATLAB toolbox called xPC Target. This toolbox allows performing prototyping, testing and development of real-time systems for running in general-purpose computers. The MATLAB/Simulink tools were used to generate executable code for the target computer from the model built in Simulink. The goal is to achieve simulation and real-time implementation of an algorithm integrating inertial navigation and GPS, using the validation and testing of the proposed algorithm in the Flight Gear flight simulator running on the host computer [2, 3]. The system i.e. multimachine is also used with the development of power system stabilizes using CCS v3.3 as shown in Fig. 5.2

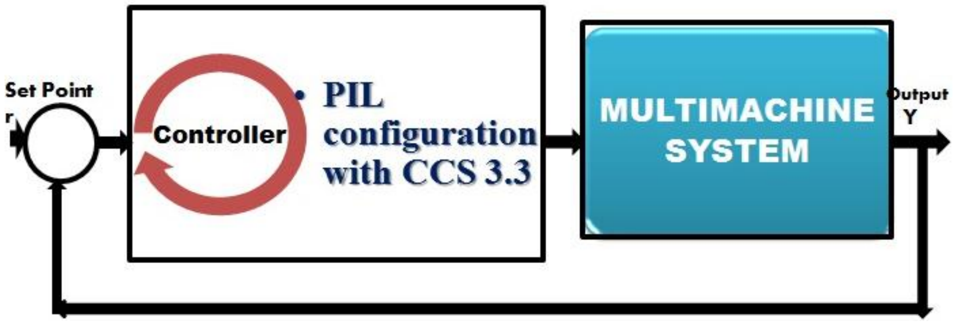

Fig. 5.2 PIL configuration for Multi Machine system

In order to exemplify the possibility of use of different development environment to perform SIL and PIL simulations of a satellite control system; in the work of a SIL simulation is made in MATRIX environment where is obtained and tested the control algorithm, simulated the space environment. This same software is used to generate the controller code, that after is downloaded in system while its model is ported to a DSP that is controlled by a computer. This computer runs the real-time simulation together with the CCS, controlling the actuators and sensors signals too, in a PIL simulation. An interface allows the user to interact with the simulation, monitoring and changing simulation parameters in real-time. In this simulations SIL, PIL and HIL can be useful in controllers and test to aerospace applications and as this simulations can be implemented in different ways, different virtual development environments and together with several hardware platforms. The results are not only obtained much more quickly but come from tests very cheap that those made in a real platform, which is often not accessible to the researchers. Furthermore, the interaction with the user in real-time and the analysis instruments, available by co-simulation tools, make the tests much more dynamic and enables the analysis of the system response in situations more realistic [2,3].

5.4 Generation of HIL configuration

Once target-specific executable is downloaded to the hardware and run it, the code runs wholly on the target and the running process can be accessed only from Code Composer Studio or from MATLAB with two powerful tools: Link for Code Composer Studio and Real-Time Data Exchange (RTDX) Link for Code Composer Studio uses MATLAB functions to communicate with Code Composer Studio and with the information stored in memory and registers on a target.Fig 5.3 & Fig 5.4 shows the block diagram of HIL configuration with SMIB & Multimachine system respectively [4].

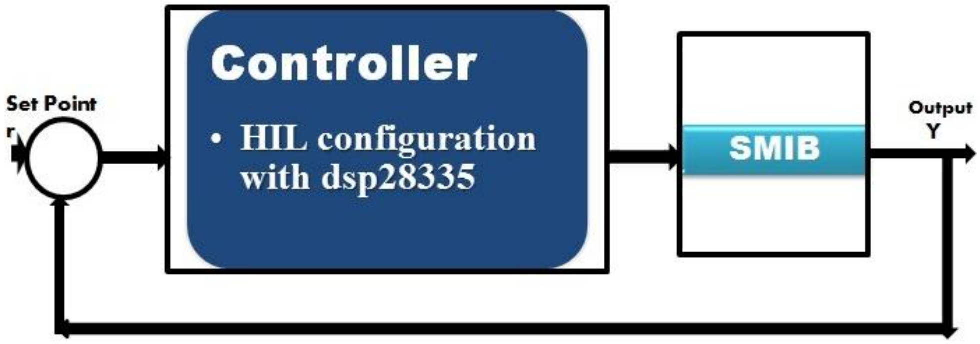

Fig. 5.3 HIL configuration for Single Machine Infinite bus system

With the links, information can be transferred to and from Code Composer Studio and with the embedded objects, information about data and functions stored on the signal processor can be retrieved. Hardware-in-the-Loop enables to write functions in MATLAB that exercise functions from project on target processor (8). From MATLAB, data can be generated, can be send to target, and a function C can be used to manipulate the data in the hardware [4].

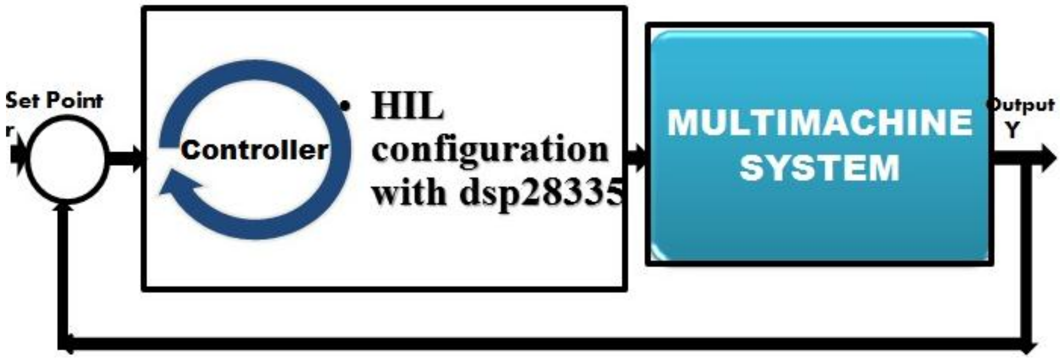

Fig. 5.4 HIL configuration for Multi Machine system

(8) Paper under preparation, "Real time implementation of power system stabilizer for SMIB system", IEEE International conference on Electrical, Computer and Communication Technologies, Feb 22-24, 2017

5.5HIL Implementation for power system stabilization

5.5.1 Single machine Infinite bus system

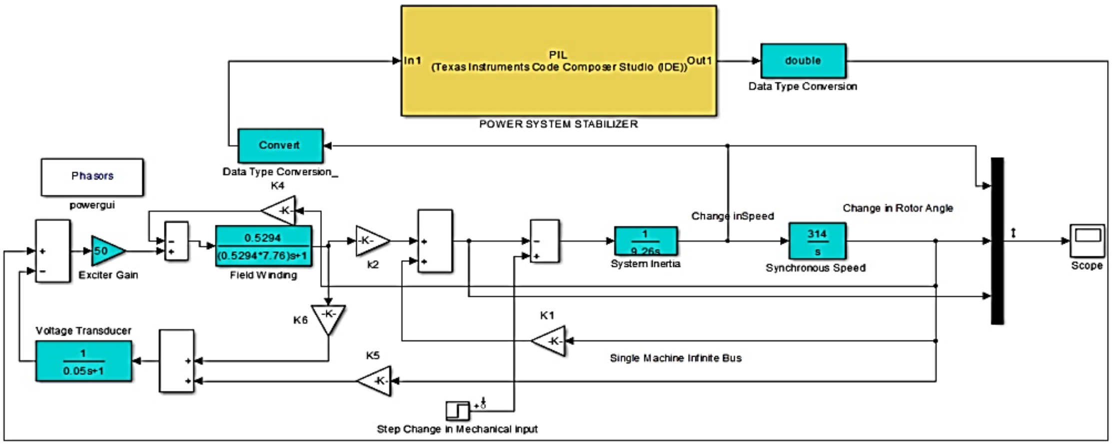

Fig. 5.5 Simulink model for the generation of PIL & HIL for SMIB system

Power system stabilizers have been developed to aid in damping these oscillations via modulation of the generator excitation. The art and science of applying power system stabilizers (PSS) has been developed over the past 40 to45 years since the first widespread application to the Western systems of the United States. To provide damping, the stabilizers must produce a component of electrical torque on the rotor which is in phase with speed variations.

Fig. 5.6 Photograph of Hardware in loop configuration for SMIB system

The PSS design is based on the linearized model of the power system (Kundur, 1994).The application of a PSS is to generate a supplementary stabilizing signal, which is applied to the excitation system or control loop of the generating unit to produce a positive damping. The most widely used conventional PSS is the lead-lag PSS. In this PSS the gain settings are fixed at certain values which are determined under particular operating conditions to result in optimal performance for that specific condition. However, they give poor performance under different synchronous generator loading conditions. The CPSS is simulated in the Simulink environment with the code composer studio for the hardware in loop implementation. The GPSS is used to add damping to the rotor oscillations of the synchronous machine by controlling its excitation. The GPSS design is based on the linearized model of the power system (Kundur; 1994)However the power systems are highly non-linear systems, with configurations and parameters that change with time and the GPSS cannot guarantee its performance in a practical operating environment [5,6].This graphs show the comparison of the various stabilizers used in order to determine which produces the best performance in terms of stabilitywhenever a fault occurs. The comparison was made by looking at the oscillation and also the time taken by each stabilizer to achieve desired value and maintain stability after system subjected to disturbances.

5.5.2 Multimachine system

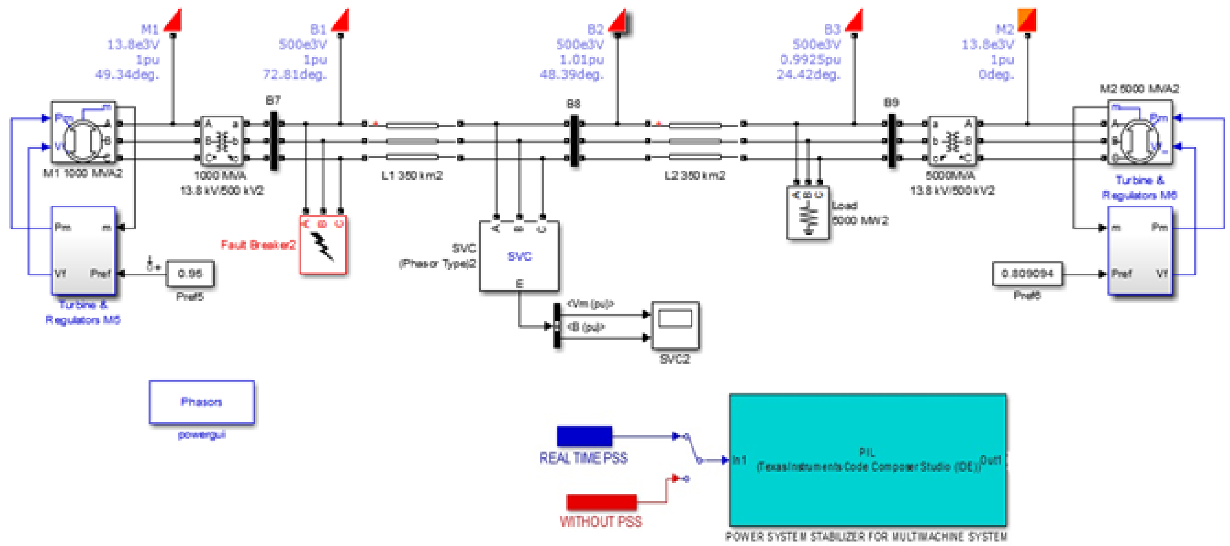

Fig. 5.7 Simulink model for the generation of PIL & HIL for multimachine system

Fig shown in 5.7 shows that power system stabilizer is being implemented with the code composer studio 3.3 i.e. for processor in loop generation and ezdsp28335 is used with PSS for hardware in loop implementation for single line to ground fault. This study shows that it is possible to stabilize power system whenever a fault occurs within the shortest time possible and therefore making the power system more reliable. The settling time reduced after the system subjected to different disturbances [8-10]. The desired value of the machine output coming in a very short time compared to the conventional stabilizer.

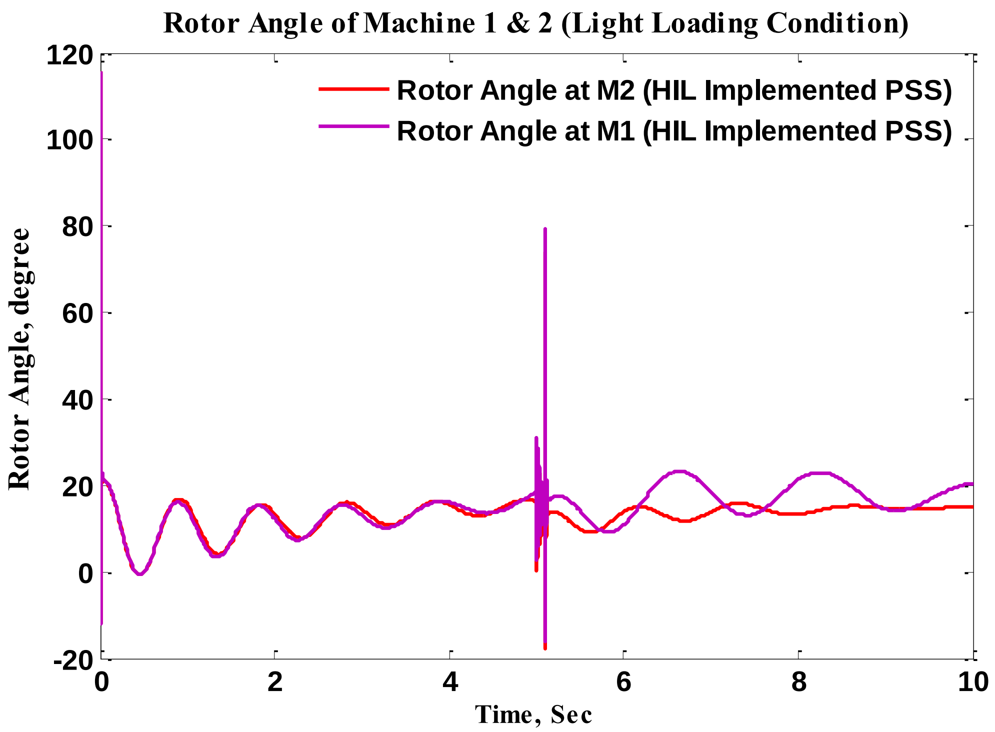

Fig. 5.8Rotor Angle Variation for the generation of HIL Implemented PSS for multimachine underLight Loading Condition

From Fig 5.8 depicts that when the PSS is connected through dspez28335 the variations in the rotor angle gets minimized. The Real Power of Machine 1 and the Real Power of Machine 2 also increases and the maximum amount of power can be transferred through the transmission line.

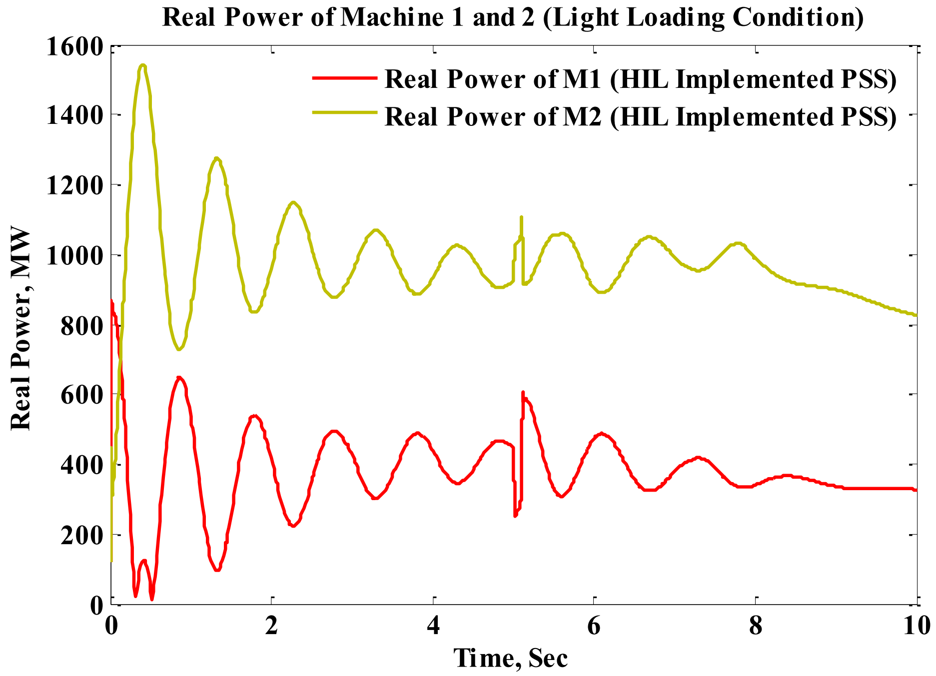

Fig. 5.9Real Power M1 for the generation of HIL Implemented PSS for multimachine under Light Loading Condition

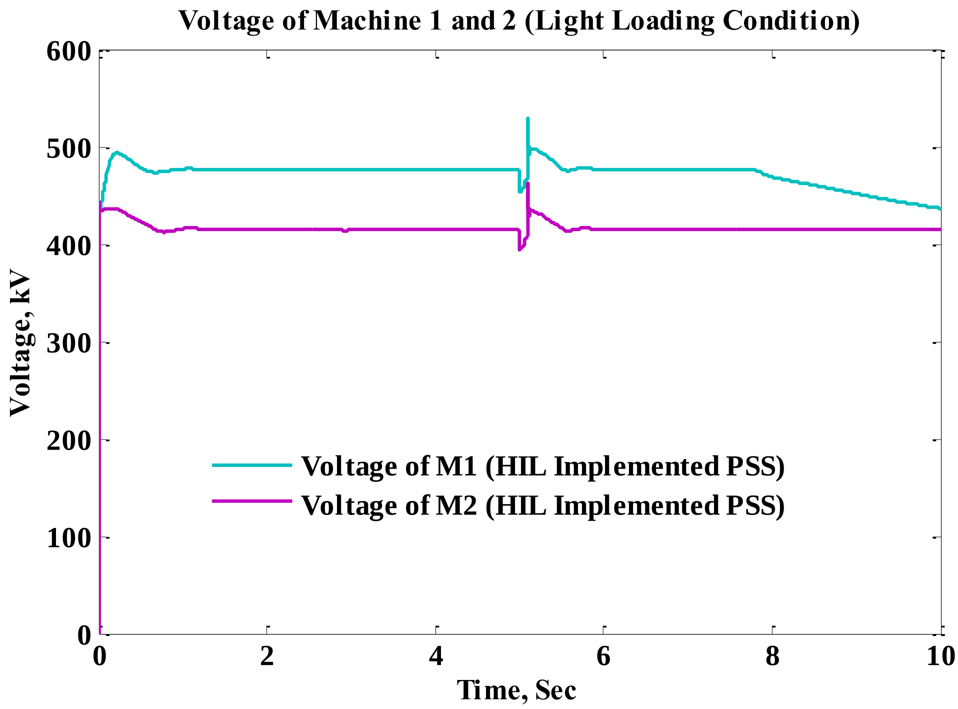

Fig. 5.10Voltage of M1 and M2 for the generation of HIL Implemented PSS for multimachine under Light Loading Condition

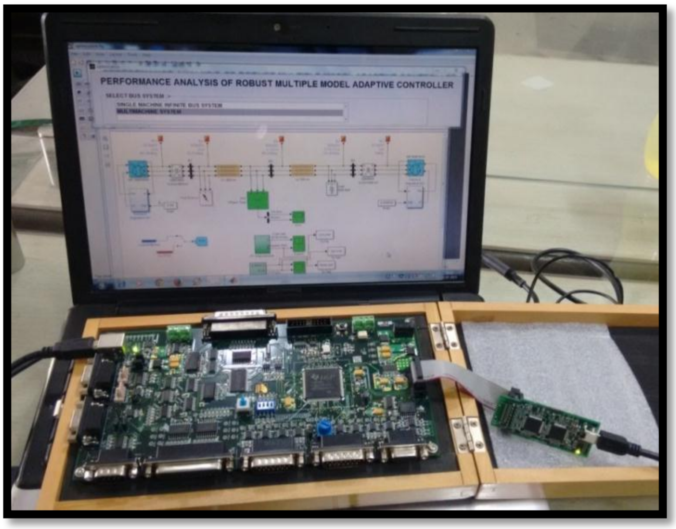

Fig. 5.11Photograph of Hardware in loop configuration for Multimachine system

In this simulation, multi-machine power system is demonstrated under a single line to ground fault simulation and then cleared with opening breaker on line which fault occurred [11]. Disconnecting one of two tie-line transmission lines can change the area power transfer level into singlelinepower transfer level. System will oscillate to its newstable point, during that time system parameters will deviate [12].

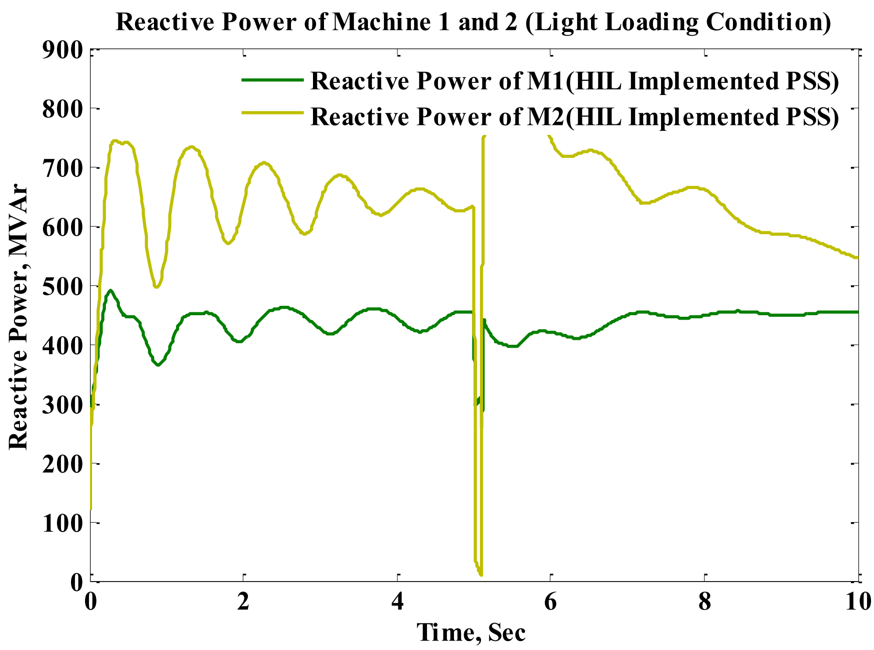

Fig. 5.12Reactive Power of M1 and M2for the generation of HIL Implemented PSS for multimachine under Light Loading Condition

Fig. 5.12shows the performance of HIL configuration of PSS for Reactive Power of machine when single line to ground fault occurs in transmission line [13-15]. The multi-machine power system has achieving the stability state in 7s, although the system has oscillating in 3s.

The HIL configuration of PSS needs to improve in order to stable the multi-machine power system more robust [16-18]. The PSS has successfully created the stability of multi-machine power system in 3s, although the system has oscillating in 2s. The time for stability is faster than conventional PSS [19-20]. Therefore HIL based PSS is more robust than other PSS in order to achieve thestability of multi-machine power system.

5.6 Concluding Remarks

The development of applications for embedded systems, as well the design of controllers to be performed by dedicate processors, can be immensely facilitated using the co-simulation approaches such as the Processor-In-The-Loop and Hardware-In-The-Loop. This validation scheme has been used as a way to move beyond on strictly computer simulations, opening the studies to realistic problems related to communication and exchange of data between embedded processor and controlled system, e.g. time delays, reliability of transmitted and received data, and processing time of the controller.