Chapter 3

Conceptual Design of MIMO Wireless Systems

Conceptual Design of MIMO Wireless Systems

3.1 Introduction

The ever-growing demands for high-speed data and multimedia services are the driving forces behind the requirements for future wireless communication systems. Next-generation communication evolution towards 4G promises to meet the demands for ubiquitous communication. Hence, the complexity of communication systems has grown dramatically in search for higher system capacity, higher data rates, and bandwidth efficiency. To meet the demand for ubiquitous communication and the ability to access and share information, wireless appliances and the supporting networking infrastructure must be able to integrate complex algorithms with adequate computing and signal processing capabilities. In order to meet the capacity needs for future wireless systems without increasing the required spectrum, accomplishment of implementation of advanced communication techniques is necessary. MIMO promises high bit rates, small error rates, reliability, increased channel capacity over rich scattering wireless channels without consuming extra bandwidth or transmit power when compared to conventional single antenna technologies [1].

Software simulations provide flexibility, but the true performance of the system can only be known by developing a hardware wireless platform and performing measurements and tests in the target environment. Recently, various MIMO wireless testbeds based on DSP and FPGA have been developed to verify the theoretical performance gains and to investigate practical issues in MIMO implementation. Overview of various MIMO Testbed for physical layer of MIMO communications is given in [2]. MIMO Communication systems are realized by computationally complex algorithms, requiring new digital hardware architectures to be developed. This chapter describes the System Design Methodology and Basic Hardware Concept for designing MIMO Wireless platform development. Mathworks model based design and MATLAB based programming and verification tools for FPGAs and DSPs are also discussed briefly in this chapter.

3.2 System Design Methodology

Next-generation communication systems promise to deliver a wide variety of new features, such as improved battery life, smaller size, high-definition video and high-bandwidth Internet connections. In designing of such systems, the development and integration of several computationally-intensive algorithms which enable these new features are to be incorporated. In spite of their performance enhancing capabilities, most of the research on MIMO technology up to the moment is based on theoretical studies. To verify the laid concepts of any newly developed system or standards, Rapid Prototyping (RP) implementations is the most popular methodology [3,4].

Rapid Prototype implementations and field trials are an essential part of the verification of new system concepts and standards. Consequently, extensive prototyping efforts have been carried out for current technologies such as WLAN [5] and 3GPP/LTE [6]. A prototype is the initial realization of a research concepts, either as a reference, or as a vehicle for future developments and improvements. The software simulations of the developed system based on research requirements often make numerous assumptions and depend on mathematical models. Examples are the assumption of perfect channel knowledge at the receiver or mathematical model of wireless channel. From an implementation point of view, prototyping efforts have the advantage of pointing out complexity issues early in the design cycle of a newly developed systems. In RP approach, to develop real-time algorithms on testbed, first the simulation implementation is done, then the simulation is migrated to the testbed and, finally real-time implementation is obtained. The speed and complexity of MIMO-based system requires comprehensive design and verification process including both hardware and software. Experiments in real-world scenarios by means of hardware implementations are necessary to measure the actual performance of system. Figure 3.1 describes the System Design Methodology for MIMO Wireless Platform. The scope of this design methodology extends from specification to implementation on Wireless Platform. According to Functional Description or Specifications various system modeling tools, software tools are chosen and based on hardware specifications, hardware simulation tools and verification is carried out and finally the Wireless platform is developed for implementation of Wireless Communication System.

For RP, design time of a new system is more critical than other factors like cost and power consumption. A major focus is therefore on the efficiency of the tools that are being used in the development process. Based on the system specifications, system modeling and simulation tools like MATLAB Simulink is used for system simulation analysis. Once the system is analyzed in Simulink, the hardware simulation and verification is carried out and finally the integration, measurement and testing is done.

3.2.1 Hardware Concept of MIMO Wireless System

The past decade has shown distinct advances in the theory of MIMO techniques for wireless communication systems. Now, the time has come to demonstrate this progress in terms of applications, where the intermediate step towards a customized product consists of more or less rapid prototyping. Due to the multitude of different MIMO schemes and different applications and standards, an ideal prototyping platform requires a high degree of flexibility and modularity in order

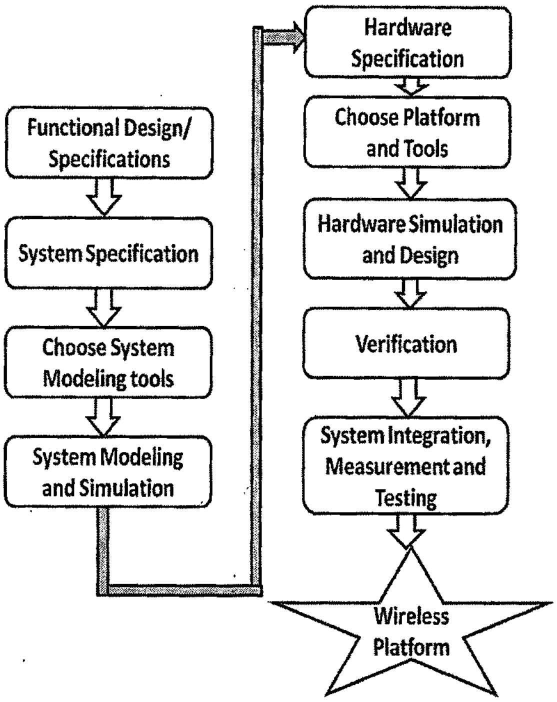

Functional Design/ Specifications
System Specification
Choose System Modeling tools
System Modeling and Simulation
Hardware Specification
Choose Platform and Tools
Hardware Simulation and Design
Verification
System Integration, Measurement and Testing
Wireless Platform

Figure 3.1: System Design Methodology

to be qualified for a wide range of potential applications. MIMO implementation is still facing a lot of challenges. There are different hardware processing platforms available for implementing high performance algorithms for MIMO Wireless system. Following are the processing platform options [7]:

- General purpose processors (GPP),
- Digital signal processors (DSPs),
- Field programmable gate arrays (FPGAs) and
- Application specific integrated circuits (ASICs),

Hardware implementation of the wireless system can be achieved using GPPS, but it faces difficulties when dealing with high processing systems. FPGAs are suitable for fast implementation, quick hardware verification and is re-programmable, whereas ASIC is designed for a particular application. DSP are suitable for complex computation of digital signal processing algorithms. The integration of powerful FPGAs and DSPs is feasible for testing MIMO algorithms for real-time systems as we can get advantage of both processing units. DSP provides a specialized core with multiple-functional units, which are optimized for digital signal processing operations like filtering

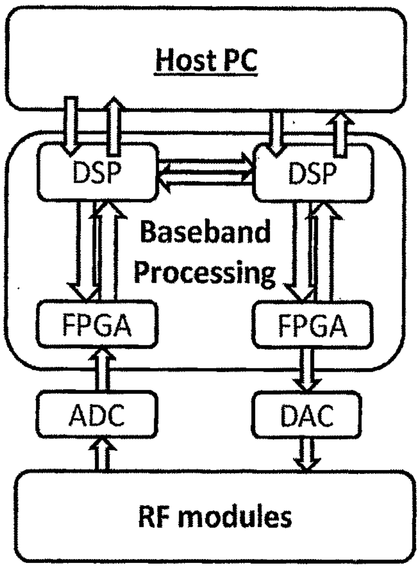


Figure 3.2: Basic Hardware Architecture of MIMO

and transformations. FPGA has the ability to provide special hardware structures in parallel, and to handle high data rates without affecting the other algorithms. It is also convenient to split the real-time implementation into several steps: firstly, a fixed-point code is implemented on DSP and, afterwards, the software modules that do not meet time requirements are migrated to FPGA [8,9].

Figure 3.2 shows the Basic Hardware Architecture of MIMO Communication System. It consists of two units: Analog RF Unit (RF) and Baseband Processing Unit (BB). The first stage consists of the RF modules mixing the signals of each antenna from RF to BB following the receiver path, the next stage consists of the Analog-to-digital converter (ADC) and Digital-to-analog converter (DAC).

The succeeding FPGA stage allows high-speed parallel data processing for various processing tasks. Therefore, the FPGA stage covers the complete MIMO front end, i.e. antenna filtering operations, the synchronization of time and frequency, the signal modulation and demodulation, and all other regular processing operations. The FPGA operates on the incoming high-data-rate digital signal and reduces the data-rate step wise, so that the subsequent DSP stage is not over strained.

In the DSP stage, all less regular MIMO processing takes place, i.e. algorithms which are under study. A'nong the hardware platform and its components, a rich set of software tools is required to develop DSP- and/or FPGA-based applications in an efficient and fast way.

3.2.2 MIMO Wireless Testbeds and Prototypes

Since the foundation of MIMO Wireless Communication Systems, considerable effect is dedicated to the demonstration of the capabilities of the technology to investigate the performance of MIMO in real-world scenarios . Hence, various demonstrators and testbeds for MIMO communication has been developed [10]. Testbeds involves real-time transmission in wireless environment and thus allows experiments with physical channels and RF Frontends. Whereas RP involves designing of transmitters and receivers hardware architectures for future products [11]. In 1998, Foschini and a team from AT& T Research Labs developed the first prototype using VBLASt Technique [12], to verify the predicted gains of Spatial Multiplexing MIMO Technique.

The testbeds mainly consists of DSPs, FPGAs combined with ASICs for performance-critical system components, RF frontends equipped with number of antennas and interfacing to the digital baseband transceiver. The MIMO Wireless channel is either physical channel or is implemented using hardware radio frequency emulator. The signal is processed either in real-time mode or offline-mode. Various demonstrators and testbeds developed have been proven to be initial research tools for the development of new technologies [13].

Various MIMO Communication Testbeds that provide physical layer functionality are based on the the Universal Software Radio Peripheral (USRP) from Ettus Research [14] together with the GNU software radio [15], the wireless open-access research platform (WARP) from Rice University [16] or OpenAirInterface systems developed by EURECOM [17]. Since then a number of universities have developed their own testbeds and experimental platforms to perform MIMO measurements and real-time experiments. Many of these systems are based on offline processing of sampled data and use commercially available prototyping platforms. These again are often intended to perform channel measurements and to verify the performance of algorithms considering implementation issues. However, some implementations are also capable and specifically designed for real-time operation to consider implementation complexity.

3.3 Model-Based Design for Rapid Prototyping

The development of embedded systems consists of a prototyping phase for feasibility studies, testing and verification of final product. Prototyping of embedded software/hardware systems is essential as it shortens the path from specifications to the implementation on target hardware [18].

Embedded Systems development organizations are seeking to adopt Model-Based Design to take advantage of enhanced ability to deal with complexity, reduce time-to-market, reduced cost, and improved quality [19]. MATLAB, Simulink and various signal processing tools provide an integrated workflow for verifying, prototyping and implementing wireless communication systems. Algorithm development for hardware targets involves converting the high-level concept codes into a version that uses low-level arithmetic or logical operations. Many rapid prototyping platforms are available such as DSP, FPGA, or mixtures of DSP and FPGA [20–22]. Various software simulation tools are provided by the manufacturer to ease the development of communication system. These tools can be integrated with MATLAB Simulink to provide a path from block diagram to system integration such as Xilinx System Generator and Link to CCS/Real-time workshop.

The ability to efficiently construct models combined with associated tools and systematic methodologies primes Model-Based Design for success by providing a complete solution that enables concurrent engineering, performance analysis, automatic test generation, building efficient specifications and execution models, code generation and optimization, and automatic refinement through different abstraction levels [23]. Following sections discusses the Mathworks model based design methodology, software simulation tools and design workflow for DSP and FPGA.

3.3.1 Mathworks Model-based Design

Model-Based Design is a systematic method to generate test cases from models of system requirements. It allows evaluation of requirements independent of algorithm design and developments. For embedded deployment automatic codes can be generated and test benches can be created for system verification using available processing tools. Model-Based Design [24] offers an efficient and cost-effective way to develop complex embedded systems for a variety of applications. Model-Based design involves using Computer Aided Engineering (CAE) Tools to simulate system behavior, verifying research requirements, designing system models, generating software for prototyping and continuous testing on hardware target throughout the development process.

Embedded C Code Generation and Verification (DSP)	**HDL Code Generation and Verification (FPGA)**
Embedded Target for TI C6000 DSP, Link for Code Composer Studio, Real-Time Workshop, Real-Time Workshop Embedded Coder, Stateflow Coder	Link for ModelSim, Filter Design HDL Coder Synplify DSP (Synplicity) Xilinx System Generator for DSP DSP Builder (Altera)

Table 3.1: MATHWORKS Products for Rapid Prototyping on DSP and FPGA

MATLAB provides various tools for Model-Based Design of FPGA and DSP. Table 3.1 lists MATHWORKS products for Embedded C-Code generation for DSP and HDL Code generation and verification for FPGA. Figure 3.3 shows the design methodology for Mathworks Model based

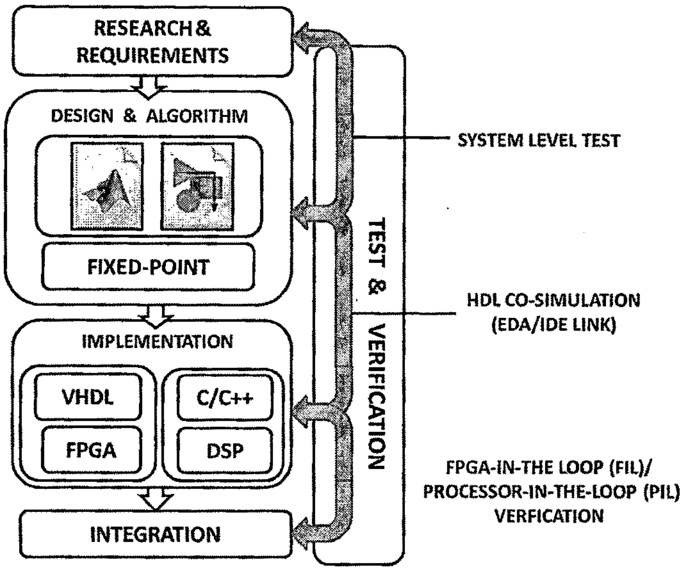


Figure 3.3: Model Based Design Methodology

design, the test and verification of algorithms developed for DSP and FPGA including the Hardware Co-simulation, test and verification on the target hardware.

3.3.2 MATLAB Design Tools for FPGA

Using FPGA for complex systems enables engineers to rapidly evaluate algorithm and architecture tradeoffs quickly. They can also test designs under real world scenarios without incurring the heavy time penalty associated with HDL simulators [25]. FPGA are used for prototyping ASIC workflows for hardware verification and early software development. To improve the performance running high-throughput, high-performance applications, algorithm designers are increasingly using FPGAs to prototype and validate innovations instead of using traditional processors [26]. However, many algorithms are implemented in MATLAB due to the simple-to-use programming model and rich analysis and visualization capabilities. For FPGA development, several design suites are available. Targeting at Xilinxs FPGAs, three major players are well-known: Xilinx Corporation with its Integrated Software Environment (ISE) [27], Mentor Graphics Corporation with its FPGA advantage suite [28], and Synplicity Inc. with its synthesis and verification solutions [29]. Model-Based Design using HDL code generation enables engineers to efficiently produce FPGA prototypes.

The process for translating MATLAB designs to FPGA target hardware consists of following steps:

1) Model the algorithm in MATLAB Simulink

2) Convert the model to Fixed point using Fixed point Toolbox

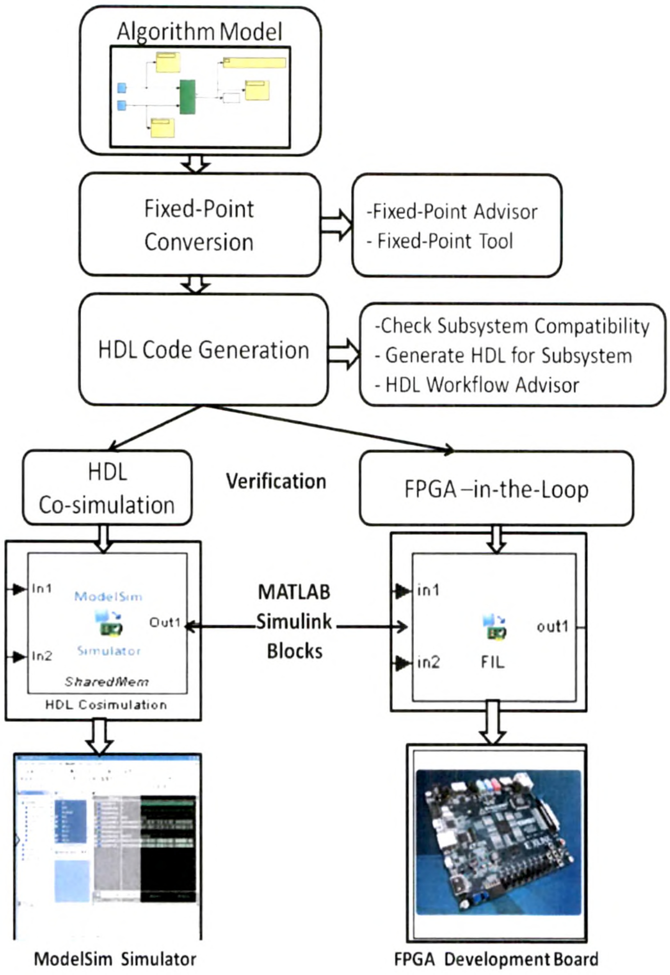


Figure 3.4: Programming and Verification steps for FPGA

3) Use HDL Workflow Advisor to generate the HDL code of the algorithm

4) Use HDL Co-simulation Wizard with Modelsim for HDL Verification

5) Use FPGA-in-the-Loop to download the code to target, for testing and verification in loop.

Figure 3.4 shows the steps for Programming FPGA using HDL coder Workflow Advisor. Electronic Design Automation (EDA) Simulator Link [30] Toolbox provides a co-simulation interface between MATLAB Simulink and HDL simulator Mentor Graphics ModelSim. Using EDA Simulator Link we can verify a VHDL implementation against Simulink model or MATLAB algorithm. HDL co-simulation enables engineers to reuse Simulink models to drive stimuli into the HDL simulator and perform system-level analysis of the simulation output interactively. While HDL simulation provides only digital waveform output, HDL co-simulation provides complete visibility into the HDL code, as well as access to the full suite of system-level analysis tools of Simulink. When engineers observe a difference between expected results and HDL simulation results, co-simulation helps them to better understand the system-level effect of the mismatch.

FPGA based verification (also referred to as FPGA-in-the-loop (FIL) simulation) of the algorithm increases confidence that the algorithm will work in the real world. Implementation of HDL code on FPGA boards to enable FIL verification for running test scenarios faster. This approach ensures that the algorithm will behave as expected in the real world. This enables engineers to run test scenarios faster than with host-based HDL simulation.

3.3.3 MATLAB Design Tools for DSP

For DSP-based development [31], Mathworks offers with Simulink and the Real-Time Workshop (RTW) [32] a seamless tool for fast prototyping in DSP based systems. This approach also offers the possibility of hardware-in-the-loop simulation, where parts of the application may run on the target DSP and others run on the host computer. Other methodologies, such as The Mathworks Real-Time Workshop in combination with the Embedded Target for the TI TMS320C6000 DSP Platform [33] allow mapping a Simulink block diagram composed of pre-existing algorithm blocks to a single DSP. Texas Instruments (TI) Code Composer Studio (CCS) is an Integrated Development Environment (IDE) for TI embedded processors . It comprises of compilers for each TI's device families and suite of tools like source code editor, project build environment, debugger, real-time operating system for development and debugging the developed embedded applications.

As shown in Figure 3.5, the DSP Design flow for programming and verification of developed algorithm on DSP. For the algorithm developed in MATLAB Simulink, the C/C++ code is generated using Code Generation Tools and Embedded Coder Toolbox. Target Preference is set to the Texas Instruments Code Composer Studio and the DSP Target Board used for verification.

Once the C code is generated and the subsystem for the algorithm is created, Software-in-the-Loop (SIL) or Processor-in-the-Loop (PIL) is performed for verification. With SIL simulation, we can verify the behavior of source code of the developed algorithm on the host computer. In SIL mode, the simulink executes the referenced model by generating the production code using the model reference target preference set in the model. The code is compiled in the C Compiler and verification is carried out. SIL mode is a convenient option to verify the code when the target hardware is not available. With PIL simulation, we can verify the compiled object code that we want to deploy, the object code runs on the real target hardware or on instruction set simulator. In PIL mode, the C code generated for the algorithm model is cross-compiled and executed on the

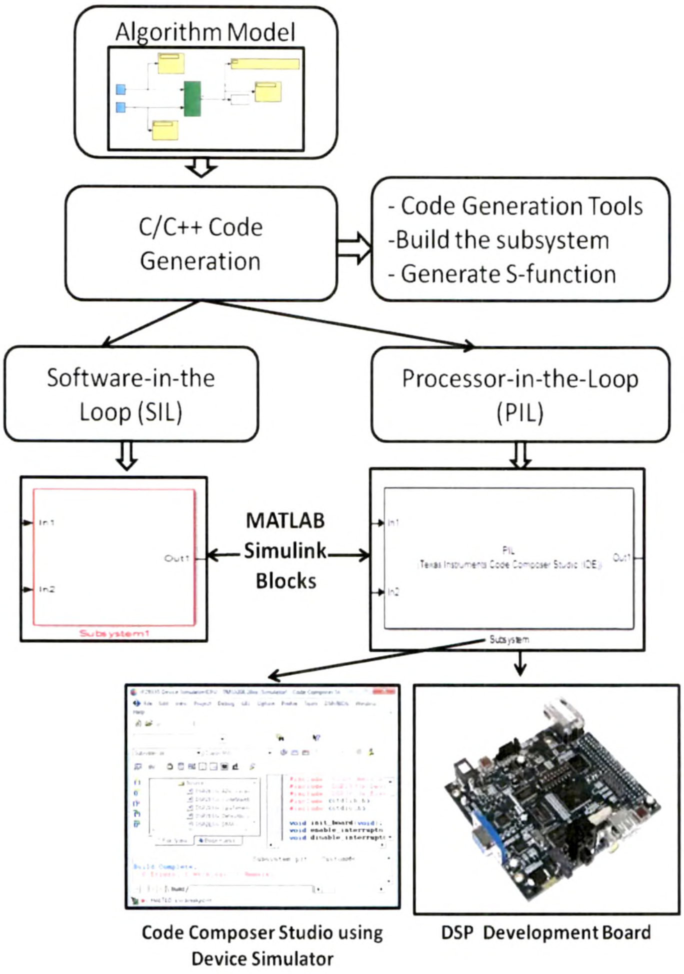


Figure 3.5: Programming and Verification steps for DSP

target processor or an equivalent instruction set simulator.

3.4 Case Study: Wireless control of FTE Robot

Wireless communication systems have range of applications for mobile robots. It includes industrial applications, agricultural robotics applications, underwater applications, and numerous military applications [34]. Wireless Sensor Networks are developed for search and rescue applications, with the use of mobile robot integrated with wireless camera [35]. Wireless control of mobile robot is done by Radio frequency (RF) Communication. Numerous wireless technologies like Zigbee and Bluetooth can be used to control the robot through remote computer, joystick etc. Here an attempt has been made to control Mobile Robot using wireless modules, to gain in-depth knowledge about wireless concepts and implementation challenges of working with practical wireless applications.

The Faculty of Technology and Engineering (FTE) Robot is a mobile device which has features like Line Follower and Maze Follower. The Robot is featured around C8051F340 microcontroller, which is reprogrammed to establish a wireless communication link between Robot and remote computer. Specific commands are generated to control the movements of Robot. It is possible to control two robots simultaneously by control commands from remote computer. This section discusses the design and implementation of Wireless control of FTE robot through remote computer It gives overview of the FTE Robot and Wireless module specifications.

3.4.1 Overview of FTE Robot

FTE robot, as shown in Figure 3.6, is developed by Mr. Jagdish Sanghani for educational purpose for Department of Electrical Engineering, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, India. The robot can be used for research in Embedded systems and Robotics. It focuses on the microcontroller C8051F340A [36] from SILABS. Figure 3.7 shows the back view of the FTE Robot with all the hardware connections.

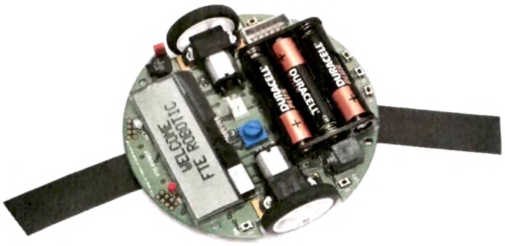


Figure 3.6: Front View of FTE Wireless Robot

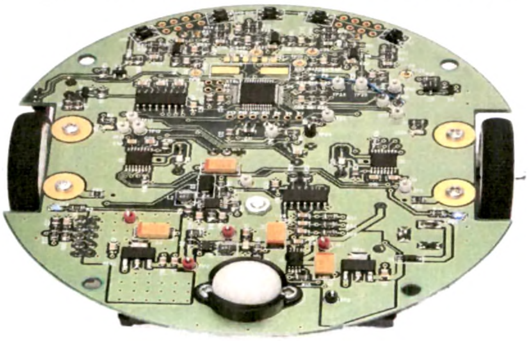


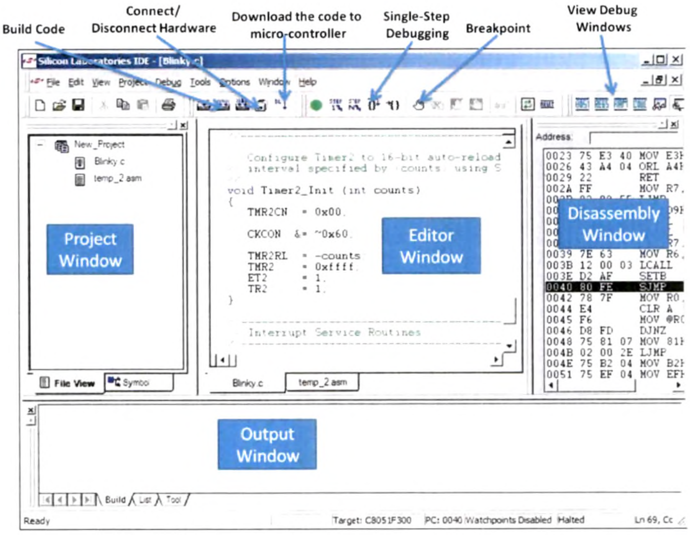
Figure 3.7: Back View of FTE Wireless Robot

The specifications of various components in FTE Robot is listed in Table 3.2. The robot con-

Components	Specifications
C8051F340A	8051-compatible microcontroller core, 64 kB Flash, ADC,SMBus, 6x PWM 2x UARTs, I2C, SPI, Four general purpose 16-bit counter/timers,
I2C based LCD	16x2
Micro metal Geared DC motors	Gear ratio-30:1, Free-run speed at 6V: 440 rpm
IR Sensor	Tolerance of + -0.15mm.
LEDs	Green, Red and Blue.
Full-Bridge Motor Driver	peak output currents up to + -2.8 A and operating voltages up to 36 V
PIZO Sounder	Sound Output 90(dB Min Typ)
On-board battery	3AAA batteries (1.5V each)

Table 3.2: FTE Robot Specifications

sists of various components like IR sensors, micro metal geared motors, I2C based Liquid Crystal Display (LCD), Pizo sounder, user pushbuttons, Light Emitting Diode (LED) and various other components. It consists of on-board Joint Test Action Group (JTAG) debugger for testing the robot operation using HyperTerminal. It also consists of LEDs to show the ON/OFF status of the power supply.



```
   Configure Timer2 to 16-bit auto-reload
   interval specified by  counts  using S

void Timer2_Init (int counts)
{
   TMR2CN   = 0x00.

   CKCON  &= ~0x60.

   TMR2RL   = -counts
   TMR2     = 0xffff.
   ET2      = 1.
   TR2      = 1.
}

   Interrupt Service Routines
```

```
0023 75 E3 40 MOV E3H
0026 43 A4 04 ORL A4H
0029 22       RET
002A FF       MOV R7,
0039 7E 63    MOV R6,
003B 12 00 03 LCALL
003E D2 AF    SETB
0040 80 FE    SJMP
0042 78 7F    MOV R0,
0044 E4       CLR A
0045 F6       MOV @R0
0046 D8 FD    DJNZ
0048 75 81 07 MOV 81H
004B 02 00 2E LJMP
004E 75 B2 04 MOV B2H
0051 75 EF 04 MOV EFH
```

Figure 3.8: Snapshot of SiLabs IDE

The FTE robot is programmed using Silabs IDE, which has Keil C compiler [37, 38]. The snapshot of the Silabs IDE is as shown in Figure 3.8. Using the IDE the programs can be debugged, and downloaded to the micro-controller using USB Debug Adapter. All the programs to control the

Robot movements, motors speed and the direction of motors are written in C language [39]. The testing of the programs is done using the Hyperterminal [40] connected to the Robot for debugging purpose. Functions are written for each command and they are executed using the main file. The program is initially tested on C8051F34X Development Kit [41].

3.4.2 Wireless Module

The wireless module is a general-purpose programmable module, which features 2.4 GHz radio and Universal Serial Bus (USB) from Pololu Robotics and Electronics [42]. The wireless module is based around the CC2511F32 microcontroller from Texas Instruments [43], which has an integrated radio transceiver, 32 KB of flash memory, 4 KB of RAM and USB interface. Table 3.3 lists the Technical Specifications of wireless module. The Wireless module can be connected using USB connection to configure the module and to transmit and receive data. USB connection also provides power to the module.

CC2511F32	2.4 GHz system-on-chip (SoC), 32 kB of Flash, 4 kB of RAM 8051 MCU, 7 12 bit ADC, Two USARTs.
Radio	Frequency: 2400 2483.5 MHz, with 256 channels Range: approximately 50 feet (under typical conditions indoors) Bit rate: programmable, up to 350 kbps Effective data rate: up to 10 KB/s
Operating current	up to approximately 30 mA
Operating voltage	2.76.5 V

Table 3.3: Technical Specifications of wireless module

Wireless module is connected to the micro-controller using UART for serial communication. Figure 3.9 shows the FTE Robot with the Wireless module connection. The Wireless Module is connected to the Robot with UART and power supply connections. To connect micro-controller to wireless module, GND, TX and RX connections are made to establish a connection link between FTE Robot and wireless module for communication as shown in Figure 3.10. The wireless module is configured using Wixel Configuration Utility. We can read, write or configure wixel module using this utility. Using Application Configuration the specifications like baud rate, serial mode and radio channel is configured. For this particular application, Wireless Serial Application is used to control robot using remote computer.

Wireless Serial application connects two wireless modules together to make a wireless bidirectional link. It uses RF bit rate of 350 kbps and can reach a range of approximately 50 feet under

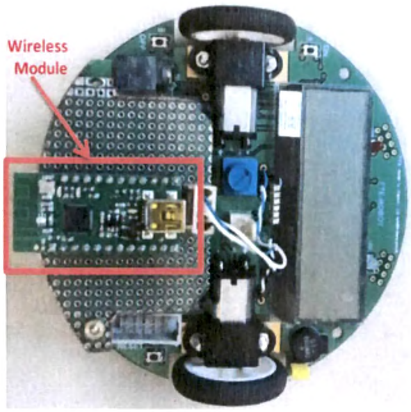


Figure 3.9: FTE Robot with Wireless module connection

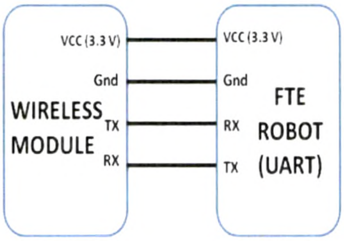


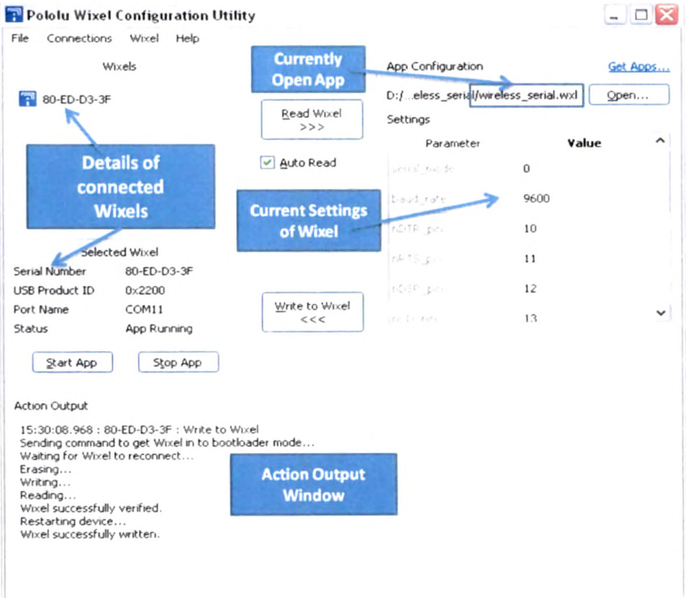
Figure 3.10: Connections between FTE Robot and Wireless Module

typical indoor. conditions. Applications for wireless module can be developed using Eclipse IDE for C/C++ Developers.

The programming of Wireless Module was done in Eclipse IDE [44]. The Wireless Serial Application was used to control robot movements. Pololu Wixel Configuration Utility as shown in Figure 3.11 shows the snapshot of Wixel Configuration Utility, with the App configuration for the Wireless Serial Application. Baud rate is set to 9600 and the radio channel is set to 128. Serial mode 0 is selected which is the Auto-Detect Serial mode and it automatically choose the serial mode based on how the module is being powered. Various parameter like baudrate and radio-channel parameter are adjusted. For this particular case study the baud rate of the transmitter, receiver wireless modules and that of robot's UART is set to 9600 bps.

3.4.3 Implementation

Once the FTE Robot and the wireless modules are programmed, the entire system is implemented as shown in Figure 3.12. The FTE Robot movements in desired direction is controlled by the commands from the wireless module. The transmitter wireless module which gives direction to robot is connected to the USB port of remote computer. The movement of robot is controlled by



80-ED-D3-3F

Selected Wixel

Serial Number	80-ED-D3-3F
USB Product ID	0x2200
Port Name	COM11
Status	App Running

App Configuration

D:/...eless_serial/wireless_serial.wxl

Settings

Parameter	Value
serial_mode	0
baud_rate	9600
nDTR_pin	10
nRTS_pin	11
nDSR_pin	12
nCD_pin	13

Action Output

```
15:30:08.968 : 80-ED-D3-3F : Write to Wixel
Sending command to get Wixel in to bootloader mode...
Waiting for Wixel to reconnect...
Erasing...
Writing...
Reading...
Wixel successfully verified.
Restarting device...
Wixel successfully written.
```

Figure 3.11: Snapshot of Pololu Wixel Configuration Utility

controlling the direction and speed of motors which control the wheel motion. The commands are given with the help of numeric keypad and the list of commands is listed in Table 3.4. The receiver wireless module is connected to the FTE robot through UART as discussed in Section 3.4.2.

To remotely control robot from remote computer, Wireless Serial application is used [45]. It turns pair of wireless modules into wireless USB/TTL serial link for communication between a remote computer and a micro-controller (in our case c8051F340 on FTE Robot).

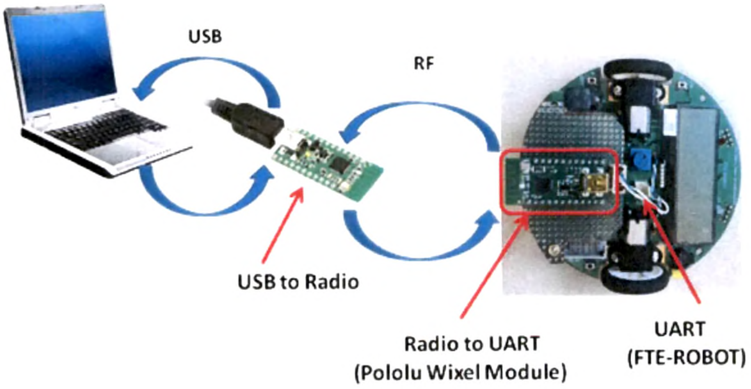


Figure 3.12: Implementation of Wireless Control of FTE Embedded Robot

Using the wireless module the remote computer is successfully able to control the FTE Robot movements. The commands for the Direction, Start and Stop for robot are successfully working.

Commands From PC	Operation By Robot
0	Robot-Reverse
1	Robot-45-degree-Left
2	Robot-180-turn
3	Robot-45-degree-Right
4	Robot-left-turn
5	Robot-stop
6	Robot-right-turn
7	Robot-90-Degree-left-turn
8	Robot-Forward
9	Robot-90-Degree-right-turn

Table 3.4: List of Commands to control FTE Robot

When two FTE Robots are connected with wireless modules, both robots are simultaneously controlled by only single remote computer commands. Wireless Networks with more than two FTE robots can be remotely controlled by remote computer. Also two robots can communicate with each other and transfer of data can be achieved. As further development, for search and rescue operation application camera and voice recorder can be embedded on the robot.

3.5 Concluding Remarks

This chapter discusses the system design methodology flow for designing Wireless Platform for experimental evaluation of developed algorithms. Rapid prototyping of MATLAB based model based design for algorithm implementation on DSP and FPGA in briefly described. MATLAB based PIL Simulation for DSP and FIL Simulation for FPDA design and development is described. PIL and FIL modes are used in this research work for verification of optimization algorithms developed for MIMO Wireless Communication System. Wireless Control of FTE Robot was carried out, and their design flow and steps are discussed in the chapter. The practical implementation of Wireless System gave technical experience and knowledge to author to work with Wireless Communication Systems.