System Requirements Specification for : Controlling Firebird V using Kinect

Smit Patel - 09005002
Priyank Parikh - 09005006
Nitant Vaidya - 09005007
Lalit Swami - 09005028

March 2, 2012

under the guidance of
Prof. Kavi Arya

Embedded and Real-Time Systems Laboratory
Department of Computer Science and Engineering
Indian Institute of Technology
Bombay

Contents

1 Introduction

The main idea of this project is to provide an interface through which we can control the Firebird V bot using hand-gestures which are captured using a Microsoft Kinect device. It will allow user to play with Firebird V in a totally different way in which user doesn't have to write code to control the bot. Different hand-gestures captured by Kinect are mapped to functions to control the motion of Firebird. Esterel language is used as an intermediary to convert movements to signals and use those signals to control the bot.

This document provides detailed information about what the system will do, the interfaces provided and constraints which will regulate the usage of system.

1.1 Definitions, Acronyms and Abbreviations

- **AVR Studios :** IDE for C programming for the Firebird.
- **Bot/Robot :** The Firebird V bot which is controlled using the controller.
- **Esterel :** A programming language for the development of complex reactive systems.
- **IDE :** Integrated Development Environment
- **Kinect :** The motion sensing input device by Microsoft.
- **Operator :** The human being whose gestures are captured by Kinect and used to control the bot.
- **SDK :** Software Development Kit
- **Software Requirements Specification :** A document describing all the functions and constraints of a proposed system.
- **USB :** Universal Serial Bus.
- **Visual Studios :** IDE for using SDK used.

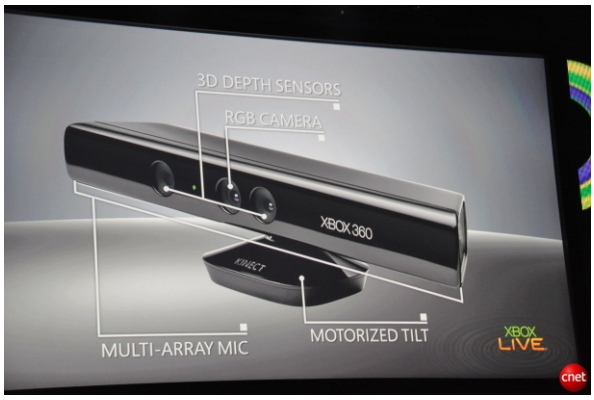
1.2 References

- NeX robotics
  `http://www.nex-robotics.com/`
- WinAVR-user-manual
  `http://winavr.sourceforge.net/WinAVR-user-manual.html`
- IAR Systems
  `http://www.iar.com/`
- AVR on Ubuntu
  `http://jumptuck.wordpress.com/2008/11/10/programming-avr-on-ubuntu-810-intrepid-ibex/`

- FireBird V Software Manual
- FireBird V Hardware Manual
- E-Yantra
  `http://e-yantra.org/home`
- OpenKinect Project
  `http://openkinect.org/wiki/Main_Page`

2 Overall Description

2.1 System Environment



The bot can move in any free-form arena. The motion of the bot is not affected by the arena layout or specifications but by the signals generated by kinect from the gestures of the operator. The gestures are converted into esterel signals which will control the movements of bot in area.

2.2 Product Perspective

- This product is useful to control the firebird conveniently without the operator having to use any sort of remote or hardware which he must learn how to operate. Using his hand gestures for controlling is very much intuitive and hence easy to train. It is a hassle-free approach to control the bot motion.
- Using Esterel to interpret the output of the Kinect device and translate it into predefined Esterel signals, we are paving the way for future projects to use the kinect-esterel interface to control various other aspects of the bot. Hence we are establishing an **abstraction** which will aid any **future** development which involves the use of a kinect via esterel.
- The bot we are developing will be a human gesture controlled vehicle. It will take its inputs from human gestures and translate them to commands. The product can be improved upon and once perfected, it can be used in various real life scenarios. The concept developed can

be incorporated into other bots such as the robotic arm which will then be much more easier to control as the operator has to perform actual arm movements instead of pulling levers to operate machines such as crane.

2.3 Product Functions

This section outlines the functionalities of the system.

2.3.1 Input via Kinect

The operator will produce some hand-gestures in front of kinect. The kinect will produce some signals according to those gestures. Those signals will be sent to system.

2.3.2 Output to Firebird

The kinect generated signals will be processed in esterel language by system. The system will generate signals in different forms. Those signals will be understandable by bot. Bot will make its movements according to those signals. The final output will be movements of bot in area.

2.4 User Characteristic

- The User is expected to be able to perform the gestures fairly consistently and with a good level of resemblance so that it can be detected properly by the Pattern Recognizer and also he is expected to have knowledge of how the different gestures are mapped to firebird.
- The operator is not required to have any kind of knowledge of any higher level programming language.

2.5 Constraints

- The operator must be within the capture range of kinect.
- There should be only one person in front of the kinect so that it identifies which person's gestures to use.
- The hand gestures should be well differentiated so as the bot does not confuse one gesture with another.
- The hand gestures should be defined so that it is possible to follow one with another so that random movements aren't attempted to be interpreted.

2.6 Assumptions and Dependencies

- Only one operator will control the bot at any time. Any attempt by more than one person to control the bot will result in the bot not moving at all.
- The hand gestures shall be distinct so that the bot will not confuse them with others.
- The signals can be sent to the bot at some limited rate, because of which control may be limited to the number of hand gestures that can be identified in a span of time.

2.7 Requirement Subsets

- Gesture Recognition
- Transmission of commands
- Interpretation of commands
- Signal generation based on commands
- Action performed based on the signals
- Response based on the environment of the Bot

3 Details

3.1 Functionality

This is a system to control bot using kinect. This system will be designed to maximize the users productivity by providing an interface to control the bot in a most natural way possible. By removing any hand holding device to control the robot the users ability to control the bot is improved as the interface is now hands free, more analog. The usability to the system is improved without compromising the ease of understanding the same. The software will facilitate recognition of the movement patterns of the user to be detected as gestures which in turn are converted to signals used to control the Robot. A set of preformatted responses from the bot can be used to detect any problems faced by the bot to perform the action or provide information on the environment.

3.2 Supportability

3.2.1 Basis for future work

This will form the basis of more complex system controlling using Kinect in the coming years.

3.2.2 Modules of the Project

- Kinect programming using Microsoft API for gesture recognition
- Firebird programming using AVR Studios IDE
- Esterel programming to connect the above 2 modules and make them interacting with each other.

Any of these module can be used/modified to be used in other improved projects.

3.2.3 Documentation

Documentation in terms of embedded comments and other explicit write-ups will help others in future.

3.3 Interfaces

3.3.1 Hardware Interfaces

- Firebird vehicle bot
- Microsoft Kinect Device
- Computer System to make interaction between Kinect and Firebird

3.3.2 Software Interfaces

- Kinect SDK
- Visual Studios
- AVR Studios
- FileZilla
- Putty

4 Quality Control

4.1 Recognizing Various Hand Gestures

Specific gesture will correspond to specific command. So all instances of that type of gesture should manifested as taht command.

4.2 Predictability

If multiple gestures come simultaneously, the bot should behave in predictable way.

5 Risk Management

- Gesture Recognition : A single pattern made by the operator will never be same as everytime even if the same person operates. So the recognition program will have to be sufficiently lenient so as to capture the various instances of same pattern by operator but not to misinterpret one gesture with another. So in order to confirm gesture, a bit of training is needed. Fall Back Plan: The gestures are pre-programmed in the code.
- Gesture to Signal Transition: The body movements of the people are made in the real world, so they will be in three dimensional and will be continuous. We need to analyze gestures constantly. We have to ensure that the bot is never stationary because of absence of signal from the user. Fall Back Plan: A signal to perform a previous gesture is emitted till a new gesture is recognized.
- Communication between System and Bot: The communication link should be fast enough to handle load and if possible then wireless to avoid hassles of wire in bot movement. Fall Back Plan: Wireless communication will be implemented only if possible. The serial link used as last resort.