SPHERE SYSTEM

Begumhan Turgut, Nevin Aydin and Damla Turgut

Department of Computer Science and Engineering, The University of Texas at Arlington, P.O. Box 19015, Arlington, TX 76019-0015, USA, {bturgut, aydin, turgut}@cse.uta.edu

ABSTRACT

Current computer simulations focus on the visual and auditory senses; yet fail to deal with physical sensations such as motion and orientation. The simulation environment must be able to incorporate these sensations in order to model more closely to the real world. SPHERE System brings a new level of realism through the incorporation of physical sensations to virtual reality applications. This provides a more interactive learning environment for the educational centers, a more interactive gaming environment for the entertainment industry and an alternative simulator basis for military training.

1. INTRODUCTION

SPHERE System [1, 2, 3] is a multiple axis physically interactive rotational chair that adds a far greater sense of realism to any supporting application. Once an application includes support for SPHERE through an interface library, a much more immerse quality to the simulation can be achieved. Anything from games [4] to environmental simulations will finally have the physical sensations of motion they have been so desperately lacking. SPHERE's orientation is computer controlled so that the application can specify to what degree and in what way the user's input is utilized. SPHERE represents an entertainment, erector set chair structure, which includes two axes of rotation with a minimum of 60 degrees for both axes. PC interface to control speed and movement of chair using pneumatics. The entire system supports and moves a person weighing at most 250 lbs and operates within a certain temperature range to ensure the functionalities of the system. For safety assurance, SPHERE contains safety kill switch for user to stop the system entirely in an emergency.

The applications of SPHERE can be attributed to a wide variety of industries [5, 7, 8]. Many organizations involved with orientational simulation of physical environments such as military trainers, the gaming industry, and educational centers will find SPHERE to be an invaluable tool. Furthermore, as virtual reality becoming more pervasive in society, production of SPHERE on a greater scale for domestic consumption becomes necessary.

The organization of the paper is as follows. Previous work is given in Section 2. Section 3 presents the SPHERE system overview. Sections 4, 5 and 6 discuss software, hardware and mechanical component of the system respectively. Section 7 concludes the paper.

2. PREVIOUS WORK

As the application presents a good opportunity for the usage of software and hardware knowledge as well as mechanical engineering fundamentals, this makes it more appealing to group of project teams and researchers to study the system for finer design and implementation improvements such as achieving better sense of realism for commercial purposes. The related work done previously along these lines are presented in this section. This establishes a good learning experience in terms of finding out what was implemented successfully that worked and what did not work which presented itself as pitfall. One thing that all various previous works have in common is that they all focused on improving particular aspects of the system. Most of the systems had the basic functionality within a certain margin of error.

The structure that the user is sitting in, which also contains the visual screen, is an essential part of the mechanical aspect; it needs to be able to support the weight and the movements during the running of a game. Keeping in mind that this was designed for future commercial entertainment set, it should also be cost effective in terms of material, size and the outlook should be reasonably presentable. One of the previous projects used water pipes as the main structural material, which presented a major problem while making adjustments to the design it is quite difficult to apply any kind of modifications to steel-built pipes.

The same group also used huge motors that were powerful enough for the movements but the size of these motors presented itself as a problem since the strength of the structure had to be increased, as a result in high cost and low quality of visual outlook The placement of the motors were also challenging part since the movements are calculated according to not only the weight of the structure but the distribution of it though the design. Another part was on the breaks. Normally, while the chair is moving if user wants to stop for any reason then the break system should bring the chair to a gradual stop. In this case, the stopping was being done immediately after the command was passed, and it had certain margin of error, which was shown in the test cases.

In another attempt, the software difficulties seemed to stand out more. There were certain parts in the modification of Descent™ code to fit it with the overall system were causing difficulties. There were also errors caused by the communication routings/protocols between the pc and the mcu. Another problem was that joystick or keyboard feedback via video signal and motion for the movement was not precise; it had a reasonably big gap meaning when the user wanted to move for instance 40 degrees, the structure was not moving anywhere close to this degree.

Overall, each of the previous works accomplished a particular goal while having certain difficulties or not being able to meet the requirements on other parts. The experience passed from a previous project to a current one still holds to be invaluable. The Sphere system provides a more complete solution to the problems encountered by the earlier work.

3. SPHERE System Overview

SPHERE is a simulation environment, which has one primary goal: to make the environment as real as possible for the user of a simulation system. The senses of motion and orientation are rarely experienced in the computer simulation realm. SHPERE will now incorporate these sensations through the use of a motor driven chair-like structure, which will "mimic" the current state of a vehicle-type simulation. The simulation itself should represent a three dimensional world in which the chair structure will position itself with a minimum of 2 axes of motion. The chair structure itself will house the user and user interface. The user will be able to control the simulation state via the user interface. In turn the simulation will interpret the user's commands and then adjust the motion, speed of rotation and orientation of the chair to match that of the current state of the simulation. This will require that the simulation know the exact position of the chair structure at all times requiring that pneumatic speed and rotation are precisely controlled by the simulation [3]. Fig. 1 shows the integration of all the components of the SPHERE system.

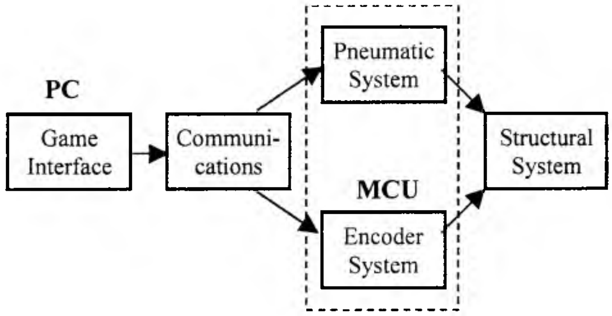


Figure 1. Sphere System Overview

The following sections describe the three components, namely software, hardware and mechanical component of the SPHERE system in detail.

4. Software Component

The software system of SPHERE is developed using C [9] and Motorola MC68332 [10] Assembly and operated under Windows NT, and Windows 95/98. To reduce the cost for maintaining the system, software modules are carefully designed and tested.

The SPHERE' s software system requires an effective way for user to communicate with the system by sending commands to the system via the joystick interface and receiving feedback via video signals and motion. The SPHERE' s software interfaces with the joystick, the user, and the controller board to get input into the system. It interacts with the chair structure to produce the movement desired and limits the movement in a certain range. It processes all data in real time with a reliable communication and keeps track of the position of the sphere [5]. To ease the use of the system, SPHERE's software allows automatically uploading the Micro Controller Unit (MCU) program and initiating the MCU and PC driver environments. Furthermore, the software system produces rotation with an accurate enough speed, and it makes the process of stopping the sphere gradual and precise. Software modules functionalities and definitions details will be discussed in the sections below in the order of implementation.

The software system requires that there be a way for the user to effectively communicate with the system, sending commands to the system via a keyboard/joystick interface and receiving feedback via video signals and motion.The following areas have been addressed:

- **GUI-Graphical User Interface**

1. An intermediate step was inserted in the integration of the Descent™ code with the SPHERE system [4]. A lightweight fully functional simulation was designed to be used for demo and testing purposes before actually implementing the Descent™ source code into the system.
2. A simple graphical interface to troubleshoot the SPHERE system. This graphical interface consists of a simple horizon displayed on a monitor and responds to movements made by a user either on a keyboard or joystick interface.

- **Communications**

1. A loader mechanism has been added to the specification that will allow the SPHERE environment on the PC to connect to and upload the hardware driver to the MCU on its own.
2. A well-defined communications/protocol package has been developed to allow the PC to communicate with the MCU [8, 9].

- **Input Devices**

Keyboard and joystick input devices have been successfully integrated into the simple graphical interface [7].

- **Descent ™**

It has been fully integrated into the SPHERE system, replacing the lightweight test interface that was used in the previous phase and includes implementation of keyboard and joystick input devices.

- **The MCU Loader Module**

This will allow the simulation to connect to the MCU, upload the MCU program, initiate the MCU driver program and establish a connection between the PC side environment and the MCU driver environment.

5. Hardware Component

The hardware system of SPHERE includes the Motorola MC68332 MCU [6, 10], pneumatics, encoders, limit switches, optoisolators, PC and other electronics devices. The hardware system requires an MCU driver program to interpret command sent from a PC and then moves the external SPHERE structure to the desired location. Externally, the controller board provides the digital interface.between pneumatics and encoders to the PC. The pneumatic cylinders [11] perform the required motion and rotation needed with braking and direction changing. On the other hand, the encoder detects the position of the chair structure and provides accurate feedback to the system.

After the entire hardware system is completed, it is packaged into wooden boxes in a stylish and protective ways. Scattered wires are neatly packaged into cable with connectors on either side for easy connection and disconnection. Wooden boxes are safely located by the SPHERE structure to avoid damage to the system or injury to users. Overall, each hardware task is designed and tested for correctness before the next one is started to enable quality control and well managed design process [3]. Hardware modules specifications and the troubleshooting details will be discussed in the sections below in the order of implementation.

The hardware system requires that an MCU based driver program will be able to interpret commands sent from a PC based user interface program, translate those commands into signals that can generate movement in an external structure, and then provide feedback to the PC based user interface program as to the position of the external structure. The following areas have been addressed:

- **Motor Requirements**

Initially, the SPHERE system was designed to use stepper motors to drive the overall structure of the system. Due to lack of financial resources and limitations on the size of stepper motors that would be needed, team SPHERE moved toward a more sensible solution. In place of stepper motors, we have implemented the use of pneumatic cylinders for movement and control. These pneumatic cylinders allow us to control the movement of the SPHERE structure more efficiently and with less energy that would have been necessary with large stepper motors. Coding for the pneumatic produced fewer overheads for the SPHERE system because of the minimal resource usage of the MCU. The Stepper Motor routines were replaced with Pulse Width Modulation routines, which offered a more general-purpose solution to driving the SPHERE structure. Special routines were also implemented to control the braking and direction changes required by the system. Although knowledge of pneumatics, air valves and relays had to be acquired, the benefits of this decision greatly outweigh the costs.

- **Motion Constraints**

Originally, team SPHERE specified that there would be a 90-degree minimum and 360-degree maximum range of motion on the SPHERE system. Due to the change from stepper motors to pneumatics, this is not physically possible. The range of motion for the SPHERE system has been redefined to meet a 30-degree minimum and 90-degree maximum total range of motion. This range of motion is fully adjustable and is controlled by strategic positioning of the pneumatic cylinders [1].

- **MCU Driver**

The MCU driver for the SPHERE system is fully functional communicating with the PC side test simulation to notify the simulation of.the current system state. The simulation is in turn is receiving input from the user and passing those inputs back to the MCU which then drives the SPHERE structure to the desired location.

- **Controller Board**

The controller board is fully functional and is responsible for two functions:

1. Isolate the MCU and primarily the solenoid valves used to drive the pneumatic cylinders.
2. Produce appropriate signals to the air valves that will in turn drive the pneumatics. This is accomplished by a unique configuration of driver chips and relays that drive 8 solenoid valves [1].

- **Movement (Pneumatic Cylinders)**

The pneumatic cylinders provide the required motion that the SPHERE system needs to be functional. There are two cylinders per axis providing approx. 300 psi of force. Braking and direction changing using this pneumatic/valve system also meet the original specifications [1].

- **Sensing Position (Encoders)**

The encoder system meets the required specification. The MCU/TPU function FQD has been implemented in the SPHERE system to interpret the quadrature phase signals that the encoders produce when motion is detected. The encoder module detects and interprets movement well within the 10-degree margin of error that was defined in the SPHERE system's original specification [1].

- **Hardware Packaging**

The entire hardware system is complete. The final system contains all hardware components packaged in a stylish and protective casing.

- **Wiring**

Wiring, solenoid valves, and air tubing are also neatly packaged and fastened to the SPHERE structure to avoid damage to the system or injury to users. Fig. 2 shows the SPHERE System Assembly Overview.

6. Mechanical Structure System

The mechanical component is responsible for the development of the sphere structure. The design for the mechanical structure is carefully planned and guaranteed the safety for the user by passing the Safety Tests. The structural system requires that a lightweight structure be built that can hold a 250lb person, respond to forces applied to it by the hardware system, and maintain structural integrity constrains and the safety of the user at all times. The structure must rotate on two axes with a total range of 30 degrees minimum and 90 degrees maximum. The following areas have been addressed:

- **Structural Design**

The spherical design of the SPHERE system has been redefined to a rectangular configuration. This is primarily due to the fact that the financial, space, and material resources that would be necessary to provide.a large spherical structure were too costly.

Considerable financial resources were invested to produce the rectangular structure, which has proven to provide added structural integrity, take up less space, weigh less, and is more easily extendible, all of which could not be guaranteed in a spherical design. This structure design meets the 250lb weight limit specified in the initial proposal. The structure meets all rotational requirements specified on both axes.

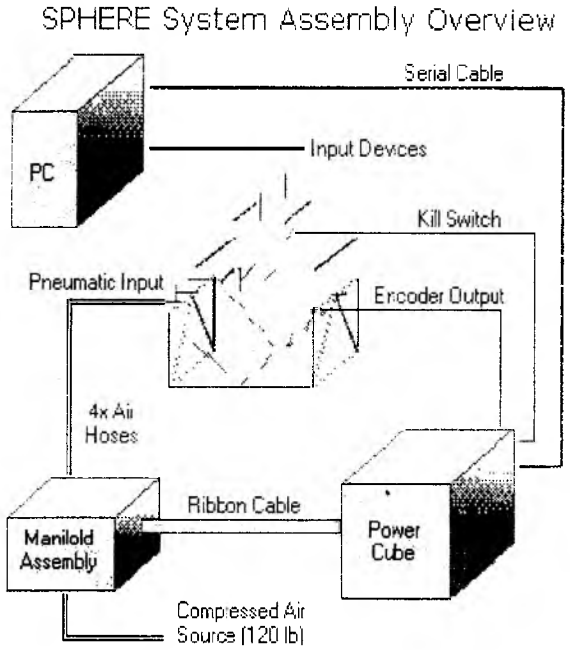


Figure 2. System Wiring and Pneumatic Hose Connection Overview

- **Rotational Requirements**

Although the structure itself can rotate a full 360 degrees on both axes, this specification has been reduced to a total of 30 degrees minimum and 90 maximum on both axes to match the specifications required by the pneumatic system [2].

- **Pneumatic Mountings**

All four pneumatic have been mounted for the movement signals that are produced by the MCU.

- **Encoder Mountings**

The encoders are mounted directly to the pneumatic cylinders providing instant feedback on the position of the pneumatic. This is accomplished through a system of pulley and elastic cord devices that interpret the cylinder's linear movement into the encoders radial movement. All.stresses and harmful forces have been removed from the encoder's using this method of mounting.

- **Axis/Bearing Mountings**

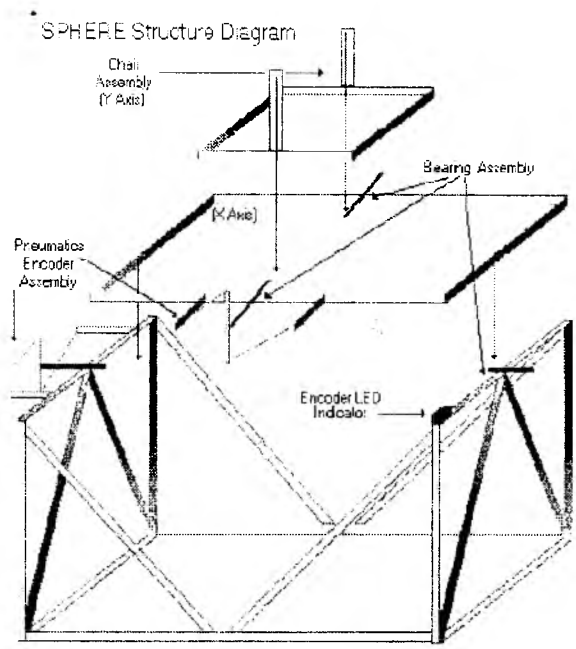
The axis/bearing mountings provide stable and strong support, which meets the 250lb weight limit specified in the initial proposal. The SPHERE structural materials used are shown in tabl. 1:

Table 1. SPHERE Structural Materials

Part Name/Number	QUANTITY	Purchase Place
1010	1"x1" T-Slotted (84 ft)	80/20 Catalog
1020	1"x2" T-Slotted (28 ft)	80/20 Catalog
4346	Joint 20	80/20 Catalog
4151	Joint 20	80/20 Catalog
4119	Joint 8	80/20 Catalog
4112	Joint 8	80/20 Catalog
Screws	Box	Hardware Store
Bearings	3	Hardware Store
Metal Rod and Plate	2	Hardware Store
Boat Chair	1	Senior Design Lab

Fig. 3 shows the Sphere structure assembly diagram.

Figure 3. Sphere Structure Assembly Diagram



6.1. Encoder Specification

Each encoder is a TTL device and is made by Clarostat. There are four pins that are available on each encoder. The four pins are labeled as: Pin 1 is +5V power, Pin 2 is quadrature data pin A, Pin 3 is ground, Pin 4 is quadrature data pin B. Each channel has an output rate of a 128 pulses/revolution. Each encoder has a 10 million revolution minimum life expectancy. Fig. 4 shows the pneumatic/encoder assembly of the SPHERE structure [1].

6.2. Pneumatic Specifications [1]:

Pneumatic cylinders: bi-directional movement with 6" of travel

- 1" diameter
- 150 psi max pressure
- pneumatic manifold: 4 - 24V control of solenoids
- 1 - input port (compressor pressure 120 psi)
- 4 - output ports (pressure/exhaust)
- 150 psi max pressure
- 30 psi min pressure
- solenoid valves 24V on/off control bi-directional
- 150 psi max pressure

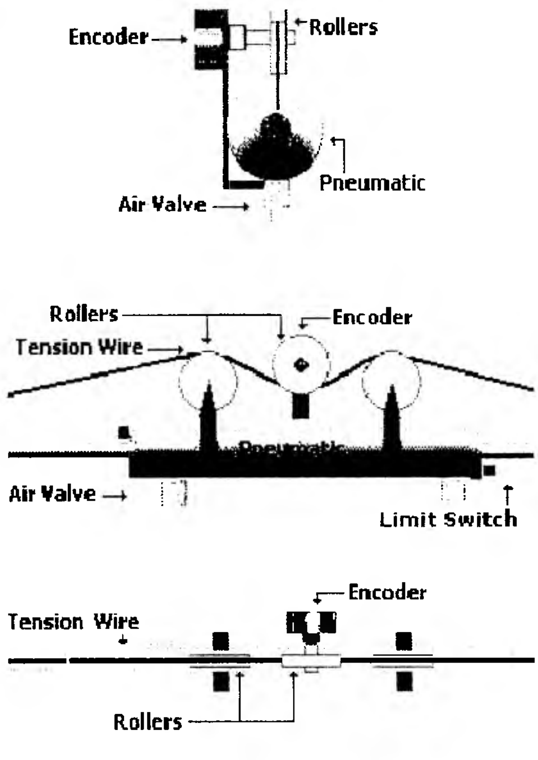


Figure 4. Pneumatic/Encoder Assembly

7. Conclusions

The SPHERE system has been developed using a bottom-up approach to produce a simulation system that focuses on achieving a sense of motion and realism,.which is missing from most of the current systems. By examining the similar previous works at the initial step for understanding what was done in the past that was either successful or not in terms of meeting the requirements, we implemented a system that combines the integration of not only software and hardware modules but also mechanical structure module as well. The main goal was to develop a system that gives more realistic feeling during the physical movement comparative to the existing systems that was accomplished through the gradual step-by-step design and implementation.

References

[1]. SPHERE Team, *SPHERE Technical Manual*. CSE 4317 Senior Design Project Deliverable.

[2]. SPHERE Team, *SPHERE User Manual*, CSE 4317 Senior Design Project Deliverable.

[3]. SPHERE Team, *SPHERE Final System Requirements Specification*, CSE 4317 Senior Design Project Deliverable.

[4]. Game Programming Techniques - SAMS Publishing

[5]. Hualon Microelectronics Corporation. http://www.hmc.com.tw

[6]. Motorola Corporation, http://www.motorola.com

[7]. Texas Instruments, http://www.ti.com

[8]. Open GL Prog.Resource, http://www.sgi.com/Technology/openGL/

[9]. H.M. Deitel and P.J. Deitel *C++ How to Program*, Prentice Hall, 1994.

[10]. T. Harman. The Motorola MC68332 Microcontroller: Product Design, Assembly Language Programming and Interfacing, 1st. Ed., Prentice Hall, 1991, http://www.prenticehall.ca/allbooks/ptr_0136031277.html

[11]. W. Bolton, *Pneumatic and Hydraulic Systems*, 1997, http://www.maintenanceresources.com/Bookstore/FluidPower/Pneumatic.htm.