International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-2 Issue-15 June-2014

Design and Modeling of a 3 DOF Machine

Marjan Sikandar

Abstract

This paper proposes an IT solution to implement a 3DOF machine by using several design patterns. A Degree of Freedom machine is a motion platform synchronized with the simulation game visually displayed. Through this research we have tried to find an optimal way to implement such a robotic machine that is supposed to be used in different virtual prototypes by industrialists. Since the development cost of an actual product often, is too high to test the initial design, many industrialists now-a-days, invest in building a prototype before actual product. Also to train the employees/students, the prototypes are very useful. For this purpose, different algorithms were studied and a design based on several patterns is proposed as a solution to implement such a machine which will benefit organizations in many ways. To prove our point we have implemented the design and hence we found out that our proposed solution which has been explained in detail in this paper is an optimal design to put into practice for any DOF machine. Thus, the given solution if approved and applied correctly can provide great benefits to the industrialists because they can save a lot amount of their time and money by using a object oriented design solution for their prototypes. The proposed design provides flexibility to the organization to develop any scenario based simulation game with any DOF machine available.

Keywords

3 DOF, Simulation, Virtual Reality, Inverse Kinematics

1. Introduction

Driving simulators are efficient tools for novice as well as experienced riders to learn and test the vehicle in a safe environment. It provides a chance for the rider, to feel the ambiance and riding sensation, with no limitation of scenarios available. [1] With a simulator in place, the organization is benefitted in terms of fuel; accident cost and has a leverage to test the vehicle in any environment [2]. Some multi-national organizations like Honda let their researchers/testers experience the vehicle design by driving simulators. Many motor-cycle and car simulators are in place, since 1988, for the trial of vehicles [1]. Most of the auto-mobile companies like Ford, General Motors and Chrysler use simulators not only to experiment the technical characteristics but also for the functional simulation of their products. Daimler – Chrysler estimates that measurement costs for each new suspension study are reduced roughly by $1500 - $2000 along with the simultaneous testing time reduction [3]. Ford claims to accomplish reduction in the development time of mechanism, by the use of virtual environments to test the product design beforehand [2].

Major portion of the architecture of such a driving simulator is based on three to six degrees of freedom platform. The motion is translated into three translational displacement components (heave, sway and surge) and three rotational components (roll, pitch and yaw) [10]. For lesser degrees of freedom, few displacement components are omitted.

This article provides an overview of a 3 DOF machine, considering the scenarios for riding simulators. The system architecture and motion kinematics of a 3 DOF platform is discussed in great detail. We have tried to discuss some object oriented design patterns to propose a flexible solution for the machine. As the importance of a driving simulator is noteworthy in today's world, where the fuel prices and cost to make an actual vehicle is quite high, one can estimate the growth in use of simulators in near future. The proposed design provides flexibility to the developer/designer/analyst to implement a DOF machine for any simulating game according to the essentials of the vehicle. The objective of the article is to familiarize developer with the platform dynamics and its implementation for any simulator in consideration.

2. Preliminaries

The initial requirements for a riding simulator are; three degrees of freedom motion platform and a simulation game [10]. For the hardware part of it, one

Manuscript received June 12, 2014.
Marjan Sikandar, School of Information Science and Engineering, Central South University, Changsha, Hunan Yuelu District, China.

needs to understand the dynamics (kinematics) of motion platform, to exploit it in its essence. For the software part of it (simulation game), one needs to deal with the graphics and sound equally to cheat the imperfections of human sensory and perceptual systems [11].

Motion Platform

A motion platform or a motion simulator is a kind of machine, which encapsulates the feeling of being in a moving vehicle [4]. It consists of two rigid bodies (base and platform) connected to each other by a number of open or closed chains (legs) [12]. With consideration to three degrees of freedom, the platform contains three legs, capable of moving in linear motion (heave) and along rotational axis (roll and pitch movements) [10].

A. Working

A motion platform can also be called as a motion simulator or a motion seat [13]. The movement of the seat is synchronous with simulation game or scenario visually displayed. It is designed to add a sense of touch, element to video gaming, simulation, and virtual reality. When motion of the seat is synchronized with the audio and video signals of the simulation game, the result is a combination of sight, sound, and touch [14]. Mostly full motion simulators (three to eight degrees of freedom platforms) move the intact driver compartment and result in not only change in orientation but also produces the effect of false gravitational forces [4]. This activity tricks the mind and driver/human is of perception that he/she is immersed in the simulated environment and experiencing kinematic changes in position, velocity, and acceleration [4]. Sometimes the driver cannot accept the mentioned alterations of movements that could lead to motion sickness [13].

Figure 1 depicts the motion seat and attached devices (monitors, gears, and steering) of a passenger car simulator. Triple Wide Screen is a UK based organization, experts at making sophisticated driving simulators [15]. As stated by Alric Kitson, representative of Toyota MR2 Rogue Racing team, that the driving simulator is more natural and allows you to react in the same way as in reality [15].

Figure 1: Prototype of a Car Simulator [15]

The visual display is divided into three screens/parts for the driver/user to get the front view and the rear views as in real life scenarios [16]. The figure below is taken from a simulated environment created for a train platform.

Figure 2: Visual Display of a Train Simulator [16]

B. Linear and Rotational Movements

Humans perceive movements of constant velocity as moving with zero speed, for example, a car moving with constant velocity is same as sitting in a chair. So, the only recognizable movements for humans are those either of acceleration or a jerk [5][11]. For both the mentioned scenarios, platform moves in linear axis. For each change in motion, the platform moves upward /forward /backward according to the simulation game and then comes back to its original (home) position with a constant velocity. Since, driver/user is so immersed in the game; he/she never gets to know that the platform is moving towards home position, after each movement realization.

Jerks are sometimes, realized by both the linear and rotational axis movements thus the platform has to achieve a certain angle, determined by the game [17]. The desired angle is subtracted from the current angle of the platform and then the delta angle is to be realized by the three legs simultaneously, thus the platform moves accordingly.

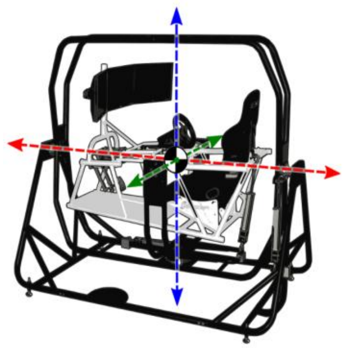

Figure 3: Roll, Pitch and Yaw Movements of a 3DOF [18]

The figure above shows the possible directions for the platform to move [18]. The platform contains the driver compartment within it. The advantage of 3DOF over any other is that there is no mechanical limit to achieve any kind of motion, because angle derived to achieve a motion cannot force the legs to achieve an improper position. Legs will return back to home position, in any case where the delta angle is greater than the limit of legs length [19].

Simulation Game

A simulation game is a combination of sound and graphical effects. Usually the organizations compile several scenarios to judge their products and train the drivers accordingly. For each game, the screen is split into three views (front and two rear views). Furthermore, the audio helps a user to be immersed in the scenario [14]. In most of the cases, the audio and video encapsulates the imperfections of the platform motion [11]. To get the sensation of real environment, the sound, display and the motion devices has to work simultaneously. Thus state-of-art technology is used for such simulators. In many professional schools, initial trainings are conducted partially in the real environment, while rest is conducted through low-cost training devices. As the student becomes familiar with the basic handling and skills of the device, the training is then shifted to actual devices [11]. For this reason, the simulation games are designed for each trainer [11], customized to the requirements of the school. The scenarios and exercises provided by such games are sometimes impractical for the actual devices to perform [11]. For example, for flight simulators the crash situations are nearly impossible to be imitated for the students.

3. Simulator Software Design

The proposed design is explained with the help of system architecture, class diagrams and implementation codes of forward kinematics algorithm.

System Architecture

3 DOF is connected to a Linux based system, where a kernel module generates pulses for each motor attached to the leg. The simulation game is running on PCs attached to the monitors. Simulation game is responsible for ordering the program to move the 3DOF machine. Thus, to successfully run a 3DOF machine, a kernel based program and a user application to calculate pulses for each movement is to be created. The figure below shows system overview in terms of the 3DOF states. The machine is in any one state, at one moment and time.

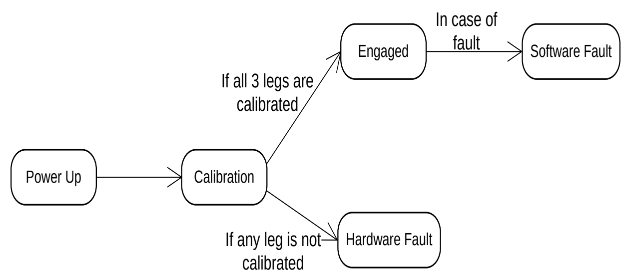

Figure 4: System Architecture

Different states that machine could be in and the transitions from one to another are discussed below.

A. Power Up

Motion platform initiates from the Power Up state. All the physical parameters and software parameters are set to default values.

B. Calibration

Calibration is a phase; where pulses are generated for a specified period of time (generally 30 seconds) and each leg tries to be in its mean position. The leg position is determined by the use of sensors. Pulse generation terminates, when sensor indicates that leg is in its mean position. The process of calibration is very important, for each leg to return back to its home from any awkward position.

C. Engaged

The machine enters engaged state, if and only if, all three legs are calibrated (i.e. all sensor pins attached to legs, are set to low). The user application does the calculations for the desired angle, maintain the state of the machine and generate the pulses for the machine. Kernel program is responsible to deliver the pulses to the machine, here in Engaged state.

D. Hardware Fault

If in any case, the motion platform's three legs are not calibrated properly (i.e. all sensor pins attached to legs are not low), the motion platform indicates a faulty state and halt the application. To resume back, both kernel and user application needs to be restarted.

E. Software Fault

From the Engaged state, if for some reason, the game is halted or a problem lies in communication between kernel module, user application and the simulation game; the machine is set to indicate a soft fault.

Class Diagram of the System

The class diagram here is divided into different parts. The first part describes the states of the system. We have use *state* design pattern [20] to realize different platform states. The figure below depicts possible states of the system and how client can access all states.

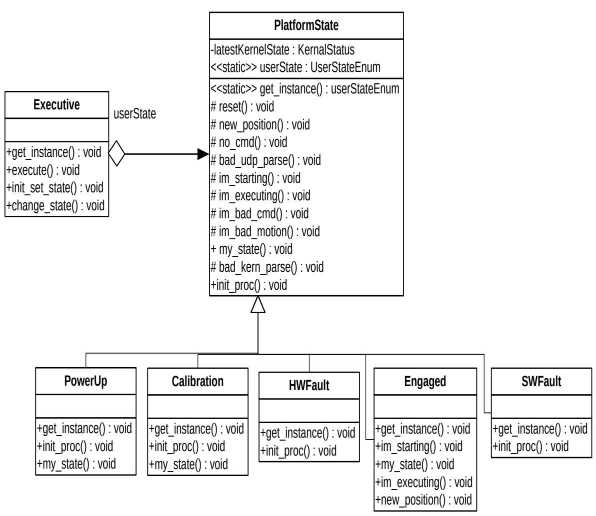

Figure 5: Platform States of Motion Platform

Platform state is the main class; from which each state is derived. Methods are declared in the parent class and each state class implements the methods in its own way. Advantage of using *state* design pattern is that states can be added or removed at any point and time. [20] Furthermore, there is only one instance of the *PlatformState* class. To ensure that, we have use *Singleton* design pattern. [20] The client can access the instance at any point in time, by method *get_instance ()*. The second part of the diagram shows how the pulses are generated for each leg. We have used *adapter* design pattern [20] to use an existing library from Stanford. The adapter (*MatrixMath* class) provides an interface for the application to use the methods of the library.

Furthermore, *MotionPlanner* is composed of *Platform* class, because *Platform* acts as structure for the main class. *Platform* contains the direction and pulses for each leg and only does one calculation at a time. Whereas, *MotionPlanner* keeps record of all the calculations and queue pulses that are to be sent to the machine. There exists no platform, if motion planner doesn't exist. Figure 6 depicts relationships between two classes.

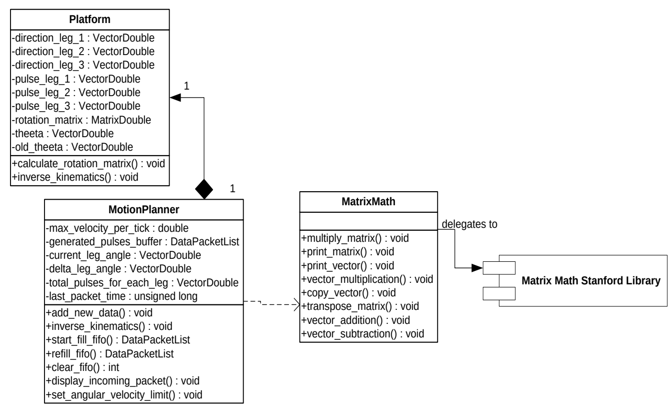

Figure 6: Motion Platform Calculations

The third part represents queuing mechanism and interactions of the kernel application with the user application.

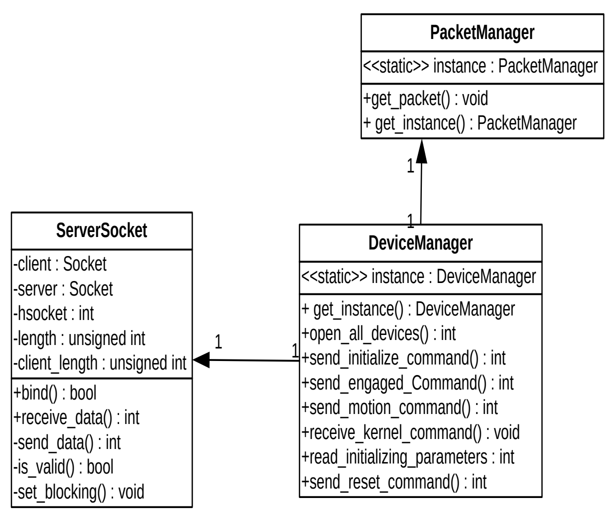

Figure 7: Interaction with Kernel Application

In the figure above, since devices and packets are the managers, thus they cannot have more than one instance. *Singleton* design pattern [20] is used to ensure that. For this purpose the instance of each class is static and there is a public member *get_instance()* to retrieve its instance at any point in time.

Motion Platform Kinematics

The desired angle determined by the simulation game has to go through several steps, before the angle is converted into pulses and direction for each leg. The generated pulses are then queued for the kernel program to execute them. The procedure of converting angles for each DOF leg is determined by the kinematics of motion platform [7].

A. Working

The procedure of motion platform kinematics is described by the diagram below. Desired angle and current position of DOF acts as an input for the inverse kinematics block. Within the kinematics block, the desired angle is multiplied with the rotation matrix, to perform the rotations in Euclidean space. [19] The result is then used for the pulse generation process. Generated pulses can be passed through filter to avoid the faulty state.

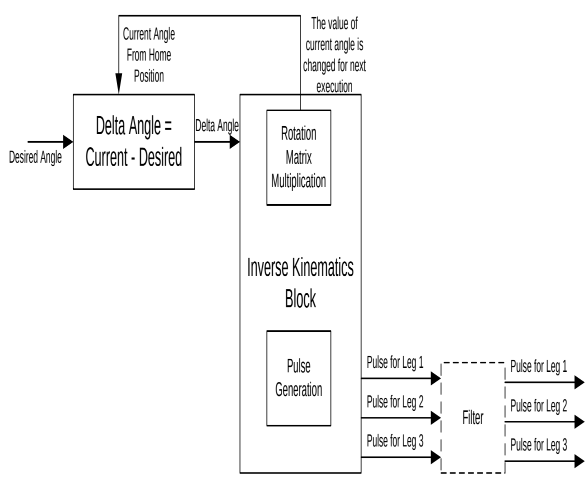

Figure 8: Procedure of Kinematics [19]

Since we are dealing with 3DOF, where roll and pitch are two movements along the rotational axis, we chose figure 9 as our rotation matrix [6][8].

$$\begin{bmatrix} \text{roll} \\ \text{pitch} \\ \text{yaw} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 9: Rotation Matrix [8][9]

B. Pseudo Code

For a 3DOF machine, inverse kinematics calculations are the most important part, thus some part of the inverse kinematics code is discussed here [19]. The major part of the code is how to deal with generating pulses and calculating the rotation matrix with the help of delta angle.

Pre Condition: Desired Angle Packet is received
1. Find Delta Angle
Delta angle = Current angle - Desired angle
2. Find Maximum Number of Pulses that can be generated Use max_velocity_per_tick
3. Rotation Matrix
Multiply rotation matrix with the max_velocity_per_tick using delta angle
4. Fill the Queue with number of pulses generated for each leg
5. Process Queue at the Kernel Application

4. Discussion

Through this research we have successfully described how a DOF machine can be implemented and the important considerations when implementing a degrees of freedom machine. To support our research we have studied motion kinematics algorithm which is a process to identify leg movements provided the angles are given [19].

The experimental design used during this research has properly addressed some known issues. The observation we have obtained from this research indicates that with the help of design patterns, we can easily solve our common day problem [20]. Not only that, we can also make the software design more flexible, easy to implement and an extensible design.

The suggested solution was implemented within controlled environment, by some fellow scientists. After implementing the solution we have found out that the 3DOF machine was working perfectly fine and there were no mishaps. Hence it proves that if the design is practically implemented in industrial applications, it might be of great use to those organizations, who are implementing it for the first time.

Proposed approach has not yet been proven to be industrially effective and various researchers have been conducting their research in different aspects of the machine. However, we are keeping high hopes and we postulate that this research which is a newer approach of implementing a 3DOF machine by using various design pattern techniques will definitely give us encouraging and positive results.

It is going to help industrialists of our country who are involved in building product prototypes since a long time by implementing various kinds of techniques. So this research will provide them a new perspective of looking to this problem and there are chances that it will really help them to implement such kind of machines.

5. Future Prospects

The future prospects of this research include performing experiments on inverse kinematics behavior of the motion platform. A part from that, different types of filters (like FIR filters) can also be applied to monitor the application's performance and to gather different results. Furthermore, the software design has to be tested by integrating it with simulation game as well.

Acknowledgment

The research was accomplished with the tremendous support of my supervisor Professor Yu Lasheng who had guided me in every aspect of the report and had given his invaluable guidance and support throughout the project. The research seems to be impossible without his assistance. Furthermore, without such love and affection of my friends, I would have never completed the job successfully.

References

[1] L. Nehaoua, S. Hima ET. al. "Design and Modelling of a New Motorcycle Riding Simulator". American Control Conference, New York, pp. 176-181, July, 2007.

[2] Nikolaos, Bilalis & Markos, Petousis & Aristomenis, Antoniadlis. "Industrial Applications Simulation Technologies in Virtual Environment". Online at http://www.m3.tuc.gr/ANTONIADIS/13 INTERNATIONAL CONFERENCES/13.22.pdf (as of July 2014).

[3] Pamela J. Emery. (May, 1999). "System Level Virtual Prototyping". Retrieved December 14, 2011 from Computer Graphics World.

[4] Scanlon, Charles H. "Effect of Motion Cues during Complex Curved Approach and Landing Tasks". NASA Technical Paper 2773. 1987. Retrieved December 20, 2011.

[5] Alvandar, Srinivasan & Nigam, M.J. "Inverse Kinematics Solution of 3DOF Planar Robot using ANFIS". Int. J. of Computers, Communications & Control, Vol. III, pp. 150-155, 2008.

[6] Rotenberg, Steve. "Inverse Kinematics". Computer Animation, UCSD, Winter 2005. Retrieved December 17, 2011 from University of California, San Diego. Available at: http://www.docin.com/p-783973766.html.

[7] Museful. "Expressing 3DOF Rotation". Retrieved December 16, 2011 from http://www.museful.net/2011/system-modelling/expressing-3-dof-rotation.

[8] Wolfram, Math World. (December, 2011). "Rotation Matrix". Retrieved December 16, 2011 from http://mathworld.wolfram.com/RotationMatrix.html.

[9] Brauer, Jurgen. (August 23, 2011). "Rotation Matrices- Human Body Representation". Retrieved December 17, 2011 from http://www.juergenwiki.de/work/wiki/doku.php?id=public:rotation_matrices.

[10] Hawley, Patricia A. & Blawwkamp, Ross A. (2010). "Six-Degree-of-Freedom Digital Simulations for Missile Guidance, Navigation and Control". John Hopkins APL Technical Digest. Vol. 29.

[11] Heyde, Markus von der. & Riecke, Bernhard E. (December 2001). "How to Cheat in Motion Simulation – Comparing the Engineering and Fun Ride Approach to Motion Cueing". Published December 2001. Online at http://www.kyb.tuebingen.mpg.de/fileadmin/user_upload/files/publications/pdfs/pdf635.pdf (as of July 2014).

[12] Zhang, Yi. "Rapid Design through Virtual and Physical Prototyping". Retrieved December 20, 2011from http://www.cs.cmu.edu/~rapidproto/mechanisms/chpt4.htm.

[13] Denne, Phillip. "Motion Platforms or Motion Seats". Published September, 2004. Online at http://www.advancedmotion.net/pdf/cyber2.pdf (as of July 2014).

[14] Denne, Phillip."Motion Systems and Visual Displays". Published January, 1994. Online at http://www.advanced**motion**.net/pdf/sim6.pdf (as of July 2014).

[15] Triple Wide Screen. (2011)“VRX iMotion Premium Turn Key”.Retrieved December12, 2011 from http://triplewidescreen.co.uk/products/.
[16] Auto Motto. (April, 2010) “15 High End Super Realistic Driving Simulators”. Retrieved December 12, 2011 from http://www.automotto.com/entry/15-high-end-super-realistic-driving-simulators/.
[17] Allerton, D. “Principles of Flight Simulation”. Wiley Press, John Wiley and Sons, Ltd., Publication. Published 2009.
[18] Creative Commons Attribution Share Alike 3.0. (August, 2009) “3DOF Motion Simulator”. Retrieved December 12, 2011 from http://wikipedia.org/wiki/File:3DOF_motion_simulator.png.
[19] Joshi, Atul Ravindra. “Design and Control of a Three Degree of Freedom Planar Parallel Robot”. Ohio University. Published August, 2003. Online at https://etd.ohiolink.edu.
[20] Larman, Craig. (October, 2004) “Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development – Third Edition”.

Table 1: Glossary Terms

Word	Meaning
3 DOF	Platform that provides three degrees of freedom namely; roll, pitch and heave
Simulation	A method for implementing a physical, mathematical or otherwise logical representation of a system, entity, phenomenon, or process
Linear Movements	Heave, surge and sway are considered as linear movements of a DOF
Rotational Movements	Roll, pitch and yaw are considered as rotational movements of a DOF
Forward Kinematics	Computes world space end effector DOFs from the joint DOFs
Inverse Kinematics	Compute the vector of joint DOFs that will cause the end effectors to reach some desired goal state
Jacobian Matrix	A vector derivative with respect to another vector
Socket	Mechanism for delivering incoming packet to destination process or thread
Rotation Matrix	Matrix to calculate pulse for given angle
Filter	Process that removes the some unwanted component or features from given signal

Marjan Sikandar was born in Karachi, Pakistan, in 1989. She received her Master’s degree in Computer Application Tech. majoring AI in 2014 from Hunan, China. Her main areas of research interest are sentiment analysis, natural language processing and software designs.