International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-1 Issue-2, July 2012

# Distributed Computing Solution for Hardware-In-Loop Simulation of Indian Satellites

**Rashmi Jagade, Sridevi. K. N, Jitendra Nath Mungara**

***Abstract*****: *The purpose of Hardware-In-Loop-Simulation (HILS) is to verify the hardware interface of On-Board- Computer (OBC) with flight sensors and actuators and to validate the closed loop performance of attitude and control elements (AOCE) for various control modes in real-time. This document presents distributed computing configuration of computing elements that interact mutually to achieve the real-time performance of hardware-in-loop simulation. The system architecture proposed is used successfully to accomplish the real time closed loop performance for HILS. By distribution of resources, we can achieve the complex computation requirements of spacecraft dynamics simulation, telemetry (TM), telecommand (TC) and star simulation with optimized delay.***

***Keywords*****: *HILS; AOCE; Real-time; Computing elements***

## I. INTRODUCTION

After the flight hardware are designed and implemented, it has to be tested extensively before it can be used with confidence. In case of complex real time embedded systems like Satellite Attitude Control System this process is hard and costly. The digital simulations on PC's are non-real time and also real hardware is not used. HILS configuration makes the hardware component behave as closely as possible to those that would be encountered in the real system [1].To have such real time test platform, an efficient architecture that consists of distributed network of systems performing complex computations is presented. This enables the applications to meet their end-to-end timing requirements.

The existing HILS setup has individual user interfaces for TM, TC, and Dynamic Front end each running on different PCs. Besides that HILS testing is carried out manually i.e. first TC is sent, accordingly TM data is viewed and cross checked. If TM data is on par with the expectations, run is carried out. The control package in closed loop with actuators and sensors will converge the initial Errors. If the errors at the end of runs converge as expected we proceed to the next run, else the same run is repeated again.

## II. SYSTEM ARCHITECTURE

In order to arrive at suitable system architecture we have to critically analyze the following issues involved, namely:

- Bring together all the computational requirements, timing requirements and electrical interface requirements.
- Identify the legacy codes and its interfaces along with its systems requirement.
- List all the user requirements in terms of the processing and display of mission parameters, with user defined processing data base system.
- Modelling of sensors and actuators, orbit modelling, 3d/6d trajectory modelling for spacecraft dynamics simulation.
- Develop a communication system that makes it easy to implement the logical system architecture on top of it, with all inter module and inter system communication as well-defined processes. This system architecture design is carefully done including all flexibility of traditional system in order to make distributed application developed to be less error prone with ease of realization.
- Study of total memory and storage requirements and configuration of various system memories.
- Study of man–machine–computer interface Interconnecting 3 axis motion simulator/servo table for space-craft motion, dynamic multi star field Simulator (dmss) along with sky background and orbit motion.

The computational requirement for HILS is comprised of Acquisition of torque signals and health signals from flight control packages solving kinematic and dynamic equations to generate attitude and rate information in real time.Driving of servo table in real time as per attitude and rate data.Processing of the acquired data for control and display of health information in engineering units.Generation of commands to all packages as per initialization, orbit sequence and termination of simulation.Star simulation on a screen to be sensed by star simulator.

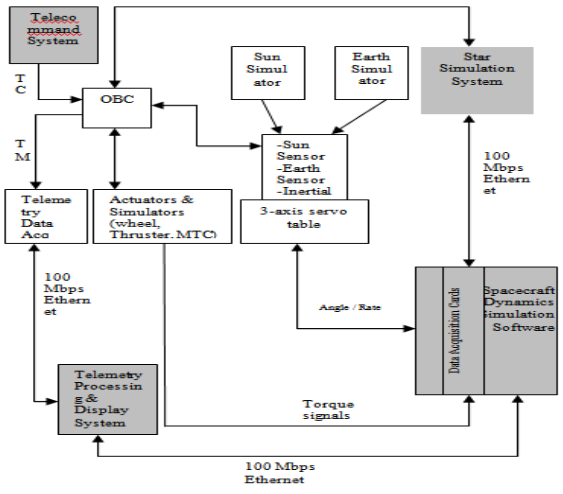

Fig 1: Distributed computing HILS Architecture

**Revised Manuscript Received on 30 July 2012**

***Correspondence Author(s)**
**Ms. Rashmi Jagade***, Department of Computer Science and Engineering, CMRIT, Bangalore, India.
**Mrs. Sridevi K.M**, Asst. Prof: Department of Computer Science and Engineering, CMRIT, Bangalore, India.
**Dr. Jitendra Nath Mungara**, Prof& Dean: Department of Computer Science and Engineering, CMRIT, Bangalore, India.

Proposed system architecture is The project here is to minimize manual operation in HILS testing, by making a common user interface for all TM, TC and Dynamic Front end. Besides runs in HILS testing are to be automated so that once a run is selected system can itself check the health parameters in TM and can start the run and can notify the user if the errors converge, so that the next run can be started.

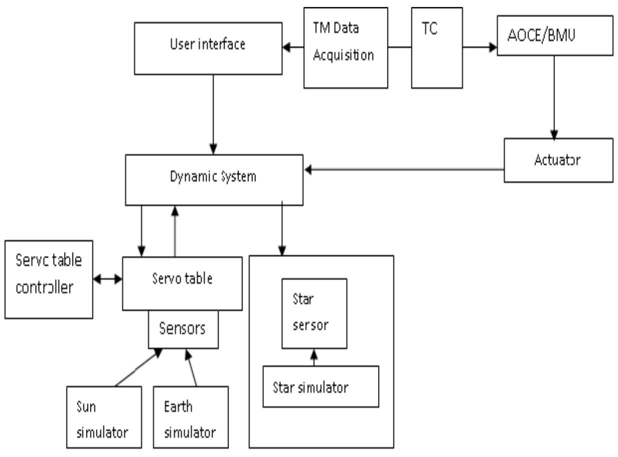

Fig 2. Proposed system of HILS

## III.SYSTEM DEIGN

The Dynamics simulation computer should be fast and efficient with real time features as it involves computation of several complex software modules as shown in Fig. 2.

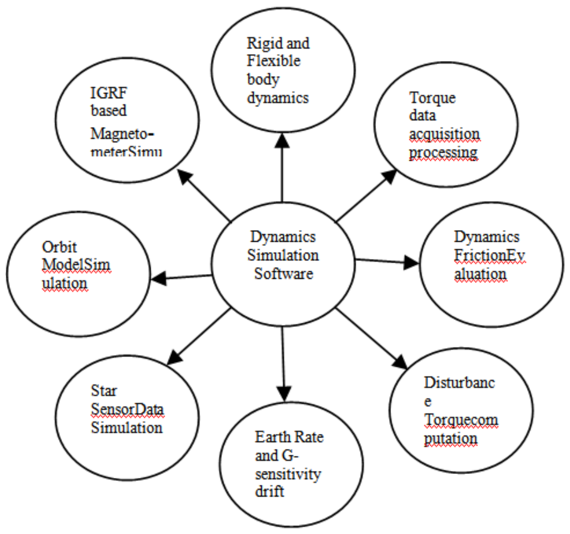
Rigid and Flexible body dynamics
IGRF based Magneto-meterSimu
Torque data acquisition processing
Orbit ModelSimulation
Dynamics Simulation Software
Dynamics FrictionEvaluation
Star SensorData Simulation
Earth Rate and G-sensitivity drift
Disturbance Torquecomputation

Fig.3. Various Software modules of Dynamic Simulation Computer

The core software module of DS computer is to findnew attitude information from acquired hardware control signals along with orbit simulation and other required simulation models i.e.

Control + Disturbance = Inertia* d (Angular velocity)

TorqueTorquedt

The spacecraft inertia along with flexibility of spacecraft elements is acted upon by control torques and disturbance torques which causes the spacecraft to experience attitude motion in the form of attitude change and body rate. As an example body rate computation and Euler rate derivation are considered.

Body rates are obtained by solving the following dynamic equation,

$$\begin{bmatrix} \dfrac{I}{K} & \dfrac{K^T}{U} \end{bmatrix} \begin{bmatrix} \dfrac{\dot{W}}{\ddot{q}} \end{bmatrix} = \begin{bmatrix} \dfrac{T - W[IW + Hw]}{-2\gamma\sigma\dot{q} - \sigma^2 q} \end{bmatrix}$$

where I = Satellite Inertia Matrix (3 x 3)

W = Body rate vector (3 x 1)

T = Total torque vector (3 x 1)

$H_w$ = Angular Momentum Vector due to Wheels (3 x 1)

K = Coupling matrix of sail and array (10 x 3)

U = Unit Matrix (10 x 10)

$\gamma$ = Dampingvector (10 x 1)

$\sigma$ = Frequency vector due to sail and array (1 x 10)

$q$ = Generalised co-ordinate vector for flexibility (10 x 1)

The body rate act on the coordinate transformation matrix to produce Euler rates and Eulerangles which are used to rotate the table gimbals.This coordinate transformation matrix is unique to the gimbals axis definition, with outer-middle-inner order of rotation. The software implementation for deriving Euler rates from body rates are explained in [4].

Euler angles are obtained from the co-ordinate transformation matrix as follows:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 1 & \dfrac{Sin\,\theta\; Sin\,\psi}{Cos\,\psi} & \dfrac{Cos\,\theta\; Sin\,\psi}{Cos\,\psi} \\ 0 & Cos\,\theta & -Sin\,\theta \\ 0 & \dfrac{Sin\,\theta}{Cos\,\psi} & \dfrac{Cos\,\theta}{Cos\,\psi} \end{bmatrix} \begin{bmatrix} Wz \\ Wx \\ Wy \end{bmatrix} \quad \text{--- (1)}$$

where,

$$\psi,\ \phi,\ \theta \rightarrow Yaw, Roll,\ Pitch\ \ Euler\ \ angles$$

$$\dot{\psi},\ \dot{\phi},\ \dot{\theta} \rightarrow Euler\ rates$$

$$Wx, Wy, Wz \rightarrow Yaw,\ Roll,\ Pitch\ body\ rates$$

Equation (1) pertains to roll-yaw-pitch rotation sequence with outer gimbal as roll, middle as yaw and inner gimbal as pitch.

In addition to new attitude determination, DS system performs other special computations such as mathematical modelling of Dynamics Friction Evaluation for wheels, Earth rate and G-Sensitive drift compensation for Gyros, Orbit modelling for position and velocity computation and magnetometer simulation as described in [1].

A star catalogue with relevant stars is stored with sl. no, direction cosines and magnitude in DS system. From mounting quaternion of star sensor, itsmisalignment, along with orbit model, attitude quaternion and Direction Cosines (DC) of catalogued stars are computed. Using DC, the corresponding pixel and line Nos are communicated to star simulator in real time. These computations are required to be executed iteratively within the time constraint of simulation step size in real-time. To achieve this, a workstation with Linux OS is chosen. This provides flexibility for application software to execute in single user mode for real-time performance. Since these mathematical computations require dedicated computer resources, no other application is made to execute. The time taken for complete iteration takes 0.6 ms without network service activities.

TM Data Acquisition (TM Data Acq) computer acquires telemetry data from BMU. This data is transferred to TM Processing computer wherein TM processing, page display, plotting etc. are performed as shown in Fig 3.The acquisition and processing software uses the concepts of shared memory and semaphore to ensure data consistency. The software for processing parameters of different processing types such as 1 bit /2 bits /8 bits /16 bits data, multiplexed data, and analog data, are defined in a database for ease of configurability. These computations are required to be executed at the rate of 1.0 second.

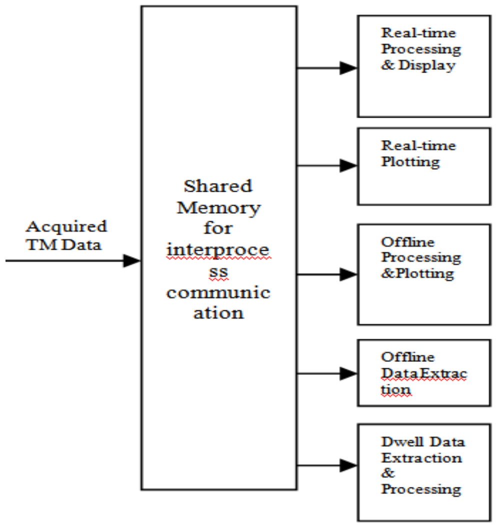
Acquired TM Data
Shared Memory for interprocess communication
Real-time Processing & Display
Real-time Plotting
Offline Processing &Plotting
Offline DataExtraction
Dwell Data Extraction & Processing

Fig.4. Various tasks performed in TM computer

The Telecommandcomputer uses add-on command generation card that accepts command inputs, modulates the data and transmits to OBC. The application software has the capability to accept different types of commands such as Pulse command, Data command and level command from user. Different processing types are specified for generation of data commands in a database. The software also allows sending group of commands in "File Mode". A database is maintained that has all possible commands that user can select for commanding BMU. Fig 4 gives the interface between OBC and Telecommand computer.

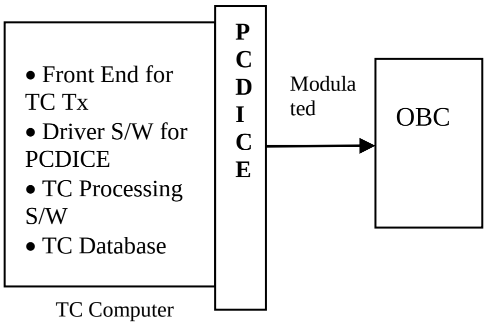

Fig.5. OBC – TC Computer Interaction

## IV.SOCKET COMMUNICATION

The communication between various computing elements is realised by means of socket communication using client/server (C/S) model. C/Sstructure allows sharing information and delivering message inevitably between many processors [3].The core of C/S system structure is distribution of task**level**application between client and server. The basis ofexchange is communication software between client and serverwhich in our case isTCP/IP.The C/S structure is as shown in Fig.5. First, the socket function call creates a typeof socket in both server and client, then server binds socket into client-side through bindfunction.Client listens attentively to theformation of connection. Servers then transfers accept function to start in the port which was bound.

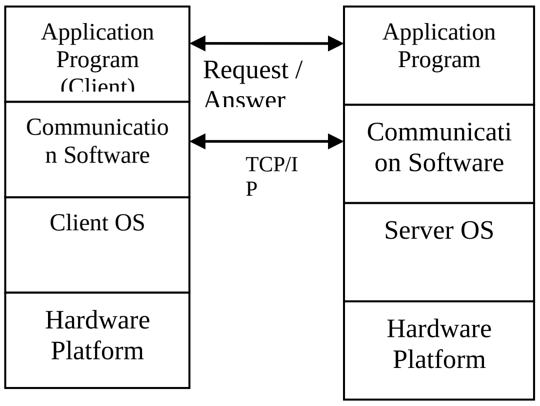

Fig.6. Client/Server system structure

TCP is chosen as it provides an unfailing connection-oriented byte stream transport layer service. TCP packs up the user's data into message segment; it activates a time controller after sending data; confirming the receiving data from the side of communication, and reordering the scrambled data, discarding the duplicative data. TCP provides flow control from end to end, calculates and verifies a complete end to end check.

## V.LOOP DELAY REDUCTION

By choosing DScomputerto be a workstation with Linux platform, we could gain advantage in terms of delay and security. Linux makes our system more secured in terms of Virus and Trojans. Development of device drivers for PCI add-on cards for Linux enabled to combine the Data Acquisition System and DS in a single computer. In the earlier configurations, a separate PC called "Front End Processor" (FEP) was used for data acquisition and Dec-alpha computer was used for DS as shown in Fig. 6.

The loop delay is compared by considering a situation of reading the thrusters PWPFM pulse of AOCE, computing the torque information and driving the servo table with the new position.

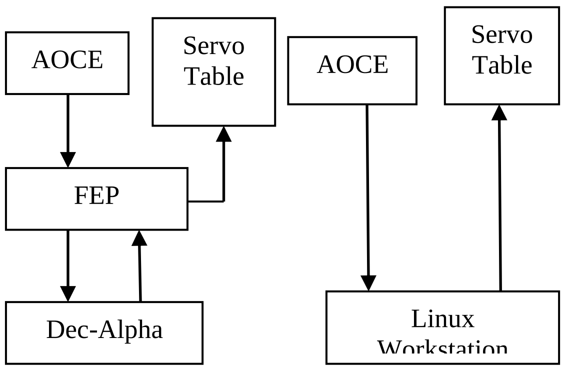

Fig. 7 Configuration of computing elements in conventional and proposed architecture

In earlier configuration, the steps involved are:

PWPFM pulseis read by FEP and the On-time of the pulse is computed.

On-time information is passed to Dec-alpha system.

In Dec-alpha system, the control torque is computed.

The attitude dynamics equation is solved to compute the new attitude information.

This new position along with rate is communicated to FEP,

FEP drives the table for the computed position.

In the present configuration with the Linux workstation, the data flow for the same situation is optimized as:

PWPFM pulse is read by Linux Workstation and On-time of pulse is computed.

Torque information is computed.

For this torque, new attitude information is computed.

Drives the table with new position and rate.

By comparing the steps, it is clear that earlier configuration involves the network communication delay two times where as such delay does not exist in the present configuration. The total closed loop delay of earlier configuration is 4 cycle time (32.0ms) where as in the present configuration it is 1 cycle time (8.0ms). The total computation time is accounted for less than 1 ms.This has resulted in getting the simulation results to match as close as digital simulation

## VI.IMPLEMENTATION

The proposed system is designed on 64 bit windows and 32 bit Linuxwith 1.84 ghz processor and 2 gb of ram. the proposed system is simulated using visual studio 2010. To implement this the project here is to minimize manual operation in HILS testing, by making a common user interface for all TM, TC and Dynamic Front end. Besides runs in HILS testing are to be automated so that once a run is selected system can itself check the health parameters in TM and can start the run and can notify the user if the errors converge, so that the next run can be started. Here is the procedure.

Selection of a table from modes and converting table into database. One table has set of N runs each run containing file command in .PCMD file and ensures all command send through TM(Manually or through a program). First run, send required commands, ensure in Telemetry through program and move to IC and monitoring to servotable to ensure IC movement. Then starts the run automatically during run servotable moves to NULL and finally end of run .

## VII. RESULTS

The main theme of this project here is toconverge the errors in the servo table. The front of this project contain of two phase one is for Telemetry and Telecommand and dynamic simulation

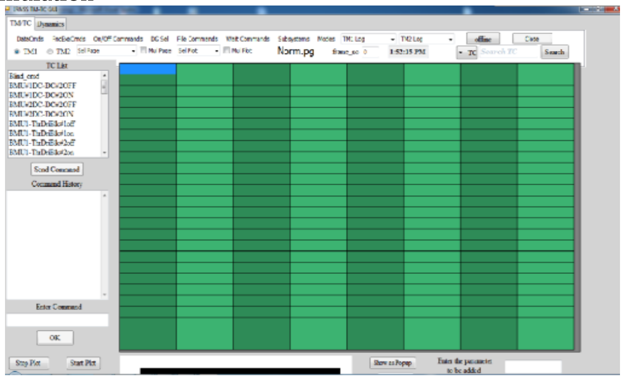

Fig 8: Front end GUI of TM/TC

The TM/TC contains the searching of a file commands of telemetry and telecomm and, waiting to send next command button, telecomm and list.

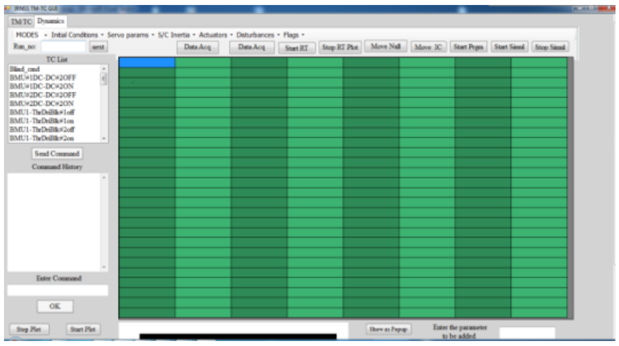

Fig 9: Front end GUI of Dynamic simulation

The dynamic simulation contains a modes, position, rate, inertia box, checking whether it is using wheels or thruster's actuator, it shows initial condition of the servo table.

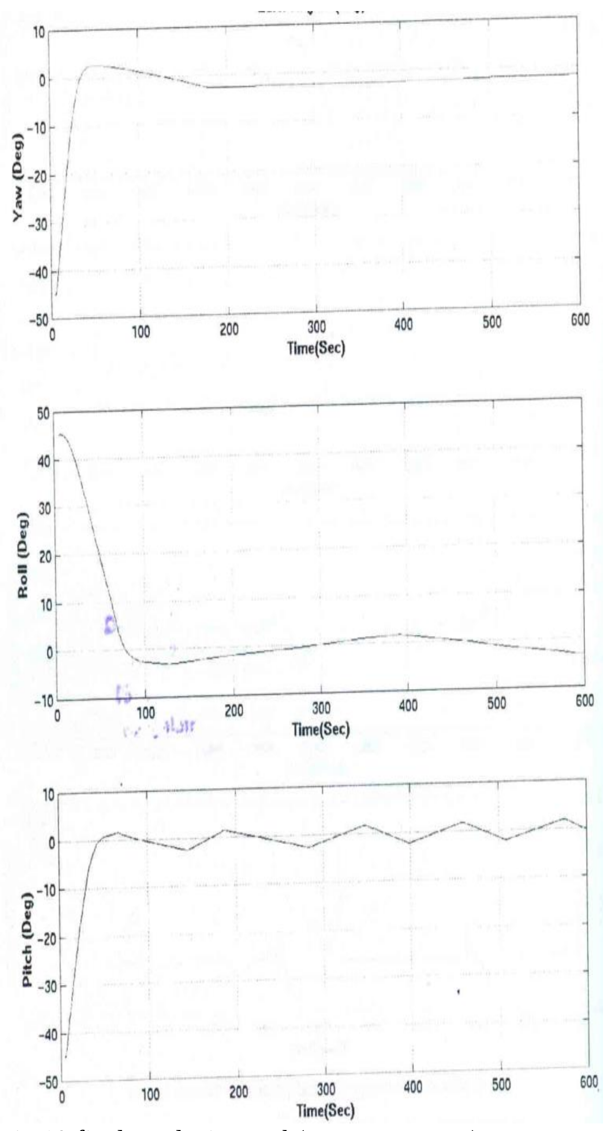

Fig 10:final results in graph(converges error)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

Fig 10 is the final result of distributed computing of hardware in loop simulation. It shows yaw, roll, and pitch graph which is converging to zero. It has given error as -40,40,-40 as shown in graph.

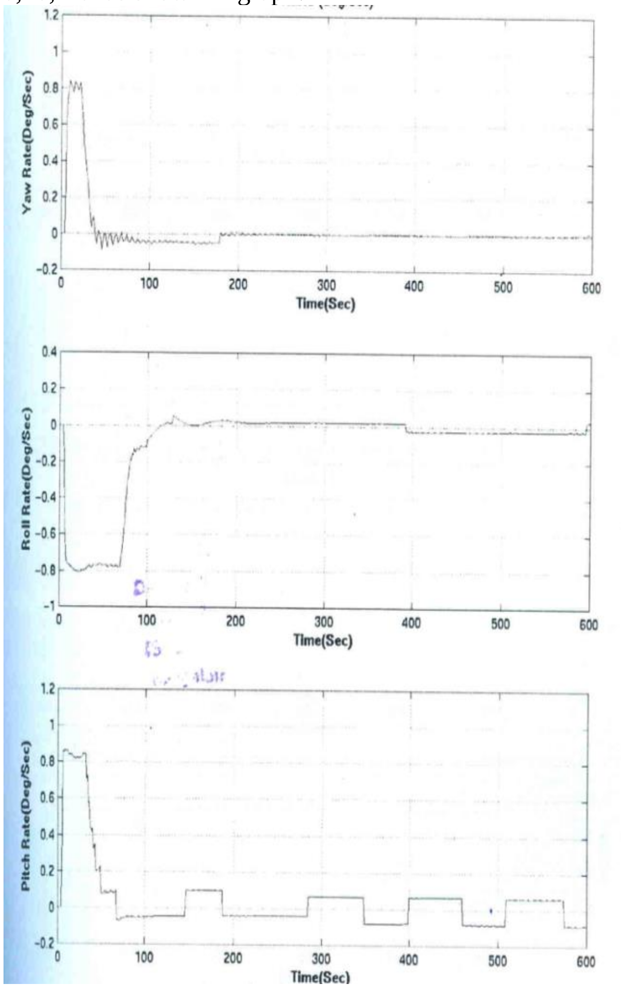

Fig 11: Rates(deg/sec) graph

## VIII. CONCLUSION AND FUTURE WORK

This paper presents a distributed approach for computing system solutionto achieve the high performance Hardware-In- Loop Simulation test platform. The distribution of tasks among three major computing elements was presented. Client-Server architecture based TCP/IP socket communication was used to transfer the data with no loss. By combining the Data Acquisition System and Dynamics Simulation System, reduction in closed loop delay was achieved from 32.0ms to 8.0ms and also combining the Telemetry and Telecomm and and Dynamic Simulation in single framework. The futures work aims at reducing this delay to maximise mission simulation.

## REFERENCES

1. Evolution of Hardware in Loop For Saellites, Indian Space Research Organization Bangalore,July 2008.
2. Jr, Harry Katzen., An introduction to Distributed Data Processing, Petrocelli Book, 1978.
3. Xue Ming, Zhu Changjun, “The Socket Programming and Software Design for Communication Based on Client/Server,” 2009 Pacific-Asia Conference on Circuits,Communications and systems, p 775-777.
4. Modifications of HILS Facility , Indian Space Research Organization Bangalore, December 2004.
5. (2002) The IEEE website. [Online]. Available: http://www.ieee.org.