Pseudo-Dynamic Testing

The World of Pseudo-Dynamic Testing

With the increasing development of faster DSP control systems, the Pseudo-Dynamic test method is fast becoming a viable alternative to large scale shaking table systems for the evaluation of dynamic integrity of large-scale structures. This method incorporates a hybrid of testing methodologies, in which computational modeling is carried out in conjunction with experimental testing of structures or sub-systems of structures. The computational element to the method involves a stepwise resolving of forces generated in the theoretical structure by a ground excitation. The dynamic forces are calculated using Mass and Damping Matrices, whereas the static Restoring forces are measured experimentally.

A World of Experience…

Servotest is a World Class Test and Motion Simulation Company, with experience of operating around the globe, for multi national corporations and smaller specialist companies and Government Departments. Since the 1950's our engineers and equipment has been at the forefront of our industry. Product and Service quality is maintained by a program of continuous training and development of our people and equipment. We operate in all of the key industry sectors for our market place, including Automotive, Marine, Civil Engineering, Aviation, Defence, Aerospace and Traction. The company holds both ISO14001 and 9001 Quality accreditation marks and is a member of many national & international trade organisations.
Ma + Cv + Fr = Fin

By modeling the dynamic elements of the above equation, it is possible to perform the experimental part of the test in ‘slow-motion’ in the computer's virtual timeframe. This not only reduces hydraulic power requirements, but also allows for better observation of the onset of structural damage. In some cases, it is possible to incorporate multiple elements of a structure into the model, leaving only a small proportion of the structure to be tested experimentally, again reducing experimentation costs.

Where:
M = Mass Matrix
a = Modal Acceleration Vector
C = Damping Matrix
v = Modal Velocity Vector
Fr = Restoring Forces
Fin = External Excitation Forces

By modeling the dynamic elements of the above equation, it is possible to perform the experimental part of the test in ‘slow-motion’ in the computer’s virtual timeframe. This not only reduces hydraulic power requirements, but also allows for better observation of the onset of structural damage. In some cases, it is possible to incorporate multiple elements of a structure into the model, leaving only a small proportion of the structure to be tested experimentally, again reducing experimentation costs.

A discrete time-integration is required in order to transform the modal acceleration and velocity vectors into modal displacements which are then applied to the structure.

Example of Simulink model of SDOF system
SOCKETS

SOCKETS is a powerful tool in the Servotest Digital Control System (PULSAR) that allows users to run SIMULINK models in the PULSAR control system in real-time. SIMULINK provides an intuitive graphical environment in which to build highly complex mathematical models, which can then be compiled into sourcecode using the Real-Time Workshop.

The method
The Newmark algorithms assume that:

$$M a_{m+1} + C v_{m+1} + r_{m+1} = F_{m+1}$$

$$d_{m+1} = d_{m} + DTa_{m} + DT^2 \left[(0.5 - \beta) a_{m} + \beta a_{m+1}\right]$$

$$v_{m+1} = v_{m} + DT \left[(1 - \gamma) a_{m} + \gamma a_{m+1}\right]$$

where $a_{m+1}$, $v_{m+1}$ and $d_{m+1}$ are the acceleration, velocity and displacement vectors respectively, at time equal to $(m+1) DT$; and $\beta$ and $\gamma$ are parameters selected by the user for stability and accuracy.

When $\beta = 0$ the next displacement is no longer a function of the next (unknown) acceleration so the integration method becomes explicit and if $\beta = 0$ and $\gamma = 0.5$ the popular central difference scheme is recovered, i.e:

$$d_{m+1} = d_{m} + DTa_{m} + 0.5 DT^2 a_{m}$$

$$v_{m+1} = v_{m} + 0.5 DT (a_{m} + a_{m+1})$$

This is the algorithm that is implemented by default in Pulsar. The user can modify the algorithm at will, within computational capacity of the DSP processor.

Continuous pseudo-dynamic testing
The conventional pseudo-dynamic method would require that, at each step, the actuators be ramped to the new desired position and held there while the new restoring forces be measured and used in the calculation of the next displacements.

The drawbacks of this technique are the load relaxation phenomenon occurring during the hold periods and the long time it takes to complete a test. Servotest implement the so-called continuous procedure [Casciati et al, 2006] whereby the servo-controller moves the actuator in such a way that the specimen follows very accurately the target displacement (due to the lack of discontinuities in the motion). The forces are measured at every control sample period (typically 1ms) and the equations of motion are integrated on the fly at the same rate. The next displacement is determined and the motion proceeds without interruption. To do so, for each gm discrete value of ground acceleration, a sequence of N acceleration values is computed by interpolation between gm and gm+1 (that is g0m, g1m, ..., gm, ..., gN-1m).

N interpolations
In these conditions, for a time scale expansion factor of 500 (each second of earthquake would take 500 seconds to test) we need $N = 2500$.

Discrete-time integration can be problematic due to numerical instability, which is the cumulative effect of numerical errors in the calculation. This has been a major focus of development of this testing method and has prompted variations in the way we integrate Acceleration and Displacement numerically.

SIMULINK provides a standard integrator block, which can be easily configured to perform step-wise integration using whichever method is most appropriate.
Servotest PULSAR Control System

Once the SOCKET has been built, it can then be linked into the existing DSP code that runs the servo-hydraulic mechanical system. By implementing the modeling element of the method at DSP level, the impact on the computational resources is low.

SOCKETS Implementation

- All Real-Time Signals are in DSP Memory, and are 32bit floating-point numbers.
- All DSP Functions can see all Signals (Data) in the DSP Memory.
- Function “inputs” are specified as a pointer to (the ADDRESS of) a Signal.
- A Function’s “outputs” are stored in the Parameter Block Memory which was allocated to the Function.
- A Function can be “instanced” (used) more than once, where each “instance” is controlled by a separate Parameter Block.
- The Host PC can access the DSP Signals in DSP Memory to control and monitor the System at Run-Time.
- The CONNECTIVITY of the Functions is defined by the Database.
- Signal NAMES are translated into ADDRESSES.
Servotest ‘SOCKETS WIZARD’ Software

- Manages the T.I. Tools – C Compiler, Assembler, Linker and many other supporting files.
- Creates the necessary Database files which create the Parameter Block(s) for the Socket Function(s).
- Automatically downloads the User Functions (DSP Executable Binary Code).
- Can Save and Restore previously compiled User Functions and Models.
- Supports creation of Models (with Simulink) on other PC’s on a network.
- Has ability to generate Run-Time adjustable User Parameters (Sliders) which can adjust Model Parameters during testing.
- Automatically checks User Functions for DSP Code which would compromise the Real-Time performance of the System, such as non-relocatable code, illegal and non-interruptable instructions.

Advantages
- Rapid Changes – “WHAT IF” testing, by modifying the Simulink model and recompiling (minutes) or by changing User Parameters at Run-time using Windows Sliders (seconds).
- Secure I.P. – Intellectual property, research, new techniques, can be incorporated into User models / functions and be tested without need to disseminate to the System Supplier to provide software.
- Visual Design, self-documenting – Using MATLAB and Simulink to design User Functions with diagrams. Integrated with PULSAR .Can mix User Functions with System Control, Input/ Output, Data Aquisition and other DSP signal processing - almost limitless possibilities.
- Rapid Changes – “WHAT IF” testing, by modifying the Simulink model and recompiling (minutes) or by changing User Parameters at Run-time using Windows Sliders (seconds). Real time running of simple single channel system running explicit algorithms.
Pseudo-Dynamic Testing using PULSAR and Remote Simulation Computer

For very large applications involving sub-structuring or complex implicit algorithms it is necessary to introduce fast parallel processing. The Servotest Pulsar controller already provides the ideal platform to perform pseudo-dynamic tests based on real-time commands generated by a separate simulation controller. It is a powerful digital controller able to control a large number of hydraulic actuators and whilst exchanging high resolution data with the outside world (external sensors and PDT simulation controller).

Pulsar automatically detects the calibration information of all the sensors which will make the practical PDT test set-up simple. Servotest Testing System is able to provide a solution that is both robust enough to perform well in a wide variety of applications and simple enough to be user-friendly.

An implementation of so-called explicit Newmark methods where the displacements achieved at each step of the algorithm are not actually monitored but are assumed to match the command. These are conditionally stable methods that are ideal for simpler structures.

An environment allowing third party algorithms written in C to run seamlessly on the PDT simulation controller. This makes the Servotest solution more attractive to those researchers who have developed, fine-tuned, and favour their own algorithms.

Implementation of a more complex family of integration algorithms (so called implicit integration methods) to be used with larger structures.
The combination of Linux and a powerful PC platform make it possible to update models and integration algorithms in synchronisation with Pulsar.

The SCRAMNet data is easily accessible to programs written in C. The models and integration algorithms can be written in C or C can be created from Simulink models using Real-time Workshop. Linux provides easy access to its real-time clock which can be used to create a 1.024 kHz signal.

Virtual testing can be performed by creating actuator simulations in Sockets and connecting the inputs and outputs of this simulation to SCRAMNet.