# A JAVA-BASED INFORMATION SYSTEM FOR WAYSIDE SENSING AND CONTROL


Larry F. Myers, P.E.*, Mike Lovette
Conrail Corp., 200 Fourth Ave. - Juniata, Altoona, PA, 16601-6702
*email: larry.myers@conrail.com

Charles C. Kilgus*, Judith A. Giannini, David C. Swanson,
Karl Reichard, Mark Mahon, Doug Mast
Applied Research Laboratory/Penn State University, State College, PA 16804
*email: ckilgus@psu.edu



## ABSTRACT

The advent of Java-based information systems is causing a new network system structure to replace the custom designed client/server architecture traditionally used at ARL/PSU for wayside sensing and control. Java applets provide the man-machine interface allowing access and control of the wayside system from multiple locations over a dedicated intranet or dial-up modems. The only user software required is a Web browser with Java providing the Graphical User Interface. ARL/PSU is implementing a Java-based information system in an experimental wayside rock slide detection system. To determine the location of the rock slide relative to the tracks, the sensor array provides both direction finding and range estimation. A Digital Signal Processing board in the computer at the sensor site (wayside server) records the sensor data and performs the function of impact detection and localization. The wayside server provides a Java-based graphical display of system information on a remote PC monitor via a telephone modem.


## INTRODUCTION

Rocks falling onto the tracks is recognized by Conrail to be a significant problem in the mountainous terrain typical of some of the heavy use operating areas. One such area is along the Mon Line, below Mt. Washington, in downtown Pittsburgh, PA., where an unstable rock face (Fig. 1) is being continuously monitored.

Currently, falling rock events along a right-of-way are detected by the falling material impacting a portion of a wire fence. There are sprung devices mounted along the length of the fence, at the pole locations, that are physically attached to the fencing. When the fence is stretched by a predetermined amount, the sprung device is activated. The activation of this device interrupts the current flow through the detection circuit. The result is a shunting of the Track Circuit, i.e. each rail is electrically connected to each other through a set of normally closed contacts on a relay. When the fence detection circuit is interrupted, the relay is de-energized, causing the rails to be shunted. The shunting of the rails will give an indication equivalent to that of rail equipment on that section, block, of track.

**Figure 1.** Mon Line rock face

This indication is called a TOL, Track Occupied Light. Regardless of what the dispatcher tries to do, he can not clear the circuit and provide a train with the proper signal to allow his movement over or through that block of track. The Engineer of the train will generally be given permission to proceed into that block at Restricted Speed. This means that he will proceed at a speed that will allow him to stop his train in half the distance that he can clearly see.

Unfortunately, experience has proven that not all falling rocks will destroy a fence, and the safety of a train crew, its contents, and a nearby community can be jeopardized. Furthermore, should a rock slide destroy the fence system, no further detection is possible until the fence system has been repaired. Often times, the desired detection along the right-of-way is in remote locations; hence, hampering inspection and repair of these systems. Additionally, these fence systems are vulnerable to vandalism, which can result in undesired false alarms and additional maintenance. A system is desired that offers low maintenance, low probability to vandalism, and high integrity. Conrail's Engineering Department has been working with Penn State's Applied Research Laboratory to develop such a system, a new Rockslide Detector that can sense abnormal ground vibrations.

As a first step in developing the system, a Rock Drop Experiment was conducted by a Conrail/ARL team at the Conrail Spruce Creek Site on September 10, 1996. Acoustic and seismic data from passing trains, sledge hammer blows and heavy rock drops were recorded and analyzed. Results demonstrated high probability of detecting the impact of large objects with low false alarm rate from thunder, passing trains or small rocks. The impact location algorithm showed reasonable precision in determining the impact position relative to the tracks.

Based on the success of the Rock Drop Experiment, Conrail has tasked the Applied Research Laboratory to develop a proof-of-concept experimental wayside rock-slide detection system. The experiment will be conducted on a section of track several kilometers long along the Mon Line in downtown Pittsburgh PA where rock slides are likely. A Java-based information system is being implemented. Java will provide a graphical display of wayside system information and allow remote monitoring and control of the site components.

This first introduction of Java in the proof-of-concept Rock Slide System has provided insight to the capabilities afforded by a true Java-based wayside information system. The general architecture of such a system has been developed and the Java realization at each level, from the distributed, remote man-machine interface down to the executable in the embedded sensor processors, is being explored.

Wayside sensor systems and wayside information systems currently under development include the IRRIS (Integrated Railway Remote Information System) (Ref. 1) and AWIS (Automatic Wayside Inspection System) (Ref. 2) at the Assoc. of American Railroads (AAR) - Transportation Technology Center Inc. (TTCI). AWIS includes two installed series of strain gauges that measure vertical and lateral loads. The data is transmitted to a central computer where user access is provided via the IRRIS data base. The TTCI Event Notification System (Ref. 3) includes a series of railroad cars instrumented to measure ride quality (accelerations) that notify the railroad of events via satellite link. The TTCI Vehicle Defect Detection System (Ref. 4) includes acoustic wayside sensors to detect faulty wheel bearings. The DOT Volpe Center Track Safety Research Program (Ref. 5) has developed cars that are currently running on Conrail in order to measure lateral strength of the track and detect track faults.

## ROCK DROP EXPERIMENT

On 10 September, 1996 a data collection experiment was executed at the Conrail Spruce Creek site. The experiment was designed to show that large and small rocks can be discriminated from each other and from passing trains, and that the sensor array can determine the direction and location of the rock impact area. The critical functions to be established are impact detection, direction finding, and localization with respect to the track.

A total of 14 drop events of a heavy weight (~1740 lb.), approximately 1.5 yards of gravel (loader's bucket), or some 30-80 pound rocks were dropped from the stone wall by the tunnel portal. Passing trains were recorded with every opportunity, and sledge hammer blows at each of 12 locations along the two tracks were recorded.

Six micro-machined accelerometers and two microphones made up the sensor array. The site plan (Fig. 2) (survey and chart provided by Michael Lovette of Conrail) depicts the layout of the micro-machined accelerometers (MMA), microphones, tracks, drop sites, and the headwall corners of the tunnel portals. The microphones are B&K model 4138 and the MMA's are EG&G (formally IC Sensors) model 3145-002 which have a sensitivity of 1 Volt per g ($g = 9.81$ m/s$^2$). The MMA's are numbered from 1 to 6 going counter-clockwise starting with sensor 1 located at coordinates $x = 16.2$ and $y = -20.1$ ft. The MMA's were used because they have a constant frequency response down to 0 Hz and an experiment objective was to determine the frequency response of a rock drop. The sensor data were collected and processed using an intelligent data logger/microcontroller, a pica-PC with memory, analog-to-digital conversion capability, and communication ports.

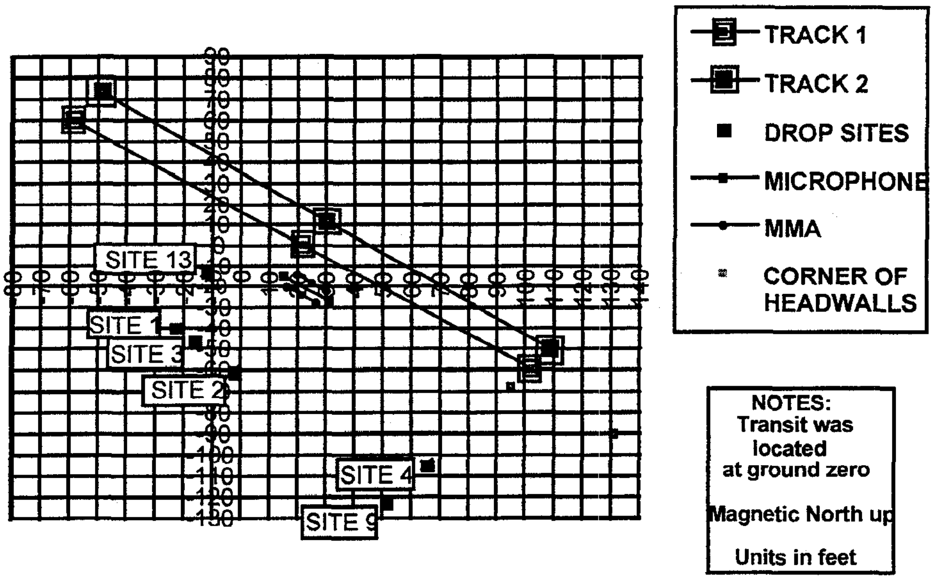


**Figure 2.** Site plan for the 10 September, 1996 rock drop experiment.

**Experimental Impact Detection Using Spruce Creek Data**

To determine whether or not a large rock has impacted the area, the sensor network must distinguish between a large impact and a passing train. This can be accomplished by recursively averaging the recorded data with several time constants and comparing results to distinguish an impact from other loud noises. The seismic/acoustic noise levels increase and decrease relatively slowly for a passing train as compared to an impact. The impact detection must be nearly independent of background noise levels, including time-varying levels due to weather and traffic. The integrated RMS levels from the accelerometers allow comparison of the signatures of a rock drop (Fig. 3) and a passing train (Fig. 4). Using the RMS output with a 0.1 sec. time constant (middle plot), the accelerometer output for the rock drop exhibits a rise time of approximately 0.076 sec. and decays to 10% of the peak value after approximately 0.51 sec. The RMS accelerometer output produced as the passenger train passed the sensor site has two distinct peaks compared to a single maximum for the rock drop. The second peak in the RMS signal has approximately the same magnitude as the first.

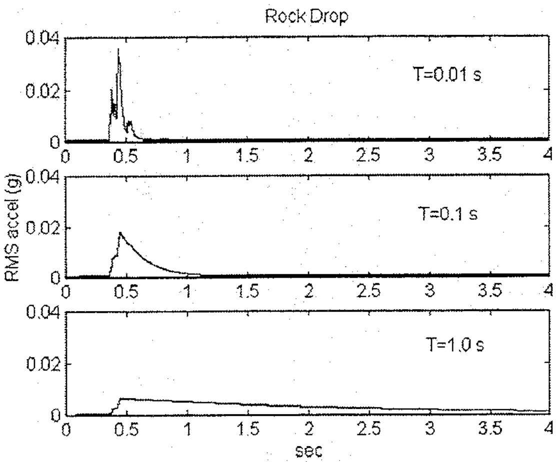


**Figure 3.** RMS signal levels from accelerometer #1 during rock drop for RMS time constants of 1.0 sec., 0.1 sec., and 0.01 sec.

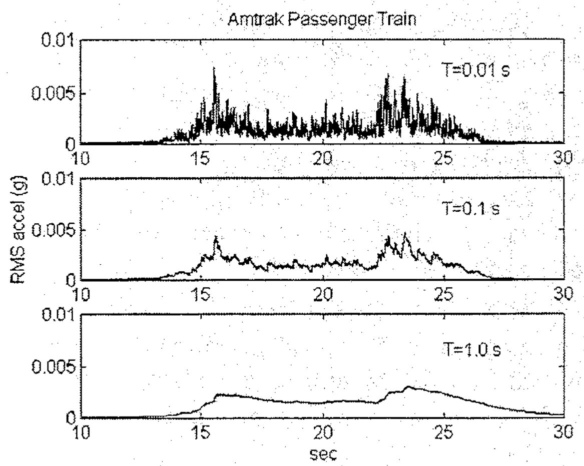


**Figure 4.** RMS signal levels from accelerometer #1 during Amtrak passenger train pass for RMS time constants of 1.0 sec., 0.1 sec., and 0.01 sec.

Using the RMS level computed with a 0.1-sec. time constant (middle plot), the rise time for the first peak is approximately 6.4 sec. The decay time from the first peak until the signal decays to 10% of the peak value is approximately 11.07 sec. corresponding to the time it took the train to pass the sensor site. The sensor system distinguishes rock slides from passing trains by comparing these features (rise time, fall time, and amplitude) using different integration constants.

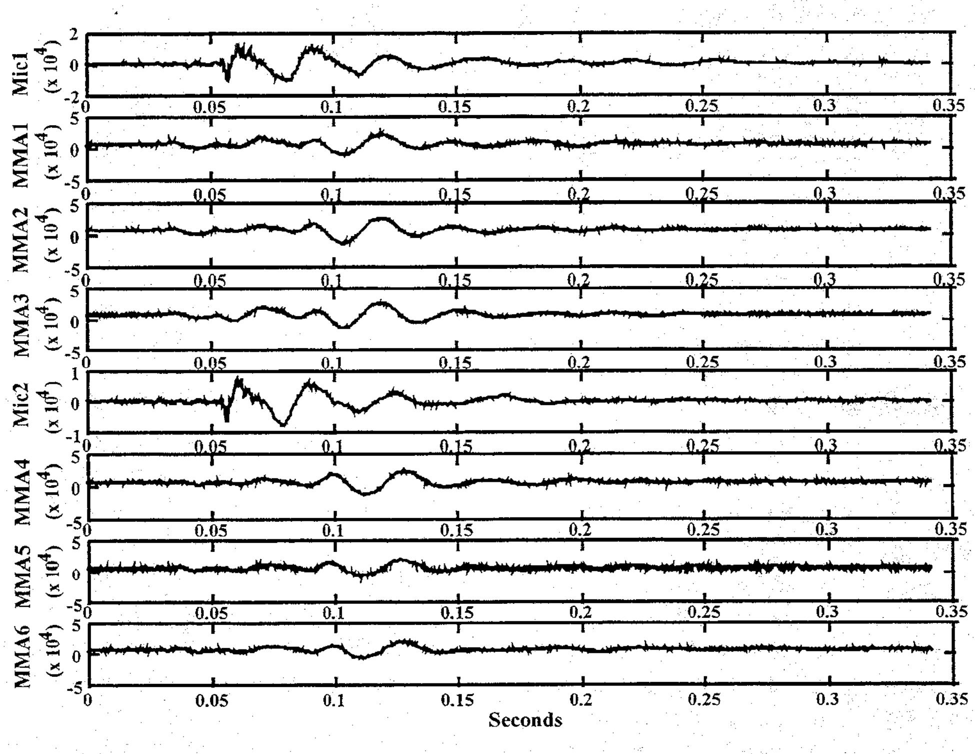


**Figure 5.** Typical acoustic and seismic accelerometer traces for a rock drop at site 2.

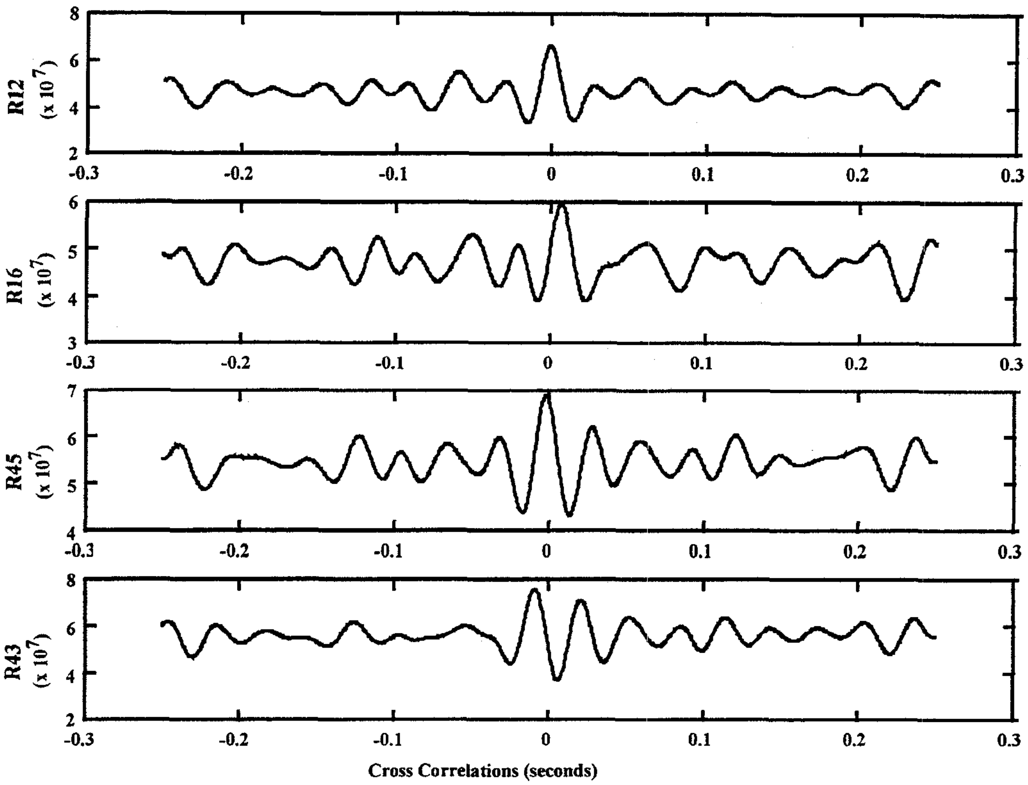


**Figure 6.** Cross correlation functions for seismic channels for a rock drop at site 2.

**Experimental Impact Localization Using Spruce Creek Data**

Typical acoustic and seismic acceleration traces for a rock drop at site 2 (Fig. 5) show the differences in acoustic and seismic waveforms. The six micro-machined accelerometers (MMA's) which have constant sensitivity from 200 Hz down to 0 Hz (gravity) are arranged in two parallel rows of three and the two microphones are centered between the rows at either end. Thus, the time-of-arrival for the two microphones is nearly the same while a slight delay is seen between the rows of MMAs. The acoustic signals also show significant high frequency response as seen in the sharp wave structure in the leading edge.

The seismic waves show significant dispersion of the impact energy. This is because a surface impact produces Rayleigh (transverse bending) surface waves (s-waves) in addition to pressure waves (p-waves). The s-waves spread circularly while the p-wave spread spherically. Thus, if the sensors are some distance from the impact, the dominant component seen are the s-waves. The wave speed for s-waves increases at higher frequencies (speed is constant for acoustic and p-waves) and the soil attenuation becomes extremely high as frequency increases. Careful examination of the s-waves (Fig. 5) shows the high frequencies arriving early and being attenuated relative to the dominant low frequency of around 20 Hz. The dominant low frequency is likely defined by the layered soil structure and the momentum of the weight impacting the ground. It appears plausible that the size and damage from a rock slide could be determined by the s-wave components.

Localization of the rock slide impact site is done using well-known time-delay-of-arrival (TDOA) techniques (Ref. 6-7). Cross correlations are calculated between the MMA signals (Fig. 6). The peaks of these functions are used to compute the TDOA for the waves relative to a reference MMA.

Using the measured TDOA's and the array geometry, an estimate of the impact site position is made. One problem of interest is the dispersion and complicated propagation in the seismic waves. A wave-speed independent localization algorithm has been developed which gives reasonable results by combining signal processing and array configuration in a unique arrangement for a particular track site. Optimization of the processing and array configuration is still a research topic.

This impact localization algorithm provides a reasonable position estimate relative to the tracks (Fig. 7).

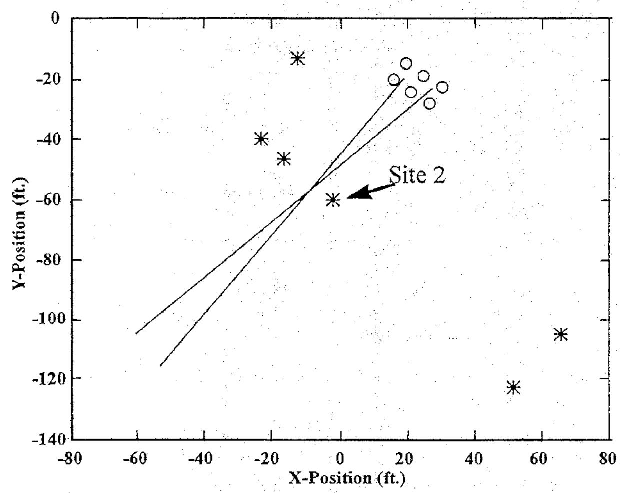


**Figure 7.** Impact localization for rock drop at site 2.

### Simulated Impact Localization Precision

The experimental localization results demonstrate that determining the array and sub array shapes that maximize the precision coverage of the impact algorithm is a significant issue. A promising sensor sub array configuration has been developed: splitting the n-element array into two n/2-element sub arrays preferably separated by a significant distance.

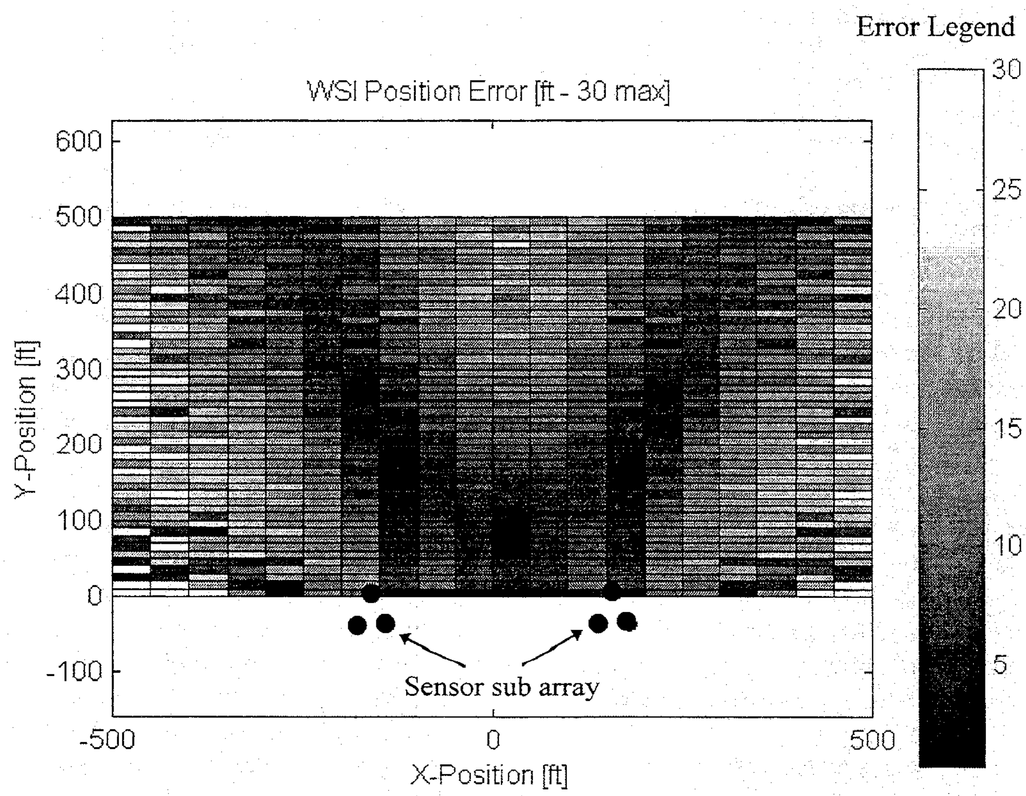


**Figure 8.** Localization error for 3-element sub arrays separated by 300 ft.

The geometric error for two triangular 3-element sub arrays separated by 300 ft. has been calculated (Fig. 8). The localization errors are typically held on the order of 6 ft. (< 2 m) or less throughout the region except for a narrow line along the axis of the two sensor sub arrays. The low localization error extends over a region nearly half a mile long. This arrangement, although not formally optimized, is seen as the most practical design for a railroad rock detection system. The arrangement can be used with both sensor groups on one side of the track, or, offset on opposite sides where the track follows one of the dark (very low error) bands indicated.

The effect of the sample rate on the seismic sensor time delay estimates as well as the introduction of a 1 millisecond random error in the time-of-arrival have been simulated (Fig. 9). A wave speed of 1500 ft./sec. was assumed. The results show that the sample rate should be at least 8 kHz and the sub array apertures should be on the order of 30 ft. on a side. The range of low localization error is then on the order of the separation of the two sub arrays. Making the sub array separation larger reduces the sample rate and random error effects, but also increases the localization error everywhere. Because the time-of-arrival differences are used to determine the angle-of-arrival, the susceptibility to random errors is rather small. However, the sample rate causes the available time "slots" for wave arrival to be truncated. This appears to be the root cause of the larger errors beyond the space between the two sub arrays. In this outside region, the angles measured by the two sub arrays are nearly identical, so small angular error translates into a large localization error. In the middle region, the two sub array angles are quite different, so small errors in angle do not translate into large localization errors.

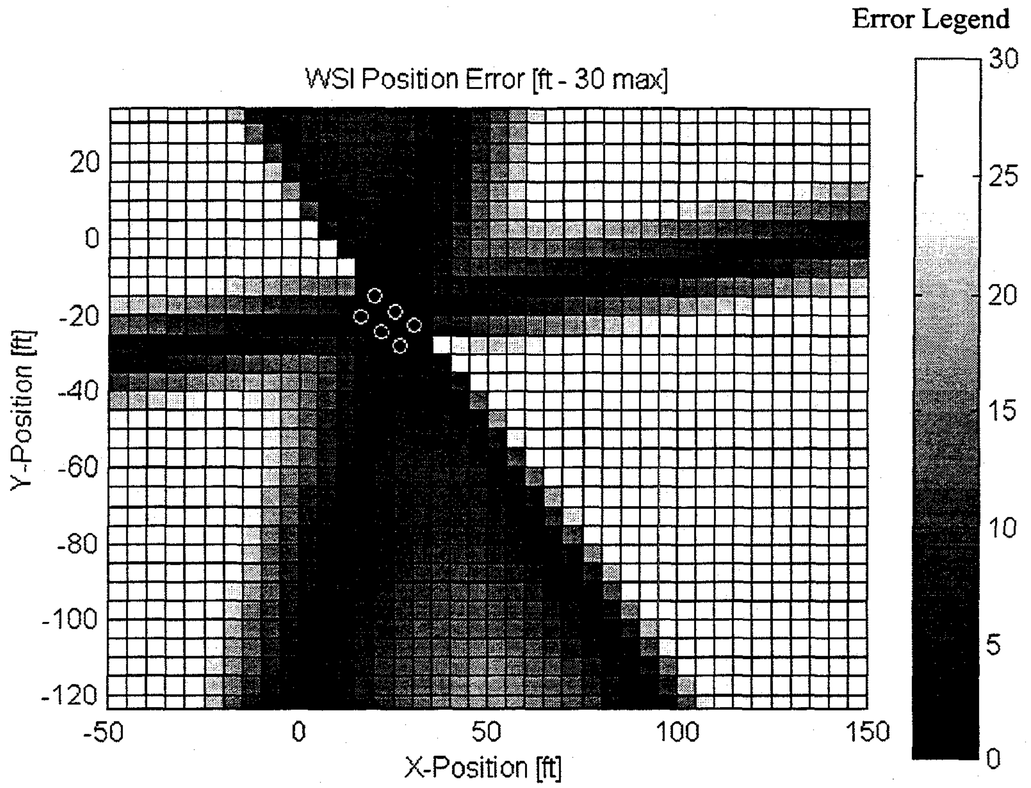


**Figure. 9** Localization error impacts of random time delay and finite sample rate.

## Rock Drop Experiment Implications for the Rock Slide System

The Spruce Creek data demonstrates that even a poor sensor arrangement and sensor device can reasonably locate the impact position of a large object. From simple analysis of the received waveforms, it has been shown that passing train and small rocks will not cause false alarms. A seismic array consisting of two sub arrays separated by a distance is the best arrangement for detection of rock-slide impacts along a railroad. The impact localization algorithm allows reasonable precision in determining the impact position relative to the tracks.

Since railroad road beds are a fairly consistent ground, it appears that a good geophone will produce highly sensitive responses with no sensor power requirements. During the experiment, severe electrical noise problems with the MMA's were experienced due to a capacitive load of the cables. The geophone is selected as the sensor of choice in this application.

## ROCK SLIDE SYSTEM

Based on the success of the Rock Drop Experiment, Conrail has tasked the Applied Research Laboratory to develop a proof-of-concept experimental wayside rock-slide detection system. The experiment will be conducted on a section of track (Fig. 10) several kilometers long through an area outside Pittsburgh PA.

### System Design

The experimental system (Fig. 11) employs a network of wayside sensors to detect and locate rock slides in order to minimize disruption to railroad operations. Recent developments in acoustics and seismic detection are employed to discriminate large and small rock slides from each other and from passing trains. To determine the location of the rock slide relative to the tracks, the sensor array provides both direction finding and range estimation.


**Figure 10.** Rock slide detection system site diagram.

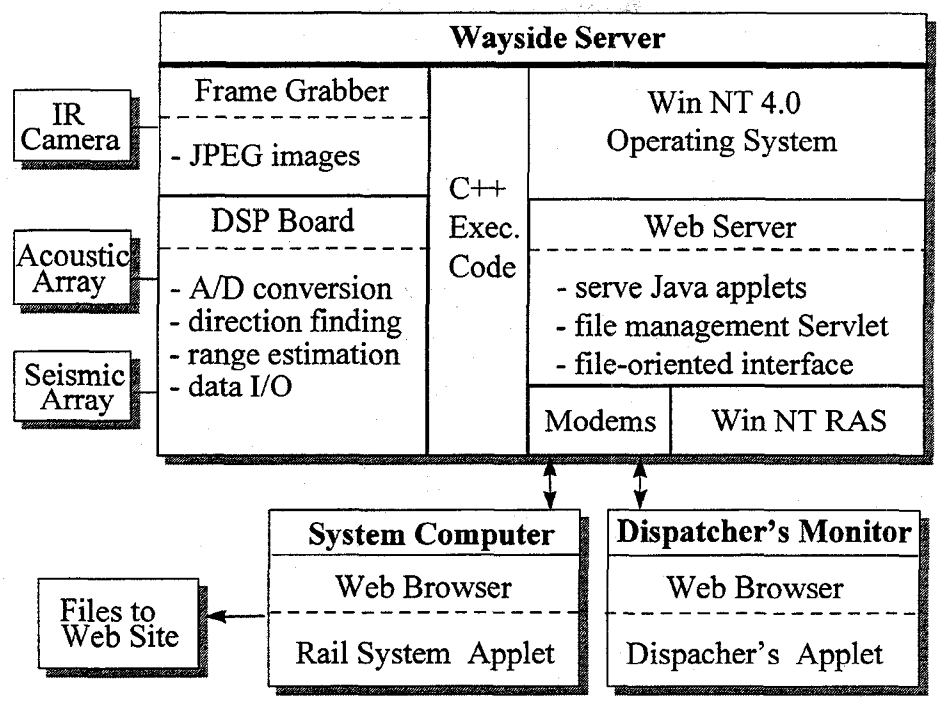


**Figure 11.** Rock-slide system block diagram

IR camera-views up and down the tracks provide independent confirmation of detected rock-slide events. The frame-grabber board periodically captures IR camera snapshots and provides JPEG encoded images. A Digital Signal Processing (DSP) board records the sensor data and performs the functions of impact detection and localization. C++ Executive code manages the boards and communications, providing data files for the Java information system.

The Web server running on the wayside computer provides a Java-based graphical display of system information on the remote Dispatcher's Monitor via a telephone modem. Camera images are periodically transmitted to the dispatcher and displayed. An alert is sounded when a rock slide is detected, a message with the computed location is displayed and a sound clip of the event is played. The combination of camera images and sound is intended to allow an independent conformation of the event by the dispatcher. For testing of this proof-of-concept experiment, Conrail engineers will fill the role of dispatcher.

A second telephone line allows the wayside server to provide a Java applet for system monitoring and control by the system engineers at ARL. The system data are archived by the System Computer. A password-protected Internet site provides an archive of site information, maps and photos, allows after-the-fact viewing of rock slide events, and provides information on and demonstrations of the rock slide system components.

**Java-Based Rock-Slide Information System**

The Java-based information system provides two user interfaces that allow monitoring and control of the wayside components. Both are realized as Java applets that reside on the wayside server as described in detail in the following section on Java-Based Wayside Information Systems. Team members can download these applets for execution in the Web browser of their local computers.

The applet, DispatchAp, provides a hands-off display of the sensor data provided by the wayside sensors and generates alert information when rock-slide events occur.

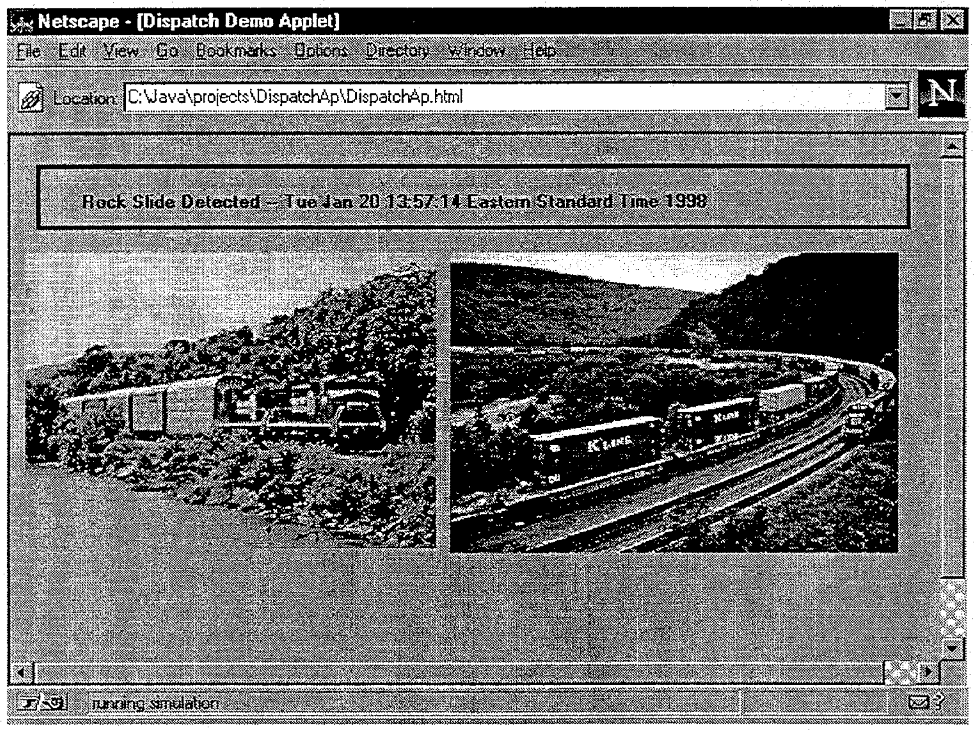


**Figure 12.** Screen snapshot of the applet DispatchAp.

The applet RailSystemAp provides a remote monitor of the health and status of the wayside components, allows control of the configuration of the remote executables, and allows near-real-time access to continuous wayside sensor data.

**DispatchAp** - As indicated by a screen snapshot (Fig. 12), IR images up and down the tracks are read from files in the wayside server and displayed. A text message indicating the alert status is displayed at the top of the images and marked with the time if a train or rock-slide event has occurred. The message frame is coded green for normal system operation, blue for train events, and red for rock-slide events. The text and frame color are read from files on the wayside server that are periodically overwritten by the C++ Executive code.

The occurrence of a rock-slide event sound an audio alarm and causes the audio measured during the event to be played on the dispatcher's computer. The applet reads the alert and audio files from the wayside server in ".au" format. These files are written by the C++ Executive code.

**RailSystemAp** - As indicated by a screen snapshot (Fig. 13), the top-level menu presented by the applet allows detailed displays for System Health, System Configuration, or System Monitoring to be selected. If the System Health Menu indicates a failure in a wayside component, the System Health selection button on this top menu is marked with red exclamation points.

When selected, the System Health display presents a block diagram of the system with the health of each component marked. A System Health text message is displayed.

The System Configuration menu allows values to be set for the executable code on the wayside processors. The values selected are written to a file in the wayside server that is monitored by the C++ Executive.

The System Monitor menu allows near-real-time measurements from each of the wayside sensors to be viewed. This function is intended to provide a tele-experimentation capability for the system.

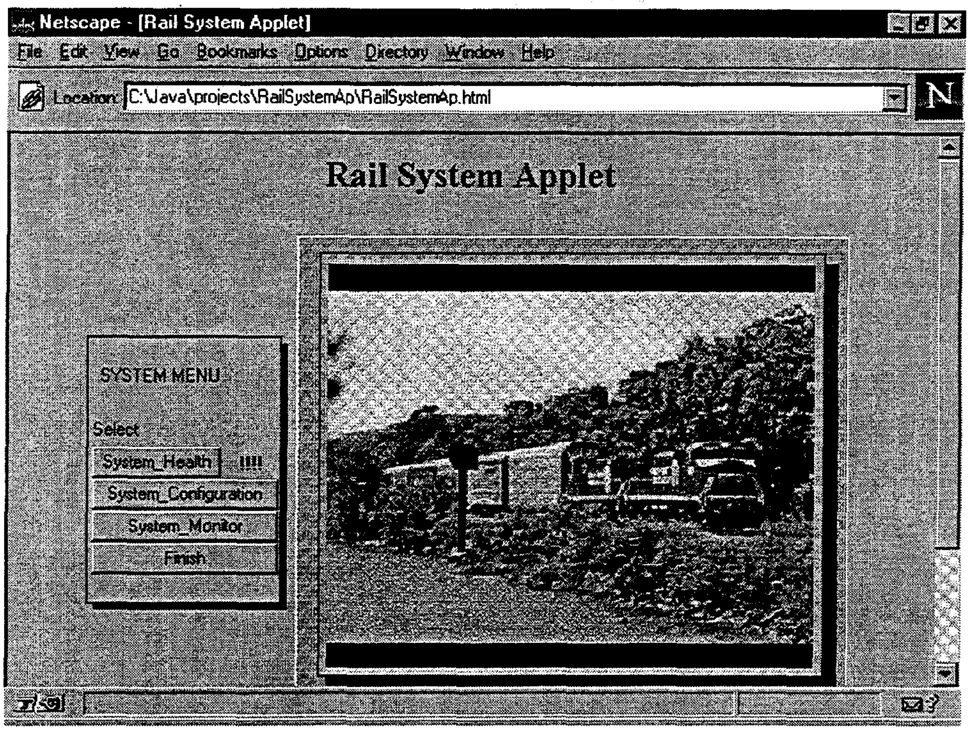


**Figure 13.** Screen snapshot of the applet RailSystemAp

**Internet WWW Site**

The Rock Slide System Web site includes a public and a password-controlled domain. The site includes: a tutorial on the system design, Conrail documentation of the geology and history of the experiment site, applets that allow visual and audio replay of selected experiment events, and documentation of the system hardware and software.

**JAVA-BASED WAYSIDE INFORMATION SYSTEMS**

The advent of Java-based information systems is causing a new network system structure (Fig. 14) to replace the custom designed client/server architecture traditionally used for wayside sensing and control. Java applets and applications can provide the processing executables for all system elements or can link to native code written in C/C++. An excellent introduction to Java information systems in embedded sensor and control applications is provided by the IEEE Computer Society Magazine (Ref. 8). A number of recent books (e.g., Ref. 9) cover the implementation of Java.

**Man-Machine Interface**

The Java applet, DispatchAp, described in the previous section is an example of the Graphical User's Interface (GUI) capabilities offered by Java. The only user software required is a Web browser. The Java applet is stored in the wayside server, called by the browser, and executed in the browser's Java Virtual Machine (JVM) on the User's computer. This provides complete machine independence, because the browser provides the machine-specific User interface. The Java applets therefore allow access and control of the wayside system from multiple locations over a dedicated intranet, the Internet, or dial-up modems. Execution of the applet in the User's browser is controlled by a security manager that prevents local system calls (file operations, etc.) on the User's computer. This protects against viruses and other computer security violations.

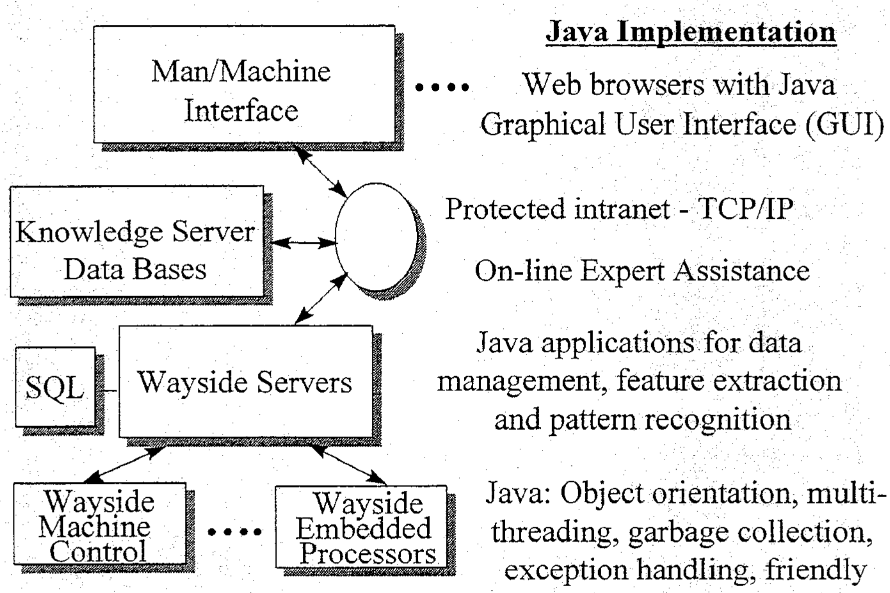


**Figure 14.** Java-based wayside sensor information system block diagram.

### Network Protocol

Communications with the wayside server using either a dedicated intranet, the Internet or dial-up modems use the ubiquitous TCP/IP net protocol and standard, e.g., Ethernet, network communications packages. Asynchronous network communications are provided by standard Java classes. Password security is provided on all net interfaces. Socket-level encryption can be provided for very sensitive applications.

### Application Servers and Data Bases

The User's Java applets allow interaction with applications servers and data bases over this network structure. Of particular importance, Java classes provide friendly interfaces to SQL data bases. Interaction with Knowledge Servers can provide expert assistance in managing the system and extracting data.

### Wayside Servers

The Java networking functionality is provided by the Web server on the wayside machine. The User's Java applets are stored by the server for downloading over the network and execution on the User's machine. Java also allows the server to provide server-side applets or "Servlets" that execute in the Java Virtual Machine on the server. These Servlets can communicate data to and receive commands from the User's applets over network sockets. The server JVM imposes no security manager, allowing the Servlets to access wayside data bases and perform system calls on the wayside machine including file access and execution of native C/C++ code.

Java also allows Applications to execute on the wayside machine. These Applications run in a JVM supported directly by the operating system. Since the wayside machine has access to data from the entire array of wayside embedded sensor processors and machine controllers, these Applications can provide high order functions, such as, data management, feature extraction, and pattern recognition. Network functionality is available via sockets to the Servlets on the Web server. This allows near-real-time monitorig and control of the executable configuration and data management by the User.

### Wayside Sensing and Control

The wayside sensors and controllers can generate a high volume of narrowly focused information. Embedded processors perform local operations for data reduction, such as, Fourier transforms and data compression, averaging, editing or selection. These operations are typically performed by machine-level code. Microprocessor JVMs are now available and the possibility of downloading Java executables to this lowest system level is a subject of research interest. Java would allow maintenance of the executable on the server with concomitant ability to modify the functionality of the wayside sensors and controllers over the network.

### Java Programming Language

As a programming language, Java is described as very "friendly", and extensive use has shown this to be true. It provides object orientation, multi-threading, interfaces, garbage collection and exception handling. The structure naturally allows Object Oriented Analysis and Design (OOAD).

Java is typically not currently used in real-time (e.g., robotics) or synchronous (e.g., low-level controller) applications. Real-time applications are limited by the interpreted (lower speed) operation of Java, although just-in-time compilers are becoming available. Synchronous applications are limited by Java's autonomous garbage collection.

## REFERENCES


1) Association of American Railroads WWW site at: http://www.aar.com/ttc/projects/project.htm. and select IRRIS button on the applet menu.
2) ibid., and select AWIS button on the applet menu.
3) ibid., and select Main button on the applet menu.
4) ibid., and select Vehicle Defect Detection button on applet menu.
5) Carr, G. A., Stuart, C., "Performance Based Tie/Fastener Inspection Technique Using the Gage Restraint Measuring System," SPIE, vol. 2458, June 7, 1995.
6) Piersol, A. J., "Time Delay Estimation Using Phase Data," AASP, 29(3) 471-477, June 1981.
7) Owsley, N.L. and Swope, G. R., "Time Delay Estimation in a Sensor Array," AASP, 29(3) 519-523, June 1981.
8) Special section: "Java and Beyond: Executable Content," Computer, the IEEE Computer Society Magazine, 30(6), 49-76, June 1997.
9) Naughton, P., Schildt, H., "Java 1.1: The Complete Reference, second ed.," McGraw Hill, 1998.