Implementing the ElarmS Earthquake Early Warning Algorithm on the Israeli Seismic Network

by Ran N. Nof and Richard M. Allen

Abstract  Earthquake early warning systems (EEWS) are being operated and tested increasingly around the globe in recent years. Following the Israeli government’s decision to build an EEWS in Israel, and as the Californian EEWS (ShakeAlert) moves toward its operational phase, we demonstrate implementation of one of its three algorithms, ElarmS, to the Israel region. We provide new tools and approaches for implementing and assessing ElarmS outside of California. The main challenges of this research are to identify, verify, and adjust the embedded location-dependent parameters in ElarmS to the Israeli region, utilizing an unoptimized seismic network and low seismicity rate. To this end, we run ElarmS in three different modes: (1) historical playbacks, (2) real-time continuous data processing, and (3) simulated data playbacks. These modes enable us to overcome the limitations of low seismicity rates in the region and evaluate the performance of ElarmS with the network that is currently available. We use historical playbacks to adjust the magnitude estimation equations of ElarmS. We then analyze real-time processing results and provide detailed analysis of two significant events in the region ($M_D$ 5.5 and 4.4). Finally, we provide the first case of how to use synthetic data to evaluate the performance of ElarmS. We find that alert times are mostly affected by the network geometry and also by data delays. Alerts are typically issued within 80 ms after the arrival of the required four $P$-wave triggers data to the system. Magnitude estimations are reliable for events with $M_D > 3.5$ within 100 km of the Israeli network using a locally adjusted magnitude relation equation.

Introduction

Earthquake early warning systems (EEWS) are being adopted around the globe and are currently operating in Mexico (Espinosa-Aranda et al., 2011), Japan (Nakamura, 1988), Taiwan (Hsiao et al., 2009), Romania (Ionescu et al., 2007), California (Kuyuk, Allen, et al., 2014), and being tested in Italy (Satriano et al., 2011), South Korea (Sheen et al., 2014), Turkey (Erdik et al., 2003), and other places. Several algorithms for EEWS exist, such as ElarmS (Kuyuk, Allen, et al., 2014), PRESTo (Satriano et al., 2011), Virtual Seismologist (Cua et al., 2009), OnSite (Böse et al., 2009), and UrEDAS (Nakamura and Saita, 2007). The basic concept of EEWS is to detect and estimate earthquake parameters, such as location, magnitude, and origin time, in the shortest amount of time possible and to deliver an alert to populated areas before the arrival of more destructive waves. This goal is achieved by processing instrumental measurements of velocity or acceleration from one (e.g., OnSite) or more (e.g., ElarmS) stations and estimating magnitudes based on proxies such as the maximal amplitude or the frequency content of the first few seconds after the arrival of the $P$ or $S$ wave.

In California, three algorithms are being tested under the California Integrated Seismic Network (CISN) ShakeAlert system, with an aim to demonstrate the feasibility of EEWS in California. The ShakeAlert decision module (DM) combines the event estimations from all three algorithms implemented in parallel (OnSite, Virtual Seismologist, and ElarmS) and reports the most probable earthquake magnitude and location to a group of test users from private industry and emergency response organizations in California (Böse et al., 2014). The ElarmS algorithm is maintained at the University of California Berkeley Seismological Laboratory and is under constant evaluation and development.

Recently, ElarmS successfully provided an alert for the 24 August 2014 South Napa $M_w$ 6.0 event. Performance evaluations for 2014 show that ElarmS sent out successful alerts for all significant earthquakes and aftershocks within the California border ($M_w \geq 4.5$, 10 earthquakes), with no false alerts and within 0.5 magnitude units of the catalog magnitudes. The alert time for the Napa mainshock was 5.1 s after the earthquake origin time and was dependent on the density of the seismic network around the epicenter (Kuyuk
ElarmS is highly customized to California and its various real-time (RT) networks. This localized customization includes many parameterizations and models, such as the relationships of the main magnitude proxies (maximum displacement $P_d$ and dominant period $T_p$; Allen et al., 2009); the fixed event depth of 8 km; a velocity model for the California region; and other factors. Implementing ElarmS algorithms in a different region is expected to require some adjustments to the user-defined, or the hard-coded, parameters (Sheen et al., 2014). In this work, we lay out the implementation of ElarmS EEWS algorithms to the Israeli Seismic Network (ISN) as the first example of the algorithm’s RT performance outside of California.

Israel is located adjacent to the Dead Sea Transform (DST), a tectonically active plate-boundary fault system (e.g., Garfunkel et al., 1981). The DST and its branches, the Yamouneh, Roum, and Carmel faults (Fig. 1), are capable of producing earthquakes with maximum magnitudes of $M_w$ 7.5–7.8 (Yucemen, 1992; Shapira and Hofstetter, 2007; Hamiel et al., 2009; Levi et al., 2010). Based on paleoseismic, historic, and instrumental records, recurrence times are on the order of 100 and 1000 yrs for $M_w$ 6 and 7 earthquakes, respectively (Shapira and Hofstetter, 2007; Levi et al., 2010). The most recent destructive earthquake along the DST was the 1927 $M_s$ 6.2 earthquake near Jericho (Shapira et al., 1993), which led to 285 deaths and ~1000 injured in the area (Avni et al., 2002). Given the growth in population during the past century and the expected recurrence interval of destructive earthquakes, the increased seismic risk for Israel has led the Israeli government to instruct the Geological Survey of Israel to establish an EEWS for Israel. Following recommendations of an international committee (Allen et al., 2012), the proposed system would include an upgrade to the current ISN; which translates into adding ~100 new stations of strong-motion accelerometers and broadband velocity instruments along the major fault line of the DST and its Carmel fault branch (Pinsky, 2015). The collected data would be processed by an EEW algorithm in order to deliver rapid alerts for potentially damaging earthquakes.

The main challenge of our current research is to identify, verify, and adjust the embedded location-dependent parameters in ElarmS to Israel and the broader DST region. These goals are achieved by running the ElarmS system using data from the ISN in different modes: (1) historical playbacks: processing archived data and historic records collected at the Geophysical Institute of Israel (GII) catalog location marked as black circles. Missed events are marked as orange circles and False events as red circles. Stations used for playback are marked by triangles. Generalized major fault systems are marked as black lines: DST, Dead Sea Transform; CFS, Carmel fault system; YMN, Yamouneh fault; and RM, Roum fault. Major cities are marked by white stars: JER, Jerusalem; AMN, Amman; TLV, Tel-Aviv; and GAZ, Gaza. The inset shows the global location of the map in the red rectangle.

Figure 1. Events locations calculated by ElarmS. First reported locations of “Matched” events are presented as green circles scaled by estimated magnitude. Gray lines point to the Geophysical Institute of Israel (GII) catalog location marked as black circles. Missed events are marked as orange circles and False events as red circles. Stations used for playback are marked by triangles. Generalized major fault systems are marked as black lines: DST, Dead Sea Transform; CFS, Carmel fault system; YMN, Yamouneh fault; and RM, Roum fault. Major cities are marked by white stars: JER, Jerusalem; AMN, Amman; TLV, Tel-Aviv; and GAZ, Gaza. The inset shows the global location of the map in the red rectangle. collected in RT (Kuyuk, Allen et al., 2014), which meant that playback capability could not be used efficiently for new data. To this end, a set of tools were created to analyze ElarmS results in RT and in playbacks (i.e., RTP or ATP), using archived or simulated data (see Appendix). Below, we present the results of our analysis of historical data playback processing for 39 events with coda magnitudes (Shapira, 1988) $M_D > 3$, between January 2012 and May 2015, fol-

and Allen, 2013b; Allen et al., 2015). Within the ShakeAlert system, ElarmS frequently provides the most rapid alerts and rarely issues false alerts.
lowed by an analysis of different aspects of RT processing and performance, including detailed system performance for two $M_D > 4$ events. We then provide a summary of simulated data playback processing of four earthquake scenarios. Finally, the implications of implementing ElarmS outside of California, and more specifically to the Israeli region, are presented in light of these results.

### Historical Data Playbacks

The high seismicity rate in California provides ample RT data to regularly evaluate ElarmS performance. However, in Israel, the lower seismicity rate in the region along the DST (based on 60,000 yrs of prehistoric-paleoseismic, historic, and instrumental records) indicates lower recurrence intervals of 5 and 15 yrs for $M_w \geq 4.5$ and $M_w \geq 5.0$ earthquakes, respectively (Hamiel et al., 2009). Thus, it is essential to use historical data to evaluate ElarmS near the DST, in addition to analyzing the RT performance of the system in Israel with smaller magnitude earthquakes.

The ability to run historical data in RTP is available within the native ElarmS. However, running multiple events spanning several minutes is a time-consuming task. Therefore, an ATP capability was introduced into the ElarmS code for this research, allowing the algorithm to run in simulated RT. This method of playback enables us to run historical data in ~30% of RT (~3 times faster than RT) depending on the amount of data (i.e., number of traces, sampling rate) and server speed. For example, by using five simultaneous processors, 21 min of 239 traces (23 MB) are processed in 412 s (~30% of RT). Moreover, processing time can be further reduced if data are provided in packets (typically of 1 s) ordered by packet starting time. This process of packetization reduces the processing time for the above example to 120 s (~10% of RT). Therefore, a whole day’s worth of prepacked RT data might be completed in ~2.5 hrs.

We note that playbacks can only be used to compare ElarmS performance under different scenarios (including parameter settings) and should not be used as a method of reproducing the exact chain of processing as it would play out in RT.

#### Playback Data and Processing

We processed data from 39 events ($M_D > 3.0$) in the GII catalog in ATP mode (Table 1). The dataset includes all of the events categorized as earthquakes by GII analysts, occurring from February 2012 to May 2015 within the geographical boundaries of 29.0°–34.5° latitude and 33.0°–37.0° longitude (see Fig. 1 for event locations).

ElarmS only issues an alert if four or more stations are triggered, and only when those alerted stations represent more than 40% of the active stations located within the epicentral distance radius to the farthest triggered station. In addition, several threshold tests are performed to prevent alerting on teleseismic or low magnitude ($M_w < 2$) events (Kuyuk, Allen, et al., 2014). We use the default settings of ElarmS, as used in the ShakeAlert system, and provide both a channel list and a set of parameters corresponding to the ISN. In addition to data archived from 23 broadband (BB) and short-period ISN core stations, we use available data from neighboring networks including five BB stations from the Jordanian Seismic Network (JSN) and seven BB stations from the Geophone Network (GE) located along the DST, Cyprus, and Turkey, all of which are archived at the GII. Data were additionally reorganized into 1 s packets. ElarmS continuously re-estimates event solutions (i.e., origin time, location, and magnitude) as new data arrive, potentially improving the solution accuracy. In our analysis though, only the first alert is evaluated because the initial alert is expected to have the most impact on the earthquake mitigation actions taken. We quantitatively estimate the ElarmS Performance Score (PS) as the ratio between real detected and alerted events (true) and the sum of True, undetected real events (missed), and false alerted events (false):

$$PS = \frac{T}{T + M + F} \times 100,$$

in which $T$, $M$, and $F$ are the true, missed, and false events, respectively.

#### Playbacks Results

Of the 39 events that we looked at, ElarmS produced triggers and issued alerts for 38 of these events. The remaining event was also detected, but no alert was issued because it failed the teleseismic test (Kuyuk, Allen, et al., 2014), which marked it as a distant event. In addition, one false event was generated and ElarmS issued an alert due to the problem of associating distant ($R > 300$ km) station triggers with a true event. The PS for the 39 historical events is 95.0%. However, data tested here include packets from only 10 min before to 10 min after each event so false alert susceptibility was not fully tested.

We distinguish between station magnitudes (estimated at each individual station) and event magnitudes (calculated for an event solution based on the average of the contributing station magnitudes). ElarmS station magnitudes ($M_w$) are estimated separately at each station that were triggered, based on the maximum displacement ($P_d$) with the relation (Kuyuk and Allen, 2013a):

$$M_w = 5.39 + 1.23 \log_{10}(P_d) + 1.38 \log_{10}(R),$$

in which $P_d$ is in centimeters and $R$ is the epicentral distance in kilometers. This relation was calculated using offline data from three regions: northern and southern California archives, Japan, and RT results from California, resulting in a 0.01 average magnitude error and 0.31 standard deviation (st. dev.) in magnitude errors (Kuyuk and Allen, 2013a). The event magnitude is calculated as the average of the station magnitudes, omitting stations with low signal-to-noise ratio.
from the network are more prone to large errors. This is
errors correspond to the distance between the epicenter and
exceeding 5 s (Fig. 8.7 km, whereas 24% exceed 50 km (Fig.
station magnitude estimates used for event magnitude calcu-
\[ \text{log} \left( P_d \times \frac{R}{R_{\text{ref}}} \right) = 1.041 M_D - 10.031 \]
\[
M_D = 5.7935 + 0.96061(\log_{10} P_d + \log_{10} R),
\] (4)
in which \(P_d\) is now in centimeters.

We recalculated the event and station magnitudes using equation (4), which yielded a significant improvement and subsequently eliminated the magnitude offsets (Fig. 2, bottom), with an event mean magnitude error of 0.08 and st. dev. of 0.34 (similar to the calculated st. dev. by Kuyuk and Allen, 2013a, for a more global dataset). Station magnitudes errors were further reduced to a mean value of −0.08 and a 0.6 st. dev. after recalculation (Fig. 3b). Finally, we implemented the new scaling equation to ElarmS and verified that we obtained the same results in a rerun of the playbacks.

The disparity of the coefficients of the scaling equations (2) and (4) might be explained by the differences in the geologic structures of the Israel region and the California/Japanese regions that were used to derive the two empirical relations. Similar differences leading to an underestimation of peak ground acceleration have also been shown by Campbell and Bozorgnia (2006), who compared ground-motion prediction equations (GMPEs) derived from global data and GMPEs, which were devised for Europe and the Middle East by Ambraseys et al. (2005). Hereafter, we use the local magnitude scaling relation of Sadeh et al. (2014) instead of the global relations derived by Kuyuk and Allen (2013a).

Figure 2. Playback magnitudes. ElarmS versus GII catalog magnitudes. Event magnitudes are marked as open circles, stations magnitudes used for calculating the events magnitudes are marked as dots. (Top) Using \(P_d\)-magnitude relations after Kuyuk and Allen (2013a) and (bottom) using \(P_d\)-magnitude relations after Sadeh et al. (2014).

Figure 3. Histograms of errors for historical playback data: (a) magnitudes errors using Kuyuk and Allen (2013a) scaling, (b) magnitudes errors using Sadeh et al. (2014) scaling, (c) locations errors, and (d) origin-time errors.
to pack the measurements and send the packet, plus the travel time of the packet, through a telemetry system (e.g., radio, internet, satellite, etc.). Delays are defined here as the difference between the time-stamp of a sample within the packet and the arrival time of that same sample to the buffer. Delays encompass the time between the sample of interest and the end of the data packet, plus the latency. For the ShakeAlert system in California, the latency for most of the networks is less than 2 s (less than 1 s for most of Berkeley [BK] and southern California [CI] stations) and most of the packet sizes are 1 s.

GI station latencies were measured over several hours on 1 May 2015. Though the ISN is not optimized for rapid RT acquisition, latencies for data arrivals are mostly below 4 s (mean latency 2.9 s). Additional stations, acquired from ISN and GE, have longer mean latencies of 8.4 s and 3.5 s, respectively (Fig. 4). However, despite the reasonable latencies, waveform packet sizes span 1–9 s (6 s on average) for the ISN stations. These long packets will delay trigger detection by half the packet length on average. For example, if the trigger is within the first samples of a 9 s packet, this would lead to a potential alert delay of ~10 s compared with the alert time for a 1 s packet waveform. This is because the full packet needs to be created and sent before the trigger is identified by the ElarmS triggering module.

Real-Time Results

During the course of our analysis period (1 May to 31 July 2015), 76 alerts were issued by ElarmS (events magnitude threshold is $M_w > 3.0$). The GI catalog contains seven events of $M_w > 3.0$ for this period with locations between 33.5° to 36.5° longitude and 28.7° to 34.7° latitude. Comparing the ElarmS alerts with the catalog, based on the space–time proximity of the ElarmS and catalog locations and origin time, matched 75 events (including underestimated catalog events of $M_p < 3.0$), 2 missed events and 1 false event. The missed events were $M_D$ 3.5 and 4.3 events in the Mediterranean Sea ($R > 140$ km) and the ElarmS teleseismic test suppressed these alerts. The false event ($M_w$ 3.4) was a result of misassociation of triggers from an $M_w$ 4.9 event in south Turkey ($R > 400$ km) similar to what we observed previously in the playback false alert. We found that ElarmS issued alerts based on an overestimation of 70 events (Fig. 5).

The GI analysis team categorized nearly all of the events for which ElarmS overestimated the magnitudes as explosions (67 out of 70 events), based on the frequency content of the $P$ and $S$ waves, their location and other considerations. The overestimation of magnitudes of quarry blasts could be a consequence of the different physical mechanisms behind explosions and natural earthquakes. Larger $P$-wave amplitudes are found for explosions at the frequencies used by ElarmS (demonstrated by Baumgardt and Ziegler, 1988, and Baumgardt and Young, 1990). Because ElarmS uses the maximum displacement of the very first arrivals for its calculations, this effect may explain the overestimation of the explosion magnitudes. Figure 6 shows an example of

---

**Figure 4.** Latencies histogram for the Israeli seismic network stations (IS, blue), the Royal Jordanian Seismic Observatory stations (JS, green), and the Geophone Network stations (GE, red) used by the GII.

The missed event, flagged as teleseismic by ElarmS, is not actually a teleseismic event. Indeed, the event average $P_d$ and $T_P$ values did not meet the threshold needed to be considered a local event by ElarmS (see equation 2 in Kuyuk, Allen, et al., 2014), but they were very close to this threshold. The simple linear threshold used to differentiate teleseismic events from local events might need further adjustments for the Israeli region.

The false alert was based on triggers that should have been associated with event 201308080824 (Table 1). ElarmS associates new triggers to an event if the measured travel time falls within a certain time window, defined by a calculated $P$-wave travel time plus or minus a few seconds. This misassociation suggests that the time-window calculation should be adjusted to the local velocity structure.

Real-Time Performance

ElarmS has been running at the GII on a testing server since November 2014. For our current work, we modified the ElarmS code to (a) enable RTP and ATP, (b) to produce more detailed and coherent log files suitable for analysis, (c) to fix some minor bugs, and (d) to adjust for the new geographical location. Because of these changes and because of the inconsistency of older log files, RT results are only available since May 2015. Further analysis of the RT performance will be examined in the future. Data analysis presented here is from 1 May to 31 July 2015.

ISN Latencies and Delays

Latency is defined here as the difference between the time-stamp for the final measurement sample in a waveform packet and the arrival time of the same packet at the ElarmS data buffer. This includes the time needed for the data logger

---

BSSA Early Edition
Real-Time Performance for two \( M_w > 4.0 \) Events

The 27 June 2015 \( M_w 5.5 \) Nuweiba Event. The most significant event during the analysis period was the 27 June 2015 \( M_w 5.5 \) Nuweiba event (28.877° latitude; 34.707° longitude; Fig. 7a). This earthquake was felt all across Israel, but only caused negligible damage. However, the epicenter for this event was in proximity to the 22 November 1995 \( M_w 7.2 \) Nuweiba event (Baer et al., 2008, and references therein), which resulted in severe damage to the nearest (≈80 km) Israeli city of Eilat. Figure 8a shows a timeline for the event. Catalog origin time for the \( M_w 5.5 \) Nuweiba event was 15:33:59.568 and the \( S \)-wave arrival time (representing the beginning of shaking) at Eilat (EIL station) was 15:34:27.190, 27.622 s after origin time. ElarmS successfully issued an early warning alert for this event at 15:34:26.330, 0.86 s before the arrival of the \( S \) wave at Eilat. The long delay in issuing the alert was due to the network geometry and large latencies of the yet-to-be optimized ISN stations. The arrival time for the \( P \) wave at the fourth station (HRFI, located 132 km from the epicenter) was 15:34:22.078 (22.51 s after origin time), and the waveform did not arrive at the ElarmS buffer until 15:34:26.267, which equates to an additional 4.189 s delay. The alert was issued only 63 ms later, so it is the scarcity of stations in the epicentral region and the packetization and delivery of data to the server that are responsible for the majority of the delay. Mean latency at HRFI was 2.7 s, but the packet size was 8.1 s (with the trigger 7.325 s from the start of the packet). We replayed the event in RTP mode, with an average latency of 3 s for all stations, and found that reducing packet sizes would have added ~1 s to the early warning time, and thus, issuing an alert 1.9 s before the arrival of the \( S \) waves to the city of Eilat. This relatively small improvement is doable due to the position of the trigger in the packet. Because the trigger occurred in the last second of the packet, the alert time was increased only by 1 s. In a scenario where the trigger occurred at the beginning of the packet, the alert time would have significantly improved. Reducing latencies from 3 to 1 s would result in an alert ~3.5 s before the arrival of the \( S \) wave.
Figure 7. Location map for two RT events. (a) The 27 June 2015 $M_D$ 5.5 Nuweiba event. (b) The 30 July 2015 $M_D$ 4.4 Dead Sea event. Catalog location is marked as dark star. ElarmS location is marked as a circle. Active stations used for first location are marked as filled triangles. Inactive stations are marked by open triangles. The location of the 22 November 1995 $M_w$ 7.2 Nuweiba event is marked as open star. Moment tensor solution is after Baer et al. (2008).

Figure 8. Timeline scheme for the RT processing of (a) the 27 June 2015 $M_D$ 5.5 Nuweiba event and (b) the 30 July 2015 $M_D$ 4.4 Dead Sea event. $P_x$, $P$-wave arrival time at a station, in which $x$ is the station trigger order of arrival. $S$, $S$-wave arrival at a location.
This event is also a good example of the potential usefulness of adjacent seismic networks, such as the JSN in Jordan. Two of the JSN BB stations, AQBJ and DRHI, are located at 100 and 60 km from the epicenter, but were not available for processing. If JSN data were available, alert times would have been increased to a maximum of ∼10.5 s, depending on data delays.

The 30 July 2015 $M_D$ 4.4 Dead Sea Event. An $M_D$ 4.4 event, felt in many areas across Israel, occurred on 30 July 2015 at 02:39:05.833 (GII catalog origin time), 48 km from Jerusalem at the center of the Dead Sea (31.403° latitude; 35.471° longitude; Fig. 7b). Assuming S-wave velocity of 3 km/s, the S-wave arrival time at Jerusalem is estimated at 02:39:21.833 (16 s after the origin time). Figure 8b shows a timeline for the event. The ElarmS first alert was issued at 02:39:19.150 (13.317 s after the origin time), giving ∼2.6 s of early warning for Jerusalem. The first magnitude estimation was $M_w$ 4.47, with a location error of 1.57 km and an origin-time error of 0.68 s. The alert was issued only 63 ms after the $P$ wave arrived at AMAZ; the fourth triggered station needed to issue an alert. The AMAZ packet size was 3.31 s (trigger at 2.05 s from the packet start), and the packet latency was 3.1 s, leading to a data delay of 4.36 s. Reducing the latency and packet size to 1 s could have added about 2.36 s to the early warning time, giving a total of ∼5 s to Jerusalem before the onset of shaking.

Earthquake Simulations

We use a set of four large earthquake scenarios (Table 2) to test ElarmS. Seismic waveforms were simulated for ISN stations using AXITRA (Coutant, 1990). AXITRA is a numerical code that, based on the reflectivity method (Miller, 1985; Kennett, 2009), evaluates Green’s functions for laterally homogeneous elastic media and approximates a full wave-train at each station (receiver) by convolving a given source function with the computed Green’s functions. Additional technical details and results of the simulation are described by Pinsky (2014). The simulated waveforms, originally provided as individual Seismic Analysis Code files for each channel, with counts representing velocity measurements, were streamed to ElarmS in ATP mode. The ElarmS results are summarized in Table 3. All four simulated events were detected and alerted upon, with no false alerts.

Magnitude errors for the simulated data are less than 0.3 magnitude units, origin-time errors are ∼0.3 s and location errors are less than 10 km, except for the $M_w$ 7.8 (event 4), originally located south of the Greek Karpathos Dodecanese Island. The poor solution of this event is due to its distance from the seismic network and the fact that ElarmS locates events on a grid limited to 200 km from the stations (Kuyuk, Allen, et al., 2014). Thus, ElarmS located the event ∼200 km from the triggered stations, at the edge of its search grid. Although the simulations were computed using the GII velocity model (Feigin and Shapira, 1994), the default ElarmS velocity model was used for processing, suggesting that the velocity model has little impact on the results. This first successful attempt at using ElarmS for processing synthetic data demonstrates the potential of using synthetic data for testing and evaluating the ElarmS system in different scenarios, both for the current Israeli network and for other networks.

### Table 2

<table>
<thead>
<tr>
<th>Number</th>
<th>Origin Time (hh:mm:ss)</th>
<th>$M_w$</th>
<th>Latitude (°)</th>
<th>Longitude (°)</th>
<th>Moment</th>
<th>Strike (°)</th>
<th>Dip (°)</th>
<th>Rake (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09:05:00</td>
<td>5.1</td>
<td>31.872</td>
<td>35.497</td>
<td>4.5 x 10$^{10}$</td>
<td>88</td>
<td>72</td>
<td>−142</td>
</tr>
<tr>
<td>2</td>
<td>09:34:50</td>
<td>5.8</td>
<td>32.117</td>
<td>35.557</td>
<td>5.0 x 10$^{10}$</td>
<td>190</td>
<td>70</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>10:17:05</td>
<td>6.9</td>
<td>31.983</td>
<td>35.498</td>
<td>2.2 x 10$^{10}$</td>
<td>202</td>
<td>77</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>12:17:05</td>
<td>7.8</td>
<td>35.2</td>
<td>27.5</td>
<td>5.0 x 10$^{10}$</td>
<td>50</td>
<td>30</td>
<td>25</td>
</tr>
</tbody>
</table>

Depth fixed at 10 km.

### Table 3

<table>
<thead>
<tr>
<th>Number</th>
<th>Origin Time (hh:mm:ss, ss)</th>
<th>$M_w$</th>
<th>Latitude (°)</th>
<th>Longitude (°)</th>
<th>Origin Time Error (s)</th>
<th>$M_w$ Error</th>
<th>Local Error (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09:05:00:32</td>
<td>5.7</td>
<td>31.8801</td>
<td>35.4643</td>
<td>0.32</td>
<td>0.6</td>
<td>3.22</td>
</tr>
<tr>
<td>2</td>
<td>09:34:49:74</td>
<td>5.6</td>
<td>32.1099</td>
<td>35.6387</td>
<td>−0.26</td>
<td>−0.2</td>
<td>7.75</td>
</tr>
<tr>
<td>3</td>
<td>10:17:05:23</td>
<td>6.7</td>
<td>31.988</td>
<td>35.5492</td>
<td>0.23</td>
<td>−0.2</td>
<td>5.87</td>
</tr>
<tr>
<td>4</td>
<td>12:18:12:80</td>
<td>7.5</td>
<td>33.2904</td>
<td>33.0402</td>
<td>−67.8*</td>
<td>−0.3</td>
<td>686.81*</td>
</tr>
</tbody>
</table>

Depth fixed at 8 km.

*Error is a result of the location search grid limits (see text for more details).
moderate and strong earthquakes, we analyze (1) historical data playbacks, (2) RT processing, and (3) simulated data. ElarmS has now been modified to be compatible with the network geometry in and around Israel. We find that it is necessary to use a regionally developed magnitude estimation equation to relate maximum displacement $P_0$ with magnitude. The magnitude relation previously derived from an independent dataset (Sadah et al., 2014) is an excellent fit between catalog magnitudes and ElarmS results for historical data with $3 < M_D \leq 5.3$ and for several available RT $M_D > 3$ events. We note that data used here do not include large magnitude earthquakes and therefore using the adjusted magnitude relations should be done with care.

We are satisfied that RT results exhibit good performance. Only two false alerts were reported ($M_w 3.4$ and 3.2) and no earthquakes of $M_D > 3$ were missed. However, the alert times are currently very short as the ISN is not optimized for RT warnings and suffers from large latencies and long data packets. ElarmS does detect the large number of explosive sources in the region and tends to overestimate their magnitude due to higher amplitude $P$ waves from these sources. However, events with the overestimate of magnitude are all $M_w < 3.5$ so applying a minimum magnitude threshold of $M_w 3.5$ to issue an alert resolves this issue. The development of additional filters to differentiate between blasts and natural earthquakes is underway and will be implemented in the future versions of ElarmS.

For the first time, we test the use of synthetic data to evaluate ElarmS performance. The results are encouraging because they show that events located less than $200$ km from the network edge were located in a range of less than $10$ km from the epicenter and $\sim0.3$ of origin time. Magnitude estimations were in the range of 0.3 magnitude units. For future planning, scenario-based testing will be needed. Further work is also required to generate more complete synthetic data incorporating 3D velocity models, noise, latencies, various rupture lengths and geometries, simultaneous events, preshocks or aftershocks, etc.

The methods and tools described in this work may be useful for implementing ElarmS in other regions, and similar efforts are being made in the US Pacific Northwest, Chile, Turkey, and South Korea. ElarmS has the potential to aid populations at risk to receive an early warning of seconds to tens of seconds to minutes before the arrival of destructive waves. The algorithm simplicity as well as the robustness and speed are highly suited to regions with limited seismic data availability. Nevertheless, efforts will be needed to verify and adjust ElarmS in any new region.

Data and Resources

Seismograms used in this study were collected by the Geophysical Institute of Israel (GII). Data can be obtained from GII at http://www.gii.co.il/ (last accessed August 2015). Moment tensor solution in Figure 7 acquired from U.S. Geological Survey at http://earthquake.usgs.gov/ (last accessed November 2015). Figures in this article were produced by Python 2D plotting module Matplotlib (Hunter, 2007) and ObsPy (Beyreuther et al., 2010). The tools created and used for analysis of data in this article are available at https://github.com/rannof (last accessed December 2015). Map used in the graphical user interface (GUI) tools is available at http://www.openstreetmap.org/ (last accessed December 2015)

Acknowledgments

The Geological Survey of Israel (GSI), at the Ministry of Energy and Water Resources, funded this research. The authors would like to thank the Geophysical Institute of Israel (GII) technical and research team for their cooperation and help with historical and simulated data.

References


Implementing the ElarmS Earthquake Early Warning Algorithm on the Israeli Seismic Network


Appendix

A set of tools were created to allow rapid deployment and analysis of the ElarmS system. These tools were developed using a combination of the Python scripting language and C programming language, combining the ease and rapid development of Python with the robustness and fast performance of C.

The main objectives of the tools were to (a) allow a real-time (RT) visual monitoring of ElarmS system modules and components, (b) enable RT and accelerated-time playbacks of historical data, and (c) review the log files and investigate ElarmS performance with respect to the Israeli Seismic Network (ISN) offline catalog.

The tools are now available online (see Data and Resources). A brief description can be found below.

ElViS

Elarms Visualization System (ElViS) is a graphical user interface (GUI) tool. It is connected to the ElarmS messaging system (ActiveMQ) and receives messages from the various modules and displays their content in a simple graphical way (Fig. A1). The tool follows ShakeAlert’s USERDISPLAY calculation methods, but is aimed at system administrators rather than end users. The tool is based on the Python module Matplotlib (Hunter, 2007). The current version has the following features:

- display of user predefined location (can be changed interactively);
- display of active/inactive stations locations and their current maximal acceleration/velocity/displacement;
- display of the trigger message logs from the EWP2 module;
• display of event message logs from the E2 module and decision module (DM);
• display an indication ActiveMQ connection;
• display event location and parameters;
• interactively send an event message to the ActiveMQ upon receiving an event alert from the E2 module or DM (can also simulate an alert of user-defined parameters);
• calculate and display the remaining warning time for the user location based on a simple distance to epicenter and a constant wave velocity;
• calculate and display the expected modified Mercalli intensity (Worden et al., 2012, their equation 3) at the user-defined location;
• display of ElarmS estimated magnitude; and
• calculate and display $P$- and $S$-wavefronts (based on a constant wave velocity).

SRTPB

SeedLink Real-Time Playback (SRTPB) is capable of reading waveform data from files supported by the Python module ObsPy (Beyreuther et al., 2010) and sending them to a SeedLink server, or saving them into a file. The data are repacked in packets of specified time intervals (typically 1 s) and ordered according to packet start time. Packets are then either sent in RT speed, or accelerated speed allowing a playback of historic, or simulated data. In addition, the tool can...
manipulate the data packet’s original time to account for predefined station latencies or to adjust time to the current time. The latter is most useful in conjunction with ElViS or USER-DISPLAY for evaluating alert times in a realistic scenario. For accelerated performance, data packets can be repacked and reordered and saved into a file to be sent directly to a SeedLink server at a later stage. This method allows replaying the same data multiple times more efficiently. Figure A2 shows a comparison of real-time playback (RTP) and accelerated-time playback (ATP) results. The differences are very small but not exactly the same due to the unrepeatable nature of ElarmS multiprocessing modules.

E2log2SC

E2log2SC converts E2 log files to Seiscomp3 event parameters xml files. This tool enables the analysis of the ElarmS results using the Seiscomp3 tools and importing the results to a Seiscomp3 database.

E2ReviewTool

E2ReviewTool is a GUI tool designed to read ElarmS log files and present the data as maps and tables, thus allowing analysis of the system performance. The current version has the following features:

- read ElarmS E2 module log files and a catalog list of events;
- associate ElarmS events and catalog events based on origin time and location;
- present an interactive map of catalog events and ElarmS events, with a color scheme for matched, missed, and false events (station locations are also available on the map);
- a table of all event information (origin time, location, magnitude, errors);
- a summary of performance (number of matched, missed, and false alerts);
- plots and histograms of magnitude estimation quality, compared with the catalog;
- a list of ElarmS solution evolution with time for each event;
- a list of ElarmS triggers used to solve each event, their properties, and relevant information; and
- filters events for certain magnitude, time, and space windows.

Figure A2. A comparison of real-time playback (RTP) and accelerated-time playback (ATP) results using the SeedLink Real-Time Playback (SRTPB) tool for 21 January 2015, $M_D$ 2.4 event selected randomly. (a) Magnitude estimation at each successive solution of the E2 module. RTP is marked by the solid line and ATP by the dashed line. ATP is set with a 0.1 s interval every 250 packets sent to the SeedLink server buffer to avoid overload of ElarmS buffers. (b) Difference in source parameters solutions between the RTP and ATP. Differences are very small but not exactly the same due to the unrepeatable nature of ElarmS multiprocessing modules.