VOLUME 5 ISSUE 1 FEBRUARY 2019

IJEG

International Journal of Engineering and Geosciences

e-ISSN 2548-0960

The International Journal of Engineering and Geosciences (IJEG)

The International Journal of Engineering and Geosciences (IJEG) is a tri-annually published journal. The journal includes a wide scope of information on scientific and technical advances in the geomatics sciences. The International Journal of Engineering and Geosciences aims to publish pure and applied research in geomatics engineering and technologies. IJEG is a double peer-reviewed (blind) OPEN ACCESS JOURNAL that publishes professional level research articles and subject reviews exclusively in English. It allows authors to submit articles online and track his or her progress via its web interface. All manuscripts will undergo a refereeing process; acceptance for publication is based on at least two positive reviews. The journal publishes research and review papers, professional communication, and technical notes. IJEG does not charge for any article submissions or for processing.

CORRESPONDENCE ADDRESS
Journal Contact: engineeringandgeoscience@gmail.com

CONTENTS

Volume 5 - Issue 1

ARTICLES

International Journal of Engineering and Geosciences (IJEG),
Vol; 5, Issue; 1, pp. 001-014, February, 2020, ISSN 2548-0960, Turkey,
DOI:10.26833/ijeg.577385

PERFORMANCE ASSESSMENT OF IGS COMBINED/JPL INDIVIDUAL RAPID AND ULTRA-RAPID PRODUCTS: CONSIDERATION OF PRECISE POINT POSITIONING TECHNIQUE

Sermet Ogutcu*****

Necmettin Erbakan University, Engineering Faculty, Department of Geomatic Engineering, Konya, Turkey
(sermetogutcu@konya.edu.tr); **ORCID 0000-0002-2680-1856**

*Corresponding Author, Received: 13/06/2019, Accepted: 25/07/2019

ABSTRACT: Satellite orbit and clock products are the key elements for precise point positioning (PPP). Contrary to the relative technique, errors in satellite orbit and clock directly lump to the station coordinates for PPP technique. Currently final, rapid and ultra-rapid (observed-half and predicted-half) satellite products have been made freely available over the internet mainly for Global Positioning Service (GPS) satellites. Final and rapid products are used for post-processing applications. For real-time and near real-time applications, ultra-rapid products with predicted and observed parts can be used. There are several analysis centers that provide the satellite orbit and clock products. In this paper, accuracy and precision of GPS rapid and ultra-rapid satellite orbit and clock products from two services, International Global Navigation Satellite Service (IGS) and Jet Propulsion Laboratory (JPL), were investigated in the position domain of PPP technique while the final products of these services were taken as the true value. Ten IGS stations around the world were chosen for PPP processes. 24-12-8-4-2 hours of non-overlapping arc-lengths of GPS observations in 31 consecutive days (DOY 1-31 of 2018) were processed for each station. The results confirm that the shorter the arc-lengths, the larger the relative error of rapid and ultra-rapid products due to the similar Gaussian distribution pattern of the orbit errors with respect to the final products. In terms of consistency between the products, Root-Mean-Square-Errors (RMSE) of final-rapid differences of IGS and JPL are at the millimeter level. Millimeter level accuracy can be obtained using rapid and ultra-rapid products for JPL whereas only rapid products of IGS maintain millimeter accuracy with respect to the final products.

Keywords: *Final orbits and clocks, GPS, IGS, JPL, Rapid and Ultra-Rapid products*

1. INTRODUCTION

With the developments of Global Positioning Service (GPS) satellite orbit and clock products, precise point positioning (PPP) could provide absolute positioning with millimetre (1-2 mm) and sub-centimetre level (4-6 mm) precision of daily solutions for horizontal and vertical components, respectively (Hayal and Sanli, 2016, Gao and Chen, 2004). PPP is gaining more popularity in the surveying community due to the simple and cost-effective approach. GPS users can obtain a homogeneous positioning accuracy around the world through PPP using a single receiver without depending on any fiducial stations (Zumberge et al., 1997).

Since the release of International Global Navigation Satellite Service (IGS) (Dow et al., 2009), broadcast ephemeris (Montenbruck et al., 2015), final, rapid and ultra-rapid products (Springer and Hugentobler, 2001, http://www.igs.org/products) have undergone dramatic development. IGS analysis centers receive and process tracking data from GNSS networks then these solutions are merged to produce combined IGS orbit/clock products.

Currently five types of GPS satellite orbit and clock products are available. The broadcast ephemeris and the IGS ultra-rapid (predicted-half) products mainly aim for real-time applications, while the IGS rapid and final products aim for post-processing applications. According to IGS official website (http://www.igs.org/products), the nominal accuracies of broadcast orbits and clocks are reported as ~1 m and ~5 ns, respectively (Tusat et al., 2018). Two types of ultra-rapid products are generated by IGS, one of which is observed-half with 3~9 h latency and the other is predicted-half without latency. The nominal accuracy of ultra-rapid observed-half and predicted-half orbits and clocks are reported as 3 cm and ~150 ps and ~5 cm and ~3 ns, respectively. The nominal accuracies of final and rapid orbits and clocks are reported as ~2.5 cm and ~75 ps, respectively.

The latency and accuracy differences between the products depend on the processing strategy and maximum used GNSS stations of the different Analyses Centers. The processing of final solutions includes minimal constraints to define the geodetic datum of the solutions. This means that only the sum of the rotations of a set of reference frame stations is constrained to be zero. For the rest, the reference frame of the solution is free in scale and origin. In the rapid and ultra-rapid solutions, the reference frame is determined by tightly constraining the 3D position of a set of selected reference GNSS stations. Due to the time constraints, rapid and ultra-rapid products typically use fewer stations compared to the final product.

The broadcast ephemeris and the IGS ultra-rapid (predicted-half) products are disseminating without latency. The latencies of IGS final, rapid and ultra-rapid (observed-half) products are 12-18 days, 17-41 hours and 3-9 hours, respectively. IGS ultra-rapid products update every six hours in order to decrease the divergence in orbit fitting throughout time. Each product is composed of 24 hr of observed orbits and 24 hr of predicted orbits.

The other commonly used GNSS service is the Jet Propulsion Laboratory (JPL) individual analysis centre. JPL has been providing GPS orbits and clocks since 1994 by processing data from the global network using GIPSY/OASIS scientific software (Bertiger et al., 2010). According to the JPL official website (https://gipsy-oasis.jpl.nasa.gov/index.php?page=data) three-dimensional Root-Mean-Square-Errors (RMSE) of ultra-rapid, rapid and final orbits and clocks are reported as 5, 3.5 and 2.5 cm, respectively. All products are provided in formats native to GIPSY/OASIS. The latency of JPL final, rapid and ultra-rapid products are reported as < 14 days, Next-Day (16:00 UTC) and < 2 hours, respectively. JPL ultra-rapid product differs from IGS. JPL ultra-rapid product does not include predicted part, it includes only observed part. JPL does not participate nor contribute to the IGS ultra-rapid product.

IGS and JPL sample intervals of orbit and clock products are the same for final and rapid products (15 min for orbit and 5 min for clock). Since IGS does not produce monitor station clocks in the ultra-rapid products, the orbit and clock sampling intervals of IGS ultra-rapid products are the same and equals 15 min. The orbit and clock sampling interval of JPL ultra-rapid product is 15 min and 5 min, respectively.

There is a significant delay between final and the other products and the trade-off between accuracy and time needed to be investigated properly. There are several studies that have been conducted in order to investigate the impact of using these products on the position domain.

Shi et al. (2017) investigated the impact of three types of real-time satellite products (broadcast, IGS ultra-rapid predicted-half and IGS-RTS CLK5 stream generated by CNES) on the position domain in relative technique while assuming the position from the final product is the true value. The results show that broadcast ephemeris provided <2 cm relative positioning error for baselines no longer than 216 km while CNES product could result in <5 mm relative positioning accuracy for baselines within 2982 km, slightly better than the predicted ultra-rapid product. Lu and Li (2011) analyzed the impact of IGS broadcast, ultra-rapid and rapid products on PPP. The results show that broadcast ephemeris reaches sub-meter level accuracy while rapid and ultra-rapid reaches cm level accuracy. Park and Yung (2014) analyzed the impact of ultra-rapid (observed-half), rapid and final IGS products on PPP and relative technique. They found that the largest standard deviation of ultra-rapid is 3.9 cm for PPP while the standard deviation of rapid and precise ephemeris is less than 1 cm. Yigit et al. (2017) assessed the PPP kinematic solutions based on both Natural Resources Canada (NrCAN) Ultra-Rapid products (commonly referred to as EMU) and IGS-Final products for detecting vertical dynamic oscillation. The results show that there is no significant difference between the IGS-Final and EMU-Ultra-Rapid products in terms of capturing dynamic oscillation. For relative positioning technique, the maximum standard deviation of ultra-rapid, rapid and final was reported as 1.4, 1.5 and 1.4 cm, respectively. They did not report the baseline lengths for relative technique. Martin et al. (2011), computed the north, east and up differences between final-rapid and final-ultra rapid products for PPP positioning. The results show that rapid and ultra-rapid (observed-half) products are at cm level when compared with the solution obtained from final files. When ultra-rapid predicted-half products are used, the differences are slightly higher than the ultra-rapid observed-half but still maintain cm level accuracy.

The above-mentioned studies only analyzed daily observations using IGS products which does not represent the real scenario most of the time and they did not consider different arc-lengths and other products from different analysis services. A comprehensive accuracy investigation of rapid and ultra-rapid products (observed-half and predicted-half) for PPP position domain is virtually uncharted territory.

In this study, the accuracy of rapid and ultra-rapid products of IGS and JPL analysis centers was investigated for the position domain of PPP while their final products are taken as the true value.

2. PPP OBSERVATION MODEL

In general, ionosphere-free code and carrier phase observations are used for PPP in order to eliminate the first-order ionosphere effect. The equations can be written for phase and code observations as follows;

$$P_{IF,r}^{s}=\rho + c * (dt_r - dt^s) + d_{trop} + HD_{P,r,IF} - HD^{P,s,IF} + \epsilon_{P,IF} \quad (1)$$

$$\emptyset_{IF,r}^{S} = \rho + c * (dt_r - dt^s) + d_{trop} + \lambda_{IF} * N_{IF}^{s} + HD_{\emptyset,r,IF} - HD^{\emptyset,s,IF} + \epsilon_{\emptyset,\iota f} \quad (2)$$

where the superscript s represents satellite, the subscript r represents receiver, $P_{IF,r}^{s}$ and $\emptyset_{IF,r}^{S}$ are the ionosphere-free combination of code and phase observations, ρ is the geometric range in meters, c is the speed of light in meters per second, dt_r is the receiver clock offset in seconds, dt^s is the satellite clock offset in seconds, d_{trop} is the tropospheric delay in meters, λ_{IF} is the ionosphere-free wavelength, N_{IF} is the ionosphere-free phase initial ambiguity, $HD_{P,r,IF}$ and $HD_{\emptyset,r,IF}$ are the ionosphere-free receiver hardware delay, $HD^{P,s,IF}$ and $HD^{\emptyset,s,IF}$ are the ionosphere-free satellite hardware delay and $\epsilon_{P,IF}$ and $\epsilon_{\emptyset,\iota f}$ are the ionosphere-free code and phase measurement noise.

The ionosphere-free code and carrier phase observables can be written as;

$$P_{IF,r}^{s}= (f1^2 * P1^s - f2^2 * P2^s)/(f1^2 - f2^2) \quad (3)$$

$$\emptyset_{IF,r}^{s}= (f1^2 * \emptyset1^s - f2^2 * \emptyset2^s)/(f1^2 - f2^2) \quad (4)$$

where $f1$and $f2$ are two carrier-phase frequencies in Hertz and $P1$, $P2$, $\emptyset1$, $\emptyset1$ are the measured pseudorange and carrier phase in meters on two frequencies. Ionosphere-free wavelength and ionosphere-free ambiguity can be written as;

$$\lambda_{IF} = \frac{f1^2}{f1^2-f2^2} * \lambda_1 - \frac{f2^2}{f1^2-f2^2} * \lambda_2 \quad (5)$$

$$N_{IF}=\frac{f1^2*N_1}{f1^2-f2^2} - \frac{f2^2*N_2}{f1^2-f2^2} \quad (6)$$

where N_1 and N_2 are the phase initial integer ambiguities on two frequencies with respect to each observed satellite. As it is seen from Equation 6, ionosphere-free ambiguity can only be estimated as a float (real-value) unknown unless using bias information from the network. There are several sources of these biases for example, the non-integer uncalibrated hardware delays (UHD) originating in receivers and satellites (Geng et al., 2009) and wide-lane and dual-frequency phase bias information (Bertiger et al., 2010).

The estimated parameters include the three-dimensional coordinates of the receiver, receiver clock offset, tropospheric zenith wet delay and ionosphere-free ambiguities.

3. DATA PROCESSING

10 IGS stations distributed around the world were chosen to perform PPP (Figure 1). When selecting the IGS stations, data availability and different satellite visibility of the stations were considered.

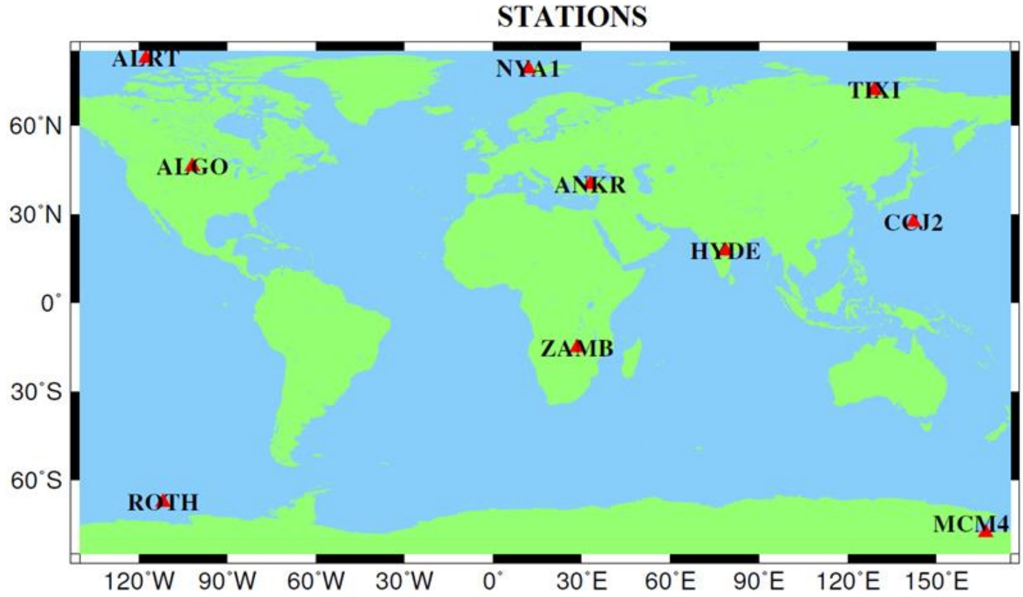


Figure 1. The distribution of the used IGS stations

31 consecutive days in the day of the year (DOY) 1-31 of 2018 were chosen to perform PPP. Each day's 24 h Receiver Independent Exchange (RINEX) data were mutually subdivided into 24-12-8-4-2 hours of non-overlapping sessions for each station. Seven different IGS/JPL products namely, final, rapid, ultra-rapid observed-half and ultra-rapid predicted-half from IGS and final, rapid and ultra-rapid from JPL were processed for each station and each non-overlapping session. In order to represent the real scenario for near-real time PPP applications, most current available parts of IGS ultra-rapid products with respect to the observation sessions were used in this study. For example, for 2h non-overlapping sessions include 00-04h and 04-10h, iguxxxxx_06 (09 UTC) and iguxxxxx_12 (15 UTC) ultra-rapid products were used. In this way, the jump in both the orbits and the clocks due to switching from one product to the next were avoided. Since JPL does not publicly provide the real-time products, JPL ultra-rapid product for real-time could not be processed in this study.

As a result, 150, 300, 450, 900 and 1800 PPP processing were conducted for 24-12-8-4-2 h subdivided RINEX data for each product, respectively. Epoch availability was checked for each daily RINEX file using in-house software. Threshold of epoch availability was set at 95% and the RINEX files below 95% availability were removed from the processing. In total, 6 daily RINEX files were not included for PPP.

GIPSY/OASIS 6.4 scientific software was used for PPP. GIPSY/OASIS is developed by NASA's Jet Propulsion Laboratory (JPL). Table 1 summarizes the processing parameters used in this study for PPP. The processing parameters were kept identical for each run except the used IGS/JPL products.

Table 1. Average DOP values of GNSS constellations

Adjustment model	Stochastic Kalman filter/smoother implemented as square root information filter with smoother
Elevation cut-off angle	7 degree
Weighting with elevation	Applied
Epoch interval	300 seconds
GNSS system	GPS
Ionospheric effect	Removed by L1, L2 linear combination
Phase initial ambiguity	Estimated as real value (float)
Data weight of linear combination phase and code	1 cm / 1 m
Antex file	IGS14.atx (Rebischung and Schmid, 2016)
A priori troposphere	GPT2 model (Lagler et al., 2013)
Zenith delay estimation	random walk 0.05 mm km/sqrt(sec)
Horizontal delay gradients estimation	random walk 0.005 mm/sqrt(sec)
Troposphere gradients	Computed
Code differential bias	Up to date DCB file
Reference frame	ITRF14
Solid earth tide	IERS 2010 Conventions
Pole tide	IERS 2010 Conventions
Earth Orientation Parameter	IERS 2010 Conventions
Ocean Tide Loading	IERS 2010 Conventions
Eclipse strategy	Eclipsing satellites were not used until satellites reach nominal attitude

IGS orbit and clock products were transformed to native GIPSY JPL products before processing. This transformation was performed using two scripts (igs2flinn.pl and sp3togoa.py) in GIPSY.

4. RESULTS AND ANALYSIS

RMSE of PPP processes using rapid and ultra-rapid products were computed while PPP results using IGS and JPL final products were taken as the true value with respect to each non-overlapping session. Estimated geocentric Earth Centered Earth Fixed (ECEF) coordinate differences were resolved to topocentric coordinates. Horizontal (2D) and vertical (Up) RMSE values of rapid and ultra-rapid products for the each session are shown in Figure 2-3.

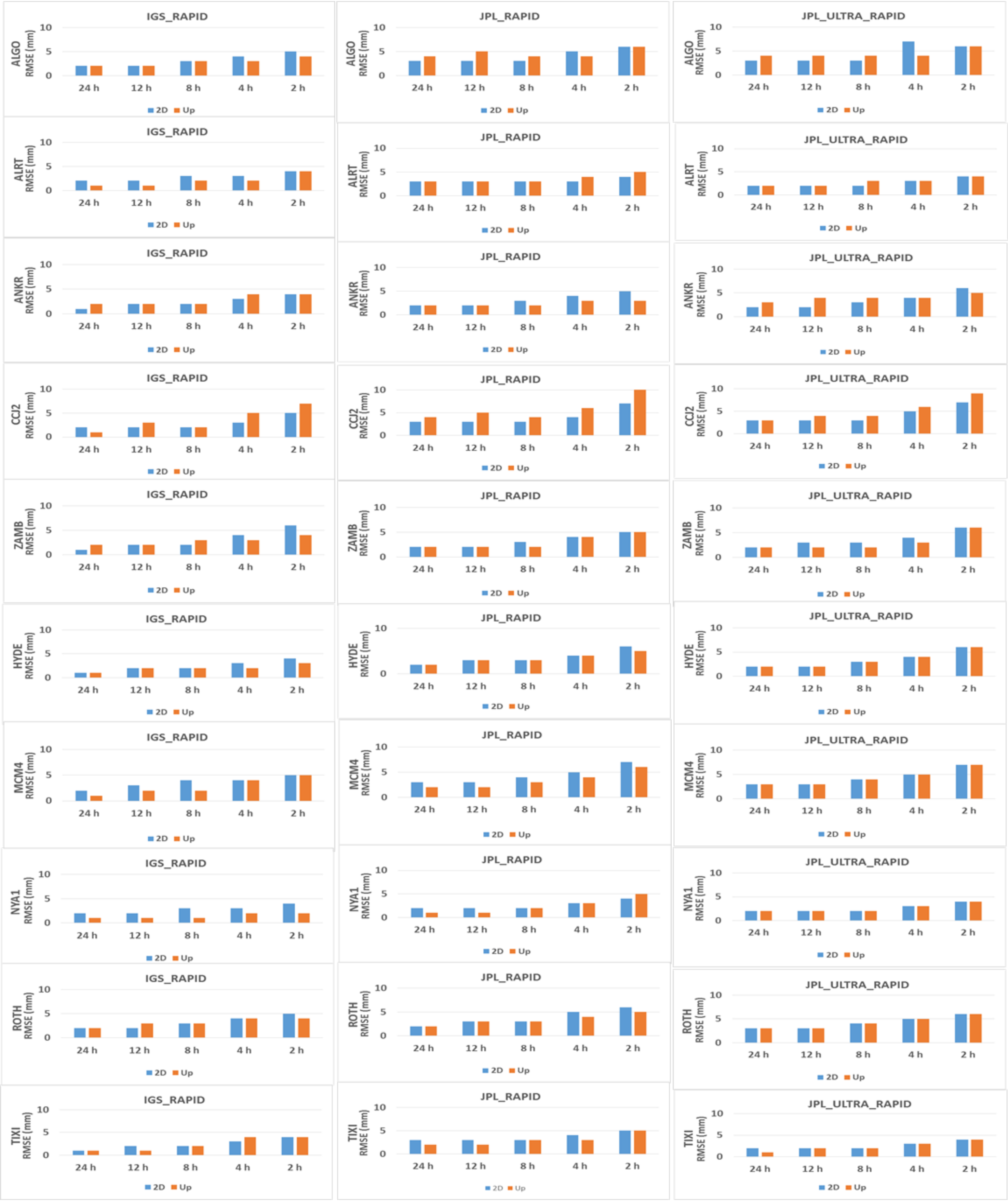


Figure 2. RMSE of PPP using JPL/IGS rapid and ultra-rapid products (Units: mm)

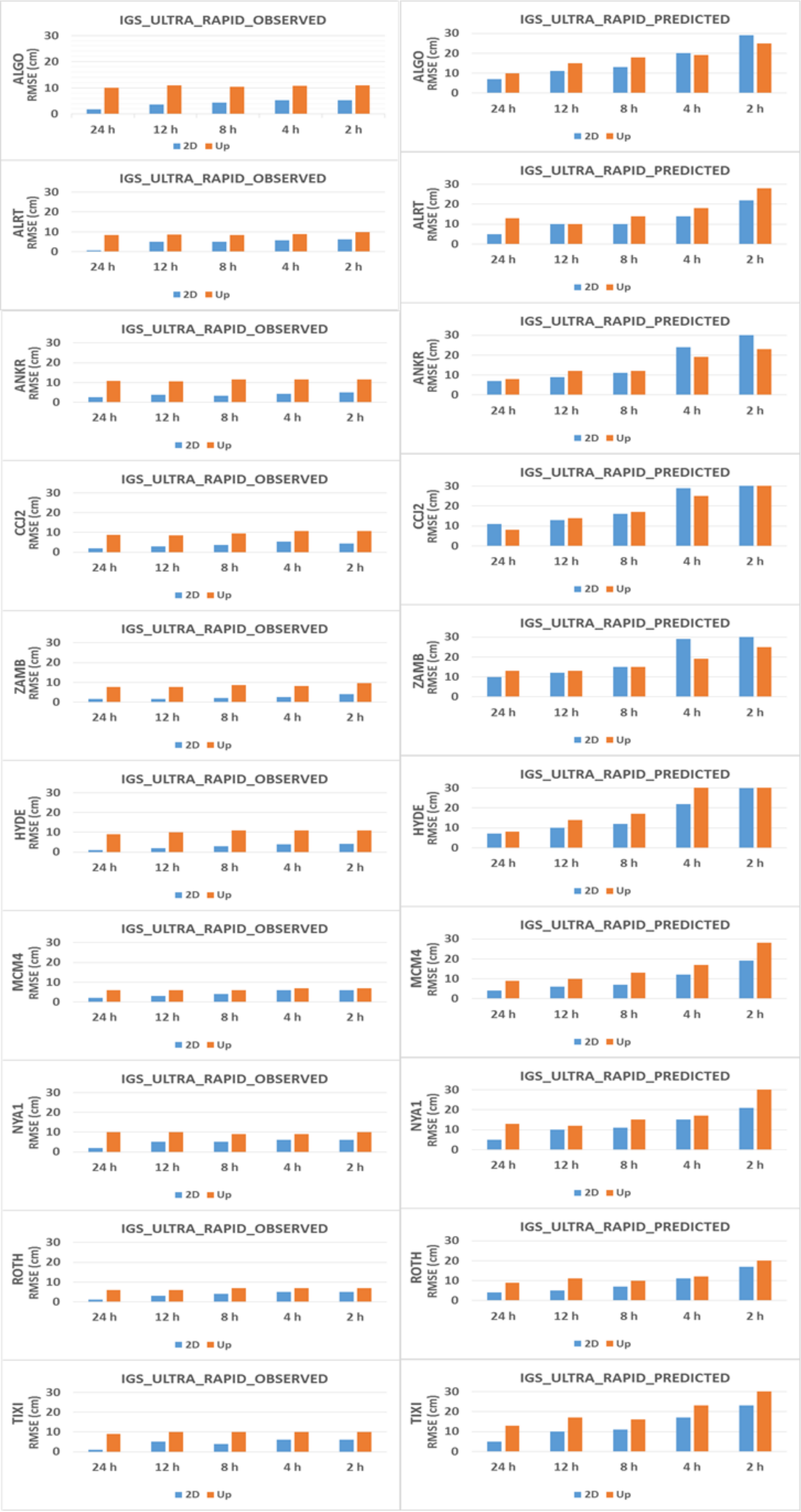


Figure 3. RMSE of PPP using IGS ultra-rapid product (Units: cm)

Table 2 shows the mean RMSE values of IGS stations for IGS rapid and JPL rapid/ultra-rapid products.

Table 2 Mean RMSE of PPP results using IGS rapid and JPL rapid/ultra-rapid products (Units: mm)

	IGS Rapid		**JPL Rapid**		**JPL Ultra-Rapid**	
Sessions	2D	Up	2D	Up	2D	Up
24	2	1	2	2	2	2
12	2	2	3	3	3	3
8	3	2	3	3	3	3
4	3	3	4	4	4	4
2	5	4	5	5	6	6

Table 3 shows the mean RMSE values of IGS stations for IGS ultra-rapid observed-half and predicted-half products.

Table 3 Mean RMSE of PPP results using IGS ultra-rapid products (Units: cm)

	IGS Ultra Rapid Observed-Half		**IGS Ultra Rapid Predicted-Half**	
Sessions	2D	Up	2D	Up
24	2	9	6	10
12	3	9	10	13
8	4	9	11	15
4	5	9	19	20
2	5	10	26	28

As it is seen from the RMSE values, PPP using IGS rapid, JPL rapid and JPL ultra-rapid products can maintain mm level of relative accuracy for horizontal and vertical components with respect to final product. JPL rapid and JPL ultra-rapid products produced nearly identical results. The JPL ultra-rapid product is much more useful for near real-time possible users (Senturk et al., 2017; Alcay et al., 2019) compared to the IGS ultra-rapid observed-half product in terms of relative accuracy. The IGS ultra-rapid observed-half product maintains cm level relative accuracy whereas the IGS ultra-rapid predicted-half reaches dm level relative accuracy especially for short sessions.

The results indicate that when observing sessions shorter than 24 h, the PPP differences between final and the other products are more prone to becoming high. It is believed that differences between orbit products follow the similar Gaussian distribution and this is mainly responsible for this phenomenon. Figure 4-5-6 show the distribution of ECEF differences within 24 h (DOY 001) between IGS final and rapid products for 01 (Block IIF), 05 (Block IIR-M) and 28 (Block IIR) PRN GPS satellites. A similar distribution has been observed for other GPS satellites and other products but are not given here due to limited space in this manuscript.

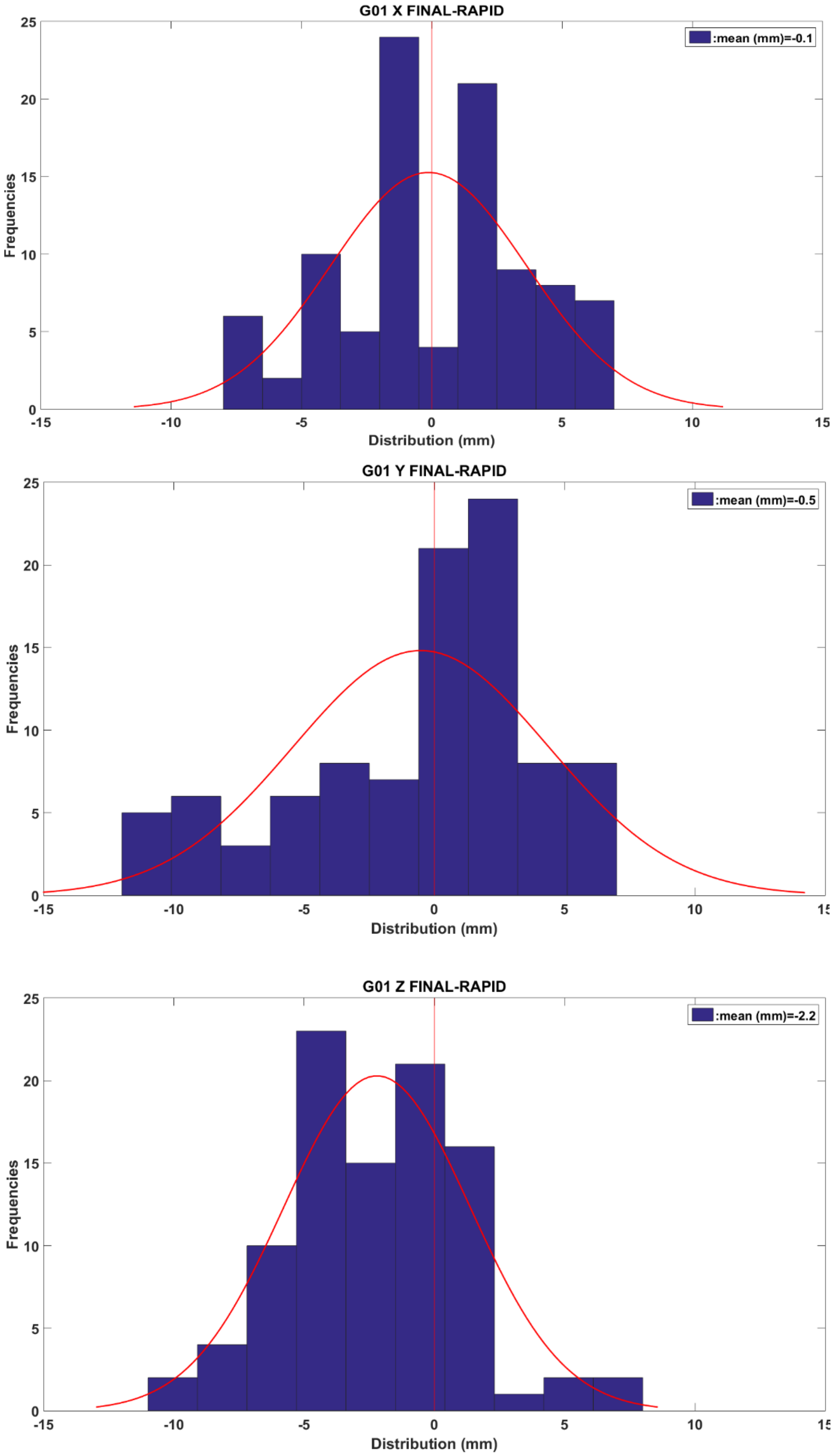


Figure 4. Distribution of ECEF differences between IGS final and rapid for 01 PRN GPS satellite

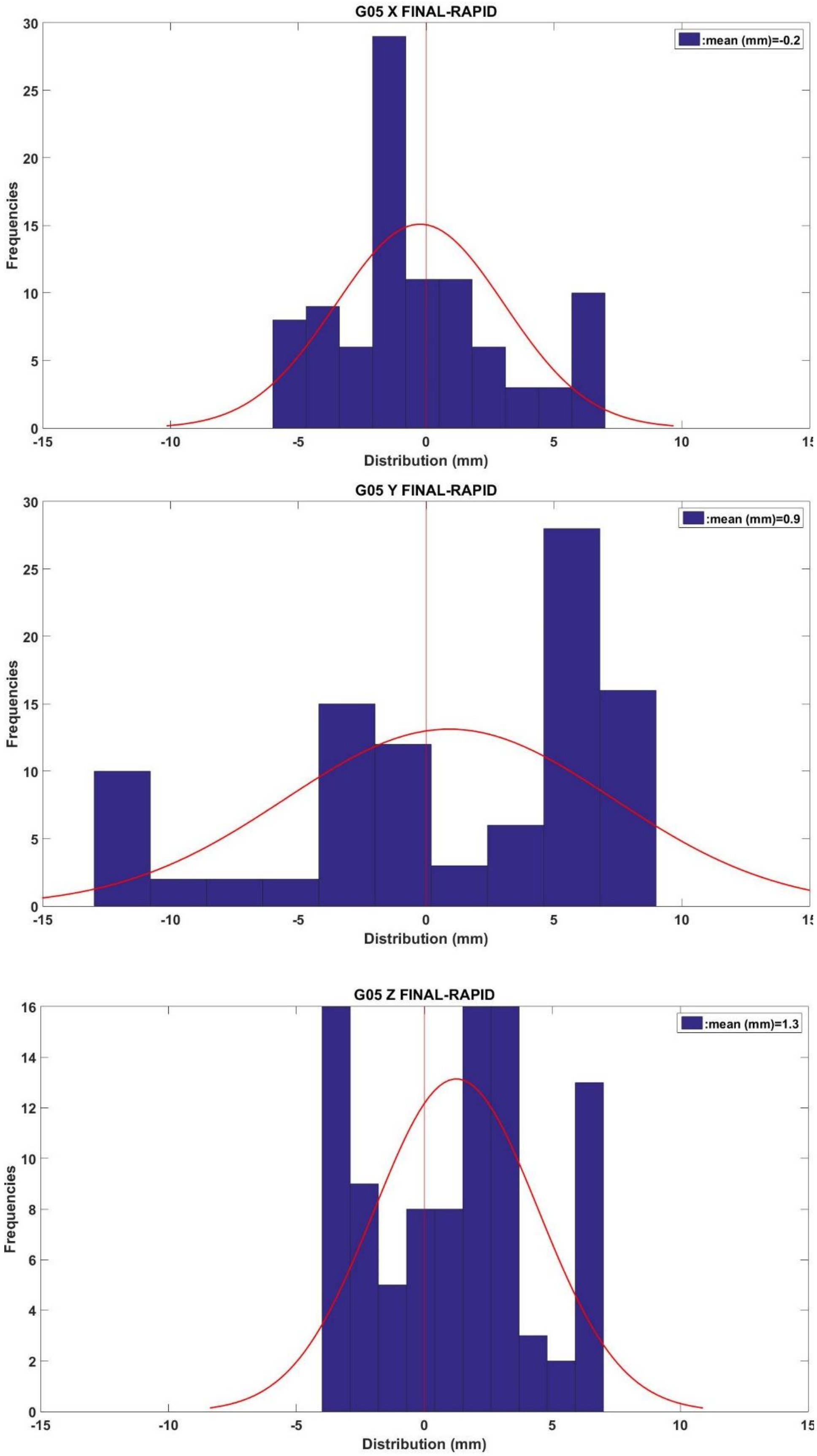


Figure 5. Distribution of ECEF differences between IGS final and rapid for 05 PRN GPS satellite

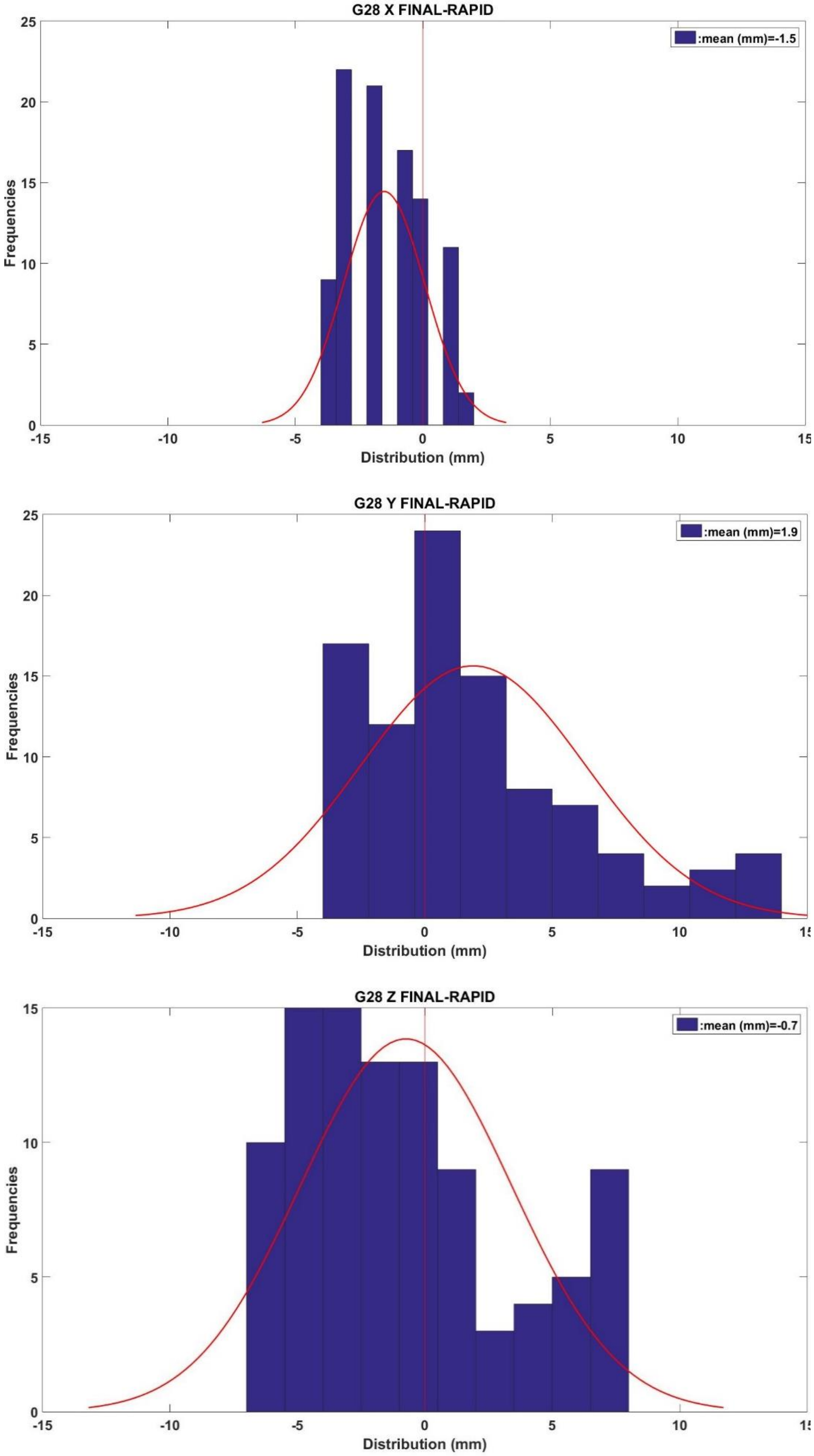


Figure 6. Distribution of ECEF differences between IGS final and rapid for 28 PRN GPS satellite

Table 4 Thresholds range of absolute PPP horizontal differences between final and rapid products for IGS and JPL

Thresholds of the absolute differences	IGS Rapid differences (units: mm)						JPL Rapid differences (units: mm)					
	0-1	1-2	2-3	3-5	5-10	10-15	0-1	1-2	2-3	3-5	5-10	10-15
24 h (%)	28	46	20	2	2		11	28	37	22	3	
12 h (%)	27	40	23	6	3		12	24	33	30	3	
8 h (%)	23	40	22	10	5		9	26	32	29	6	
4 h (%)	16	34	23	20	4		7	18	24	34	16	1
2 h (%)	12	25	25	24	11	2	5	14	18	30	27	5

Table 5 Thresholds range of absolute PPP vertical differences between final and rapid products for IGS and JPL

Thresholds of the absolute differences	IGS Rapid differences (units: mm)						JPL Rapid differences (units: mm)					
	0-1	1-2	2-3	3-5	5-10	10-15	0-1	1-2	2-3	3-5	5-10	10-15
24 h (%)	60	34	9	2	2		31	31	18	16	10	
12 h (%)	45	26	10	4	2		28	24	20	23	9	
8 h (%)	47	35	13	8	2		27	26	21	24	11	
4 h (%)	38	29	17	5	5		22	22	21	24	16	2
2 h (%)	31	26	18	17	11	2	16	16	15	25	26	4

Table 6 Thresholds range of absolute PPP horizontal differences between IGS final and IGS ultra-rapid products

Thresholds of the absolute differences	IGS Ultra-Rapid Observed-Half differences (units: cm)						IGS Ultra-Rapid Predicted-Half differences (units: cm)						
	0-1	1-2	2-3	3-5	5-10	10-15	0-5	5-10	10-20	20-25	25-30	30-50	50-60
24 h (%)	25	53	19	2			51	37	11				
12 h (%)	4	13	25	41	16		22	44	31	3			
8 h (%)	2	15	17	42	24		20	39	34	5	2		
4 h (%)	4	7	13	33	42		10	26	41	8	5		
2 h (%)	4	8	11	32	44	2	7	19	34	11	8	14	5

Table 7 Thresholds range of absolute PPP vertical differences between IGS final and IGS ultra-rapid products

Thresholds of the absolute differences	IGS Ultra-Rapid Observed-Half differences (units: cm)					IGS Ultra-Rapid Predicted-Half differences (units: cm)							
	0-5	5-10	10-15	15-20	20-25	0-5	5-10	10-20	20-25	25-30	30-40	40-50	50-60
24 h (%)	2	74	25			28	32	36	3				
12 h (%)	10	62	28	1		27	29	31	9	3			
8 h (%)	17	52	24	7		23	24	35	8	5	4	2	
4 h (%)	18	50	22	9		22	19	29	8	8	9	2	
2 h (%)	27	42	20	7	3	16	16	27	11	8	10	7	5

Tables 4-5 show the percentage of thresholds range of the absolute PPP horizontal and vertical differences between final and rapid products for IGS and JPL. The threshold range represents the minimum and maximum absolute differences between PPP results using final and rapid/ultra-rapid products. For example, 0-1 in Table 4 indicates the ratio of absolute differences that are bigger than or equal to 0 mm and less than or equal to 1 mm, among all the absolute differences with respect to each session. Tables 6-7 show the percentage of thresholds range of the absolute PPP horizontal and vertical differences between IGS final and IGS ultra-rapid products. The same thresholds range scale was not given for IGS ultra-rapid products due to the high difference between the products. Since the results of JPL rapid and ultra-rapid products are similar, JPL ultra-rapid percentage of thresholds range is omitted for convenient comparison.

The results of thresholds range indicate that PPP horizontal and vertical coordinate differences between final and the other products increases significantly as the duration of observing sessions decreases. The differences become more obvious especially for IGS ultra-rapid products. For example, maximum range was computed as 10-20 cm for 24 h IGS ultra-rapid predicted-half (11% of all residuals) whereas for 2 h observing session, maximum range reached was 50-60 cm (5% of all residuals) for the horizontal component.

Besides relative accuracy, precision of PPP using IGS/JPL products were also investigated for three IGS stations. 24-12-4 h sessions were chosen for comparison. Figure 7 shows the horizontal and vertical precision values of PPP results using final, rapid and ultra-rapid IGS/JPL products. Due to the similar results between JPL rapid and ultra-rapid products, the results of JPL ultra-rapid are omitted

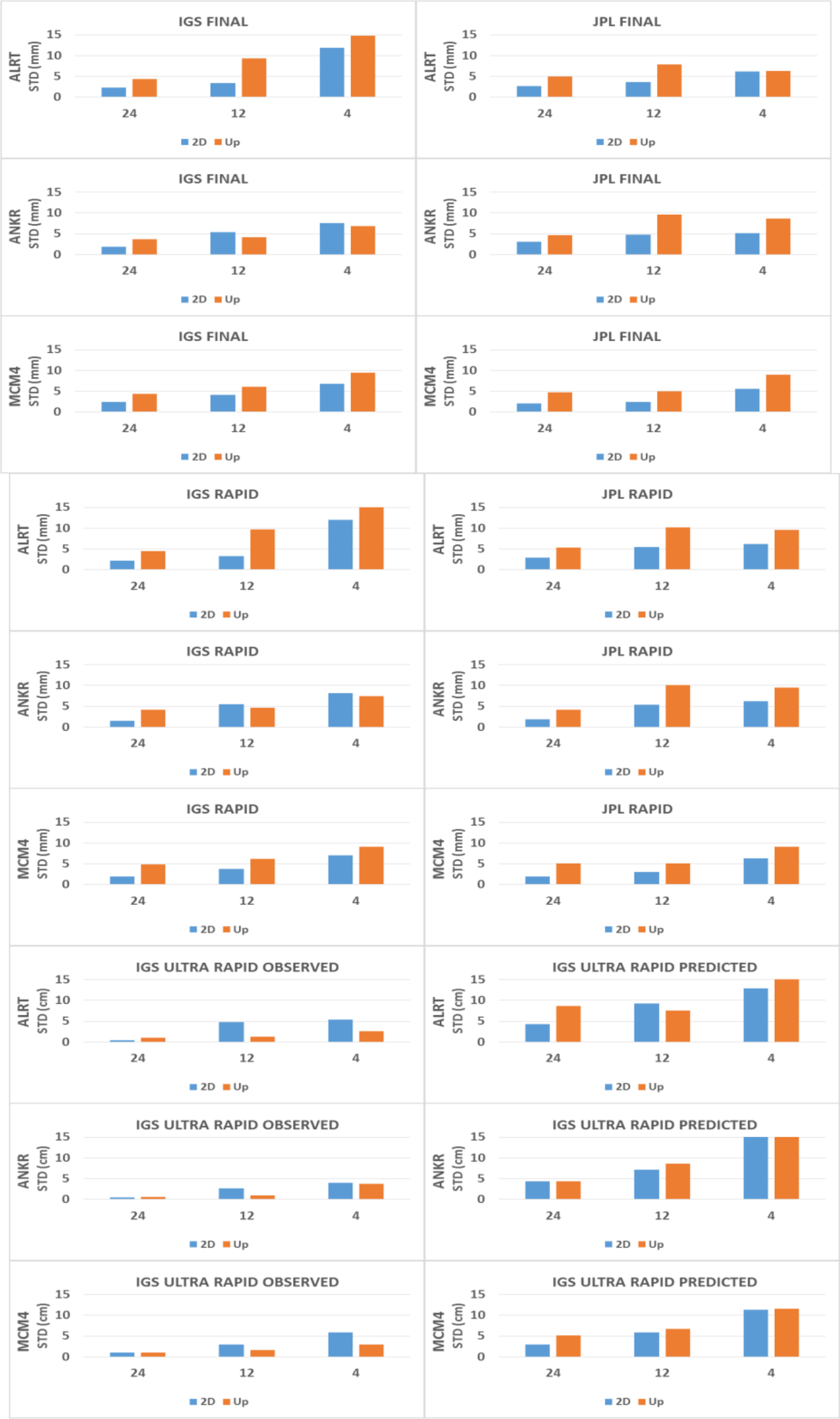


Figure 7. Precision values of PPP using IGS/JPL products

For an intuitive interpretation of the relative location of the satellites from different orbit products, local radial (e_{R_ref}), along-track (e_{A_ref}) and cross-track (e_{C_ref}) reference frame (local orbital frame) can be used. The corresponding unit vectors of this reference frame can be computed as follows (Montenbruck et al, 2002);

$$\boldsymbol{e_{R\_ref}} = \frac{\boldsymbol{r}}{|\boldsymbol{r}|} \qquad (7)$$

$$\boldsymbol{e_{C\_ref}} = \frac{\boldsymbol{r}\times\dot{\boldsymbol{r}}}{|\boldsymbol{r}\times\dot{\boldsymbol{r}}|} \qquad (8)$$

$$\boldsymbol{e_{A\_ref}} = e_{C\_ref} \times e_{R\_ref} \qquad (9)$$

$$[\boldsymbol{e_a}\ \boldsymbol{e_c}\ \boldsymbol{e_r}] = \left[e_{R\_ref} e_{C\_ref} e_{A\_ref}\right]^{-1} * \delta_{XYZ_{ECEF}} \qquad (10)$$

where r and $\dot{\boldsymbol{r}}$ are the satellite position and inertial velocity vector from the final ephemeris and $\boldsymbol{\delta_{XYZ_{ECEF}}}$ is the ECEF coordinate differences between the final and the other associated products.

Table 8 shows the average RMSE values of the radial, along-track and cross-track components of all GPS satellites for IGS/JPL rapid and ultra-rapid products relative to the final product for 31 consecutive days. Computations were performed using orbit overlap analysis software in GIPSY/OASIS.

Table 8 Average RMSE values of radial, along-track and cross-track components of all GPS satellites (Units: mm)

IGS Rapid			JPL Rapid			IGS Ultra-Rapid Observed-Half			JPL Ultra-Rapid			IGS Ultra-Rapid Predicted-Half		
R	A	C	R	A	C	R	A	C	R	A	C	R	A	C
4	6	6	9	11	10	7	12	11	10	14	12	10	53	56

Table 8 indicates that orbital errors are much higher in along-track and cross-track components compared to the radial component. It is understood from the results of the orbit errors that satellite clock errors affect the accuracy of PPP more significantly than the orbit errors. This situation is much more evident for the IGS ultra-rapid observed-half product.

5. CONCLUSION

PPP is a relatively new approach compared to differential positioning for mm to cm level accuracy in static positioning using a single geodetic-grade GNSS receiver and predetermined precise satellite orbit and clock products. Since the satellite orbit and clock errors directly affect the station coordinates for PPP technique, the choice of orbit and clock products play a significant role in PPP. In this paper, the relative accuracy of rapid and ultra-rapid orbit and clock products from two commonly used GNSS analysis centers, IGS combined and JPL individual, were investigated for PPP horizontal and vertical coordinates while their final products were taken as the true value. 24-12-8-4-2 hours of non-overlapping sessions of GPS observations in 31 consecutive days were processed using IGS/JPL final, rapid and ultra-rapid products for ten IGS stations distributed around the world.

The results show that the relative horizontal and vertical accuracy of PPP using rapid and ultra-rapid products decreases as the observation session becomes shorter. It is believed that since the same part of non-overlapping arc-lengths was compared between the final and the other products, distribution of the orbit errors (similar to normal distribution) is mainly responsible for this phenomenon.

RMSE of PPP horizontal and vertical coordinates indicate that mm-level horizontal and vertical relative accuracies with respect to PPP using the final product can be obtained using IGS/JPL rapid and JPL ultra-rapid products. RMSE of JPL rapid and ultra-rapid products are nearly identical and this is very beneficial for near-real time users who need mm-level accuracy. IGS ultra-rapid observed-half and predicted-half products can provide cm to dm-level relative accuracy with respect to the IGS final product. It should be emphasized that the most current available parts of IGS ultra-rapid products with respect to the observation sessions were used for PPP processes. The accuracy would be much lower if the out of date ultra-rapid products with respect to the observing sessions were used. In this study, 300 sec sampling interval was used for each process; thus, third-order polynomial interpolation was conducted only for IGS ultra-rapid clock products. If less than 300 sec sampling interval is used for the processes, clock interpolation error might have occurred for the other products. As a result, RMSE of PPP using IGS/JPL rapid and JPL ultra-rapid products may be increased. The difference between horizontal and vertical RMSE is much more evident for IGS ultra-rapid products. PPP relative vertical accuracy is significantly lower than relative horizontal accuracy for IGS ultra-rapid products while PPP relative horizontal and vertical accuracies are similar for the other products. Since IGS ultra-rapid clocks interval is 15 min, its relative accuracy is expected to be much lower than JPL ultra-rapid relative accuracy.

It should be emphasized that PPP processing model for orbit and clock between IGS and JPL is different. In this study, only the relative accuracy of rapid and ultra-rapid IGS/JPL products with respect to theirs final products were evaluated. Since the absolute accuracy of rapid and ultra-rapid products is not the scope in this paper, the difference in PPP processing strategy for orbit and clock products between IGS and JPL does not affect the overall result of relative accuracies.

The results of the threshold ranges show that nearly 100% of absolute horizontal and vertical differences are within 0-5 mm for IGS and JPL rapid/ultra-rapid products if 24 h observing sessions are used. The maximum difference reached 15 and 25 cm for 24 h IGS ultra-rapid observed-half and predicted-half products, respectively. As it is expected the differences are getting higher as observing sessions become shorter. For example, the maximum horizontal absolute difference is 20 cm and 60 cm for 24 and 2 h sessions, respectively, using IGS ultra-rapid predicted-half products.

Radial, along-track and cross-track components of rapid and ultra-rapid products indicate that relative satellite clock errors rather than the relative orbit errors are mainly responsible for the coordinate differences with respect to PPP using final products. It can also be concluded from the results that orbit errors are mainly lumped in the along-track and cross-track components for IGS/JPL rapid and ultra-rapid products.

The author believes that this study has contributed to the impact of rapid and ultra-rapid orbit and clock products on the PPP coordinates while taking into consideration the trade-off between the relative accuracy and time delay with respect to final product.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to NASA Jet Propulsion Laboratory (JPL) for providing GIPSY/OASIS scientific software.

REFERENCES

Alcay, S., Ogutcu, S., Kalayci, I and Yigit, C.O. (2019). Displacement monitoring performance of relative positioning and Precise Point Positioning (PPP) methods using simulation apparatus, Advances in Space Research 63, 5, 1697–1707.

Bertiger, W., Desai, S. D., Haines, B., Harvey, N., Moore, A. W., Owen, S., and Weiss, J. P. (2010). Single receiver phase ambiguity resolution with GPS data. Journal of Geodesy, 84(5), 327-337.

Dow, J. M., Neilan, R. E and Rizos, C. (2009). The international GNSS service in a changing landscape of global navigation satellite systems. Journal of geodesy, 83(3-4), 191-198.

Gao, Y and Chen, K., (2004). Performance analysis of precise point positioning using rea-time orbit and clock products. Positioning, 1(08), 0.

Geng, J., Teferle, F. N., Shi, C., Meng, X., Dodson, A. H and Liu, J. (2009). Ambiguity resolution in precise point positioning with hourly data. GPS solutions, 13(4), 263-270.

Hayal, A. G and Sanli, D. U (2016). Revisiting the role of observation session duration on precise point positioning accuracy using GIPSY/OASIS II Software. Boletim de Ciências Geodésicas, 22 (3), 405-419.

Lagler, K., Schindelegger, M., Böhm, J., Krásná, H., and Nilsson, T. (2013). GPT2: Empirical slant delay model for radio space geodetic techniques. Geophysical research letters, 40(6), 1069-1073.

Lu, F and Li, J (2011). Precise point positioning study to use different igs precise ephemeris. In Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference, 3, 592-595.

Martín, A., Anquela, A. B., Capilla, R and Berné, J. L (2010). PPP technique analysis based on time convergence, repeatability, IGS products, different software processing, and GPS+ GLONASS constellation. Journal of Surveying Engineering, 137(3), 99-108.

Montenbruck, O., Steigenberger, P and Hauschild, A. (2015). Broadcast versus precise ephemerides: a multi-GNSS perspective. GPS solutions, 19(2), 321-333.

Montenbruck, O., Ebinuma, T., Lightsey, E. G and Leung, S. (2002). A real-time kinematic GPS sensor for spacecraft relative navigation. Aerospace Science and Technology, 6(6), 435-449.

Rebischung, P and Schmid, R. (2016). IGS14/igs14. atx: a new framework for the IGS products. In AGU Fall Meeting 2016.

Park, J. K and Jung, K. Y. (2014). Accuracy Analysis of Influences by Satellite Ephemeris. Contemporary Engineering Sciences, 7(24), 1.

Şentürk, E and Erener, A. (2017). Determination Of Temporary Shelter Areas In Natural Disasters By Gis A Case Study For Gölcük/Turkey. International Journal of Engineering and Geosciences, 2(3), 84-90.

Shi, J., Wang, G., Han, X and Guo, J. (2017). Impacts of Satellite Orbit and Clock on Real-Time GPS Point and Relative Positioning. Sensors, 17(6), 1363.

Springer, T. A and Hugentobler, U. (2001). IGS ultra rapid products for (near-) real-time applications. Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy, 26(6-8), 623-628.

Tusat, E and Ozyuksel, F. (2018). Comparison of GPS satellite coordinates computed from broadcast and IGS final ephemerides. International Journal of Engineering and Geosciences, 3(1), 12-19.

Yigit, C. O and Gurlek, E. (2017). Experimental testing of high-rate GNSS precise point positioning (PPP) method for detecting dynamic vertical displacement response of engineering structures. Geomatics, Natural Hazards and Risk, 8(2), 893-904.

Zumberge, J. F., Heflin, M. B., Jefferson, D. C., Watkins, M. M and Webb, F. H. (1997). Precise point positioning for the efficient and robust analysis of GPS data from large networks. Journal of geophysical research: solid earth, 102(B3), 5005-5017.

International Journal of Engineering and Geosciences (IJEG),
Vol; 5, Issue; 1, pp. 015-025, February, 2020, ISSN 2548-0960, Turkey,
DOI: 10.26833/ijeg.587023

AN IOT ARCHITECTURE FOR FACILITATING INTEGRATION OF GEOINFORMATION

Ümit Işıkdağ [1]*

[1] Mimar Sinan Fine Arts University, Department of Informatics, Istanbul, Turkey
(umit.isikdag@msgsu.edu.tr); **ORCID 0000-0002-2660-0106**

*Corresponding Author, Received: 04/07/2019, Accepted: 09/09/2019

ABSTRACT: Background: GeoInformation, is very valuable for a range of fields ranging from location based services and navigation to smart cities and homes. On the other hand today many fields benefit from Internet of Things (IoT) implementations, where the machine-to-machine and machine-to-human transmission of GeoInformation frequently occurs. This transmission usually occurs in multi-source/multi-target and multi-platform IoT environments. **Problem Statement:** In many cases real-time GeoInformation stays in its own island of automation, and thus its real value cannot be uncovered. This happens mainly due to inefficiencies and problems that occur in the storage, sharing and exchange of real-time GeoInformation as a result of multi-source/multi-target and multi-platform nature of the IoT architectures. **Research Approach:** Integration appears as a critical paradigm which should be focused in order to store, manage and transfer of GeoInformation efficiently in these complex environments. In this context, the focus of the study was to test the applicability of different technologies and integration methods for acquisition, transmission and visualisation of multi-source GeoInformation through implementing an IoT Integration Testbed Architecture which is utilizing low-cost hardware (to acquire information), graph databases(to store information) and standard IoT protocols (to exchange information). The implementation explained in this paper covers acquisition of real time GeoInformation from a set of real and virtual sensors, storage of this GeoInformation in Graph Databases, exchange of information through two different communication models (request/response and publish/subscribe) based on standard IoT protocols, and visualization of information by web pages, web mapping services and using a GIS software. **Results:** The implementation results demonstrated a proof-of-concept on how multi-source GeoInformation acquired from different type of IoT nodes can be integrated, stored and visualised on different platforms by utilising a standard IoT communication paradigms and multiple communication models.

Keywords: *IoT; GeoInformation; Integration; Graph Database; Arduino; Raspberry Pi; REST*

1. INTRODUCTION

Today we are living in a more connected world, where seamless communication between humans and machines (M2H) and between machines themselves (M2M) is becoming an inevitable need. On the other hand, more and more information is becoming geographically referenced, and GeoInformation appears a key element of the today's information infrastructures, along with the increasing need for location based services. The use of geoinformation for enabling and facilitating smart city environments covers a wide range of fields including 3D modelling of terrain, remote sensing for building boundary extraction, urbanization studies, energy efficiency for buildings, tourism, and 3D modelling of cultural heritage (Goksel and Dogru,2019; Mulazimoglu and Basaraner,2019; Karsli et al,2018; Dogan and Yakar,2018; Erdem and Ince,2018; Yemenicioglu et al,2016). In smart cities, devices operating in many different platforms, specifically in smart city environments are providing georeferenced information and a considerable amount of this information is in real time. In fact, although this real time information is very valuable for different domains, most of this information stays in its own island of automation, and thus the real value of information cannot be uncovered. Information integration would be a key facilitator in this case, to bridge the isolated islands of GeoInformation in order to reveal the value information, and a well-established integration would make GeoInformation available for big data analytics, machine/deep learning and other AI related tasks. The Internet of Things (IoT) paradigm can help very much in acquisition and presentation of real-time GeoInformation.

IoT is a newly emerging model of communication which focuses on facilitation of information exchange between all different types of Things. The Internet of Things envisions a connected world of billions of smart things, devices, smartphones, smart cars, smart homes, and smart cities (Black and White, 2017). In an IoT environment Things may be real or virtual, moving or steady but things are active participants in the whole system. Things communicate with each other (things-to-things communication) and also communicate and interact with humans (things-to-human communication.) (Gaikwad et al.,2015) Although in the beginning it was thought that connecting everyday objects to the Internet would have no real purpose and use, but today connecting Things to the Internet enables boundless possibilities for smart environments, workspaces and cities. According to CISCO, the number of things connected to the internet exceeded the number of people on earth during 2008; furthermore, the company foresees that the number of internet-connected things will reach 50 billion in 2020 (Spalazzi et al.,2014) Connected Things provide many interesting opportunities if the information travelling between them, and between them and humans, can be handled, stored, managed and presented efficiently. On the other hand once information is georeferenced (i.e. represented in the form of GeoInformation), its' value rises exponentially. As IoT architectures are multi-source, multi-platform and multi-domain, integration is the critical paradigm which would unlock the full potential of the GeoInformation transferred, exchanged and shared in the IoT architectures.

In order to contribute to the integration of GeoInformation in IoT environments, the aim of the study was to test the applicability of various integration methods for acquisition, storage, transmission and visualisation of multi-source GeoInformation. This is accomplished through implementation of an IoT Integration Testbed Architecture by utilizing low-cost IoT hardware, graph databases, standard and commonly used IoT protocols, and multiple communication models.

The methodology of this research was composed of two steps. The first one was the review of literature on IoT concepts and standards, and on IoT and GeoInformation, and the second step was the development of the proof-of-concept IoT integration architecture, which included development of use cases and software components.

Following the review of basics of, IoT and the role of GeoInformation in IoT environments, the paper provides details of the developed Testbed architecture, by providing in-depth information about each software component developed to test the GeoInformation integration in an IoT architecture.

2. IOT & IOT PROTOCOL STACK

The core concept of the IoT is the integration of physical objects into a global information network, where information about these objects are represented in the Internet. This makes a direct interaction with and between the physical objects possible (Jung et al.,2012). An IoT architecture connects anything to the Internet and in an IoT environment Things can exchange information according to the agreed protocols in order to achieve intelligent identification, positioning, tracking and monitoring (Xiaoying and Huanyana, 2011) IoT is a complex interconnected system of sensors, actuators, smart devices, software applications that communicate together to accomplish a task (Elkhodr et al.,2015). It is believed that IoT architectures will make Internet even more immersive and pervasive, where concepts like Persistent Computing, Omnipresent Computing and Ambient Intelligence will become more popular. Furthermore, by enabling easy access and interaction with a wide variety of devices such as, home appliances, surveillance cameras, monitoring sensors, actuators, displays, vehicles, the IoT will foster the development of a number of applications that make use of the potentially enormous amount and variety of data generated by such objects to provide new services to citizens, companies, and public administrations (Zanella et al., 2014). Different domains ranging from home/industrial automation, to mobile healthcare and assistance and intelligent energy management will benefit from IoT architectures. Libelium (2019) classified the domains where IoT Technologies can facilitate as Smart Cities, Smart Environment, Smart Water, Smart Metering, Security & Emergency, Retail Sector, Logistics, Industrial Control, Smart Agriculture, Smart Animal Farming, Domotic and Home Automation, and e-Health. Futuristic transport applications such as robot taxis will benefit much from IoT architectures. The autonomous UAV's are also in the same group. The possibilities offered by IoT architectures can be considered as limitless (Spalazzi et al.,2014).

IoT information integration can be defined as combining data from different connected devices/sensors to produce more accurate, complete, and dependable information. This integration provides advantages as information acquired from multiple sensors helps, in understanding the surrounding environment more accurately and in seeing the big picture about certain events and environments, which is not possible using individual devices/sensors separately. In addition, the combination of information of multiple sensors produces new knowledge, which helps to build a context awareness model that helps to understand situational context.

IoT architectures makes use of both standard internet communication protocols and protocols developed for resource-constrained architectures. Protocols for communications in resource-constrained architectures are mainly developed by IETF. The literature presents many approaches for grouping and classifying the standards and protocols used in IoT architectures. IoT messaging standards and communication protocols (i.e. known as the IoT Protocol Stack) can be grouped into in three levels.

Level 1: The first level of the stack is composed of physical and datalink layers. The physical layer is the lowest layer in the communication protocols stack which deals with bit-level information transfer (Technopeida, 2019). The role of the datalink layer which resides above the physical layer is to encode/decode and organize the data bits prior to their transportation in form of frames in the network between the two nodes (Techtarget, 2019). The IoT paradigm benefits from several wireless technology standards at this level. These include; WiFi, low-energy networking standards such as LoRaWAN (Long Range Wide Area Network), IEEE 802.15.4, ZigBee, Bluetooth Low Energy (BLE) and finally high-energy LTE-A, (Long-term Evolution Advanced) which is (4G) cellular communication standard allowing data to be transferred in high-speed using a cellular network.

Level 2: The second level of the stack is composed of the network and the transport layers. The network layer provides paths for routing of data packets. The layer's role is finding the best path for the transfer of the data between the devices (Plixer,2019). The transport layer divides the data into segments where each data block has a port and sequence number, for ensuring the data reaches the target application in a correct order. This layer is also responsible for adjusting the speed of data transmission between the devices depending on the data sending and receiving capacities of the end-nodes. There are several standards and protocols used in IoT implementations at network layer including IP, 6LoWPAN, RPL and IPv6. At the transport layer the key standards are TCP and UDP. User Datagram Protocol (UDP) is the transport protocol used for achieving higher speeds for IoT applications.

Level 3: The third level of the IoT protocol stack is composed of session, presentation and application layers. The session layer opens, maintains and closes the sessions between multiple devices and determines which packets belongs to which data stream. Presentation is the layer where encryption and decryption (of critical) data occurs. The application layer is the layer where interaction with the user occurs. This layer aims to provide services to the end-user. Many well-known Internet protocols such as HTTP, FTP, SMTP, Telnet, DNS, and DHCP operate at this layer. There are several other protocols worth to mention in this level such as COAP and MQTT. Constrained Application Protocol (CoAP) is a protocol designed in IETF for resource-constrained devices based on RESTful architectural principles. Message Queue Telemetry Transport (MQTT) is another key protocol for IoT applications. MQTT is designed to operate on Low Power Networks. MQTT supports publish/subscribe (Pub/Sub) model of communication, while implementing a Message Broker. Many applications ranging from smart energy meters to monitoring applications makes use of MQTT today. The protocol fits well for M2M communication, when power consumption and bandwidth is a bottleneck. Also, MQTT is suitable for resource constrained devices that use unreliable or low bandwidth links. MQTT-SN was defined specifically for sensor networks and defines a UDP mapping of MQTT and adds broker support for indexing topic names (Al-Fuqaha et al.,2015; Salman and Jain,2017)

Another very commonly used architectural term in IoT architectures is the REST. REST stands for REpresentational State Transfer, and is neither a communication protocol nor a data encoding standard, but it is an architectural style for communication. REST has a key role and impact on IoT architectures. The REST architectural style utilises HTTP methods to enable and facilitate interaction in M2M and M2H communication. REST makes use of methods defined in RFC 2616 (HTTP) protocol, such as GET to request and acquire a representation of a web resource (i.e. in form of any type of data such as text, image), POST to create a web resource, PUT to change the representation of a web resource (i.e. update a resource), and DELETE to remove a web resource. The structure of a RESTful URI (Uniform Resource Identifier) should be straightforward, predictable, and easily understood (Rodriguez, 2018).

The IoT architecture explained in this paper implemented some of these messaging standards and communication protocols as:

- Wi-Fi for Level 1
- TCP/IP for Level2
- HTTP and MQTT for Level3
- RESTful architectural style is implemented for the integration of some components

Furthermore both (HTTP) publish/subscribe and (MQTT) request/response models of communication is implemented and tested within the architecture.

3. GEOINFORMATION AND IOT

Three main functions of IoT architectures are, acquiring information from the environment, conducting actions, enabling communication and interaction. Through the use of the sensors and actuators the real time information about a state and location of a Thing can be collected/stored, and also an action can be performed at a certain location. The georeferenced information is valuable for IoT architectures either when acquiring information about states of "Things" that is located at a certain region or when taking an action to change the state of "Thing" at a certain location. Furthermore, an action in an IoT environment may not only change a state of a "Thing" but also may change its location. In addition, an action can also change the form of a "Thing" or a "Space", and also the connectedness attributes (topologic relations) between the "Spaces". Thus, GeoInformation exchange is essential for IoT architectures to get informed about form, state, space and location, and topologic relation changes.

Internet of Things architectures can collect real time information regarding the state of many objects/utilities and environmental factors in the urban built environment, such as power grids, railways, bridges, tunnels, roads, wastewater systems, buildings, indoor facilities, street lights, parking spaces, trash containers, air quality, traffic, snow levels, water levels. Combining cloud computing, next-generation communication networks, and intelligent data mining technology, the Internet of Things helps in forming the smart city by making physical and informational resources integrate systematically (Zhou and Zhang, 2011) IoT technologies will change the geography research in 4 dimensions. Firstly, a new space will emerge, that extends beyond real geographical space through the real time information & interactions that are made possible with IoT. This new space will have impacts on Economic Geography, Social and Cultural Geography. The IoT will bring new human-land relationships through intelligent identification, positioning, tracking, monitoring of humans, furthermore real time information about land will be available seamlessly. The IoT will enable a more efficient logistics process lowering the distance/time barrier in movement of “Things” between spaces. Intelligent transportation refers to smarter use of transportation networks through utilisation of real time GeoInformation. Intelligent transportation would be very much facilitated by the IoT and will help humans to travel more quickly and safely. The IoT will help the formation of new behavioural geography where IoT will help in navigation and also travel decisions of humans, based on real time GeoInformation acquired about places. Furthermore, a person’s health condition, eating and buying habits can also be tracked by indoors and outdoors by making use of GeoInformation. Personalised advice can be provided to humans in these areas based on the GeoInformation gathered by the sensors (Ming and Ling, 2012). In terms of urban geography, the IoT will contribute to the gathering and diffusion of, real time information about humans, spaces, land and movement of humans and Things. IoT architectures will help in localisation and positioning in urban spaces. Today, the movement of “Things” and humans generate mass amount of data via connected wireless devices such as smartphones (Keng and Koo,2014). This GeoInformation can later be organised to determine the semantics of locations. Real-time GeoInformation gathered by IoT architectures can be used for managing emergency response operations, for example information related to temperature, CO density, light density can aid the evacuation crew in fire situations (Liu and Zhu, 2014).In order to facilitate the processes that require the use of GeoInformation, IoT architectures, standards and protocols provide unique opportunities for information integration. Recent work in the field include Kamilaris and Ostermann (2018) which provides a survey that investigates the opportunities of location-aware IoT, and examines the potential of geospatial analysis in this research area. Rieke et al. (2018) presented a selection of approaches developed in different research projects to overcome the gaps that retain certain geospatial applications from using real-time information. Laska et al (2018) presented an architecture for real-time processing of spatiotemporal IoT stream data. Pozzebon et al. (2018) proposed a wireless sensor network framework for real-time monitoring of height and volume Variations on sandy beaches and dunes.

4. OVERVIEW OF THE INFORMATION INTEGRATION ARCHITECTURE

The aim of this study was to test the applicability of various technologies and integration methods for acquisition, transmission and visualisation of GeoInformation by implementing an IoT Testbed architecture that is utilizing low-cost hardware and graph databases. As hardware related costs form the key cost item in the IoT implementations, first objective of the study focused on testing the applicability of the low-cost hardware. As many IoT architectures would have a need to store many relationships between the nodes, the second objective of the study was focused on testing the applicability of the graph databases in an IoT Architecture. Previous studies indicating the potential and uses of Graph Databases for GeoInformation storage and retrieval include Agoub et al. (2016) and Nguyen et al.(2017).As the data in a graph database is more connected than standard SQL database, the query response times of graph databases are much faster, especially when there is a lot of relationships between objects. In parallel with these aim and objectives of the research, the user requirements are identified and unified in a generic high-level single use-case scenario. The scenario is developed with a focus group of 15 experts from academia and industry working in the field of IoT and GeoInformation Management. All use cases identified during the user requirements definition stage are illustrated in Figure 1. The use case scenario developed involves several use cases which can be grouped under “interaction for generation of virtual sensor values”, “observation of sensor values through different interfaces” and “observation of sensor values through MQTT messages”. The generation of sensor values involves the generation of transient and persistent values. The observation of sensor values can be done using web interfaces and also utilizing GIS software. This group of use cases also involves background processes such as update of a graph database and generation of views. Another group of processes involves the observation of sensor values through the MQTT messages. The architecture developed based on these use cases consist of several hardware and software components which are explained in this section. Figure 2 illustrates the deployment diagram of the IoT Integration Testbed Architecture implemented in this study.

Hardware Components: The hardware components of the system include a Data-Tier Web Server (Processor/Controller: x64 architecture / OS: Windows), an Arduino Yun (Processor/Controller: Atmel ATmega32U4 + Atheros AR9331 / OS: LininoOS), an Arduino Uno (Processor/Controller: ATmega328P), a Raspberry Pi 2 (Processor/Controller: ARMv7 / OS: Raspbian), and a Middle-Tier Web Server (Processor/Controller: x64 architecture / OS: Windows). The boundary boxes shown in the Figure 2 illustrates the software components hosted by each hardware component in the architecture.

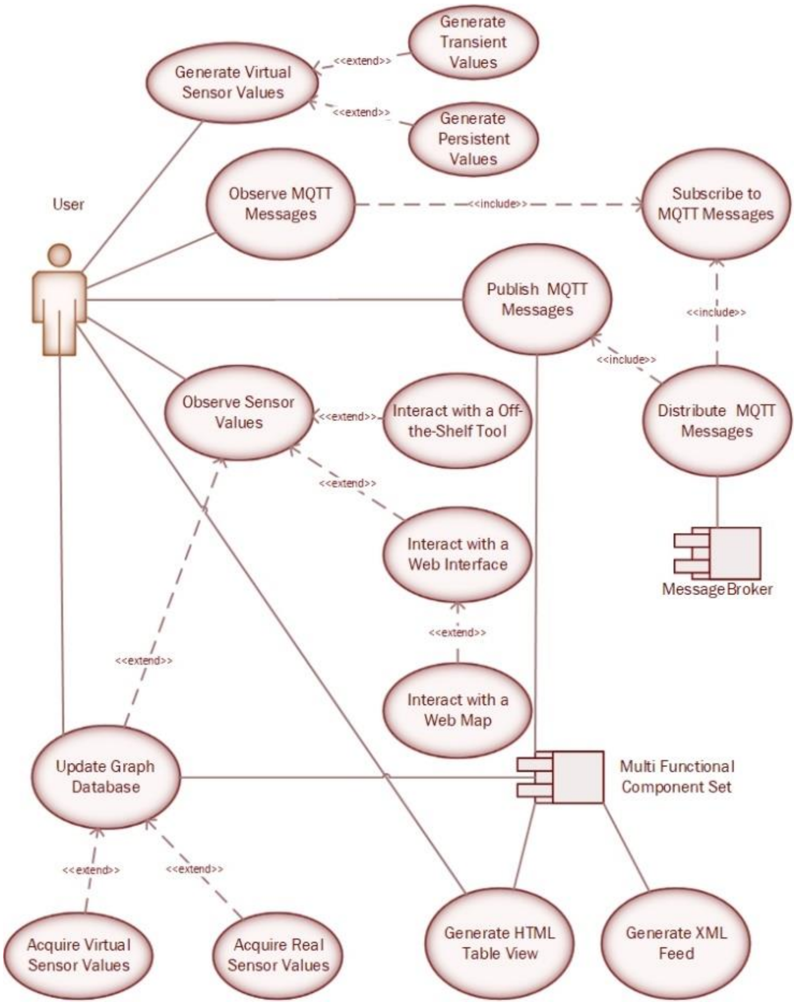


Figure.1 Use Case Diagram

Data-Tier Components: The data tier components of the system are, a Graph Database (Neo4J) server with Spatial Extension, XML and GeoRSS files residing in Data-Tier Web Server, IoT Node DB (which is an SQLite Database) and transient XML and HTTP files (generated on demand) residing in Arduino Yun , a transient XML file (generated on demand) residing in Arduino Uno.

Software Components (Developed): The software components of the system include Persistent Virtual Sensor (PVS) Generator, Graph Database Updater Component for Virtual Sensors, Cypher Query Translator (for Neo4J), Arduino Yun Local Database Updater & Data Viewer , Arduino Yun Graph Database Updater, Arduino Uno Simple XML Feed Generator, MQTT Message Publisher/Subscriber for Arduino Uno and Arduino Yun, MQTT Message Publisher/Consumer, Transient Virtual Sensor (TVS) Generator, GeoJSON Transformer, GeoRSS / GeoJSON Visualizers, Table View Generator, Statistical Data Generator.

Software Components (Utilized/Used): The software components that are utilised for the implemented architecture include an Apache Web Server on Data-Tier Web Server, NodeJS on Middle-Tier Web Server, uHTTPd Web Server on Arduino Yun, Mosquitto MQTT Server on Raspberry Pi 2, Arduino Client Library for MQTT (on Arduino Uno and Arduino Yun), QGIS Software on a client PC.

Output Files: The implemented architecture provides outputs in form of HTML and several other output file formats including XML, GeoRSS and GeoJSON. XML is a general purpose markup language, developed with the purpose of facilitating the exchange of data across different systems. As XML does not offer predefined tags, the structure (schema) and tags of XML files that are generated in this architecture were defined by the author. GeoRSS is a lightweight approach which extends RSS feeds, the standard provides an interoperable way for encoding geolocation to enable the geo-tagging of feeds. GeoJSON is a JSON (JavaScript Object Notation) compliant encoding of GeoInformation. RFC 7946 is the current standard that provides current GeoJSON specification. GeoJSON supports the following geometry types including Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon.

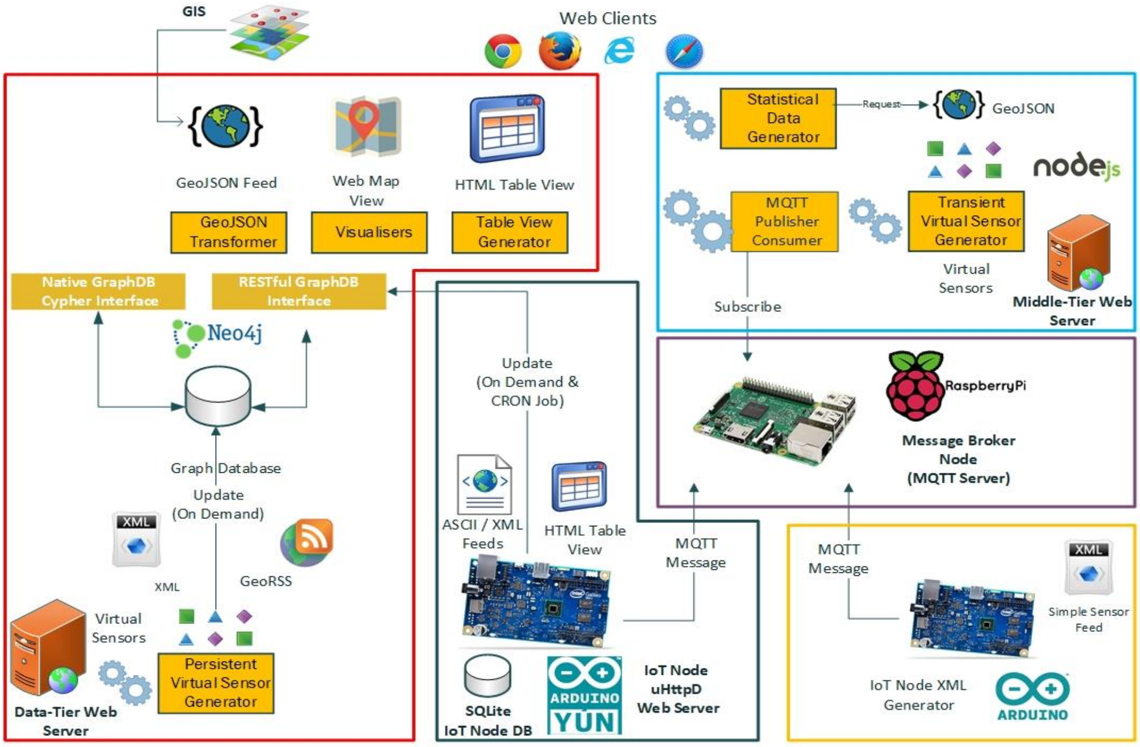


Figure.2 Deployment Diagram

5. COMPONENTS OF THE INFORMATION INTEGRATION ARCHITECTURE

This section will provide the details and the roles of the different software components that were developed and tested during the study. Figure 2 can be used as guide to observe the communication and interactions between these components.

Persistent Virtual Sensor (PVS) Generator: This component is developed to simulate the data coming from various sensors operating in multiple geographic locations, in order to test the data integration in the scenario of this architecture. The virtual sensor generator is composed of 2 software components that generate readings of 100 virtual single board computers (SBCs) where each virtual SBC contains readings from 500 virtual sensors. In total, 50.000 sensor readings can be generated as result of an HTTP Request. The types, names, values and coordinates of the sensors are generated randomly based on parameter value pool, i.) provided by a file authored by the end-user and ii.) also as the parameters of an HTTP Request. For instance, the end-user can specify i.) a set of geo-bounding-box coordinates where these virtual sensors would be located in as HTTP GET Request parameters, or ii.)a value pool of sensor types in a file. One of the PVS components generates a simple XML Feed, the other one generates a GeoRSS Feed, where the information about the location of the sensor is provided as a GeoRSS file. Both outputs are persisted in files.

Graph Database Updater for PVS Generator: The component is developed to test applicability and performance the RESTful calls for spatial object creation in Graph Databases by utilizing the mass GeoInformation in Sensor Feeds. This component processes/parses the all GeoInformation in XML and GeoRSS feeds generated by the Virtual Sensor Generator, then converts them into in-memory-objects and later makes use of HTTP POST requests to persist these objects as spatial objects in the Neo4J Graph Database. The storage of spatial objects in Neo4J is made possible by the utilisation of the Spatial Extension of the Neo4J database. The existence and structure of the persisted spatial objects can be observed by logging into the native user interface of the Neo4J. This component can be utilized on user demand or as a scheduled OS task.

Arduino Yun Local Database Updater and Data Viewer: This component is developed to test the capabilities of Single Board Computers (SBCs) in terms of real-time data acquisition, data storage and data presentation. The Arduino Yun SBC in this setup provides two interfaces, the first one is to the real-time a Sensor Feed generated by programming the Atmel ATmega32U4 microcontroller. This feed is served in form of an ASCII text or Simple XML on a specific port of the Arduino Yun LininoOS (port:80). The second interface is realized by uHTTPd web server running on another port (port: 82) of LininoOS. The Arduino Yun Data Updater and Viewer is set of components residing in this uHTTPd web server (port: 82). There is also an SQLite instance (i.e. IoT Node DB) running on the same operating system. The component operates on demand and once called, the Arduino Yun Data Updater and Viewer component acquires real time senor information (from port: 80) provided by the real time Sensor Feed (in ASCII text or Simple XML) on the same device, parses this information, and uses SQL to update the local SQLite Database, and following this, acquires records from the same database, and visualises it in form of a an HTML table at port:82.

Arduino Yun Graph Database Updater: The purpose of this component is to test the applicability and performance of the RESTful calls for spatial object creation in Graph Databases for storing real-time sensor information through the use of the hardware resources of SBCs such as Arduino Yun. The component parses real-time GeoInformation provided by a light level sensor connected to the Arduino Yun, and uses RESTful approach to persist this information in form of a spatial object in the Neo4J Graph Database. The spatial objects in the Graph Database can be observed by logging into the native user interface of the Neo4J. This component is invoked periodically through a CRON Job on the LininoOS, but also can be utilized by the user demand.

MQTT Publish/Subscribe Component for SBCs: This component is developed with the aim of testing the capability and performance of Single Board Computers (SBCs) for the IoT publish/subscribe model of communication by sending and receiving real-time GeoInformation provided by the sensors. The components are implemented in an Arduino Yun and in an Arduino Uno. The components are based on the Arduino Client for MQTT Library (2016) and Developed using Arduino IDE and runs on the Atmel ATmega32U4 / ATmega328P microcontrollers. The component is not illustrated in the deployment diagram (Figure 2) as it is running directly on the microcontroller level of the SBCs, instead the deployment diagram illustrates the messages sent by the component.

Once the component is compiled and run on Arduino Yun and Uno, these devices begin to broadcast MQTT messages in plain text (on port: 1883) to a MQTT Broker (implemented in the Raspberry Pi). The developed component also allows Arduino Yun and Arduino Uno act as subscribers of an MQTT Broker, so they also listen, receive and dump messages coming from the MQTT Broker. The operation of these components and the MQTT Broker in Raspberry Pi are observed and tested in a client PC, with the help of an MQTT client software, MQTT.fx.

Arduino Uno Simple XML Feed Generator: This component is developed to test the data acquisition and presentation capability of a very basic IoT node, i.e. an Arduino Uno equipped with an Ethernet Shield. The component is developed using Arduino IDE and runs on the Atmel ATmega328P microcontroller. The component is not illustrated in the deployment diagram as it is running directly on the microcontroller level of the Arduino, instead the deployment diagram illustrates the file generated by the component. The job of the component is very straightforward. The component can be called on user demand and once called, it acquires Light Level information from an LDR sensor using an analogue pin of the Arduino Uno, and generates a Simple XML Feed (a micro feed) and presents it to the user.

RESTful MQTT Publisher/Consumer for Async Frameworks: Asynchronous event driven server-side frameworks such as NodeJS, are popular for the IoT Server Side / IoT Cloud architectures. The aim of this component is to test the applicability of utilizing such a framework i.) for connecting to a MQTT Queue and also ii.) for publishing to /and consuming messages transmitted by the MQTT Queue. The developed component is a NodeJS Service, exposing a REST style URI. Using the service the end-user can subscribe to an MQTT Broker over HTTP, by providing the MQTT Broker's IP and the name of the MQTT channel. Once an URI such as [http://192.168.1.11:8081/mqtget /192.168.1.25/ch?channel=test/umit] is provided with an HTTP REQUEST, the component connects to the MQTT Broker for example on 192.168.1.25 (i.e. Raspberry Pi 2) and starts to listen a channel e.g. test/umit. Then, when the service provided by the component receives a message from the MQTT Broker, it parses this message and publishes it into a Web Page.

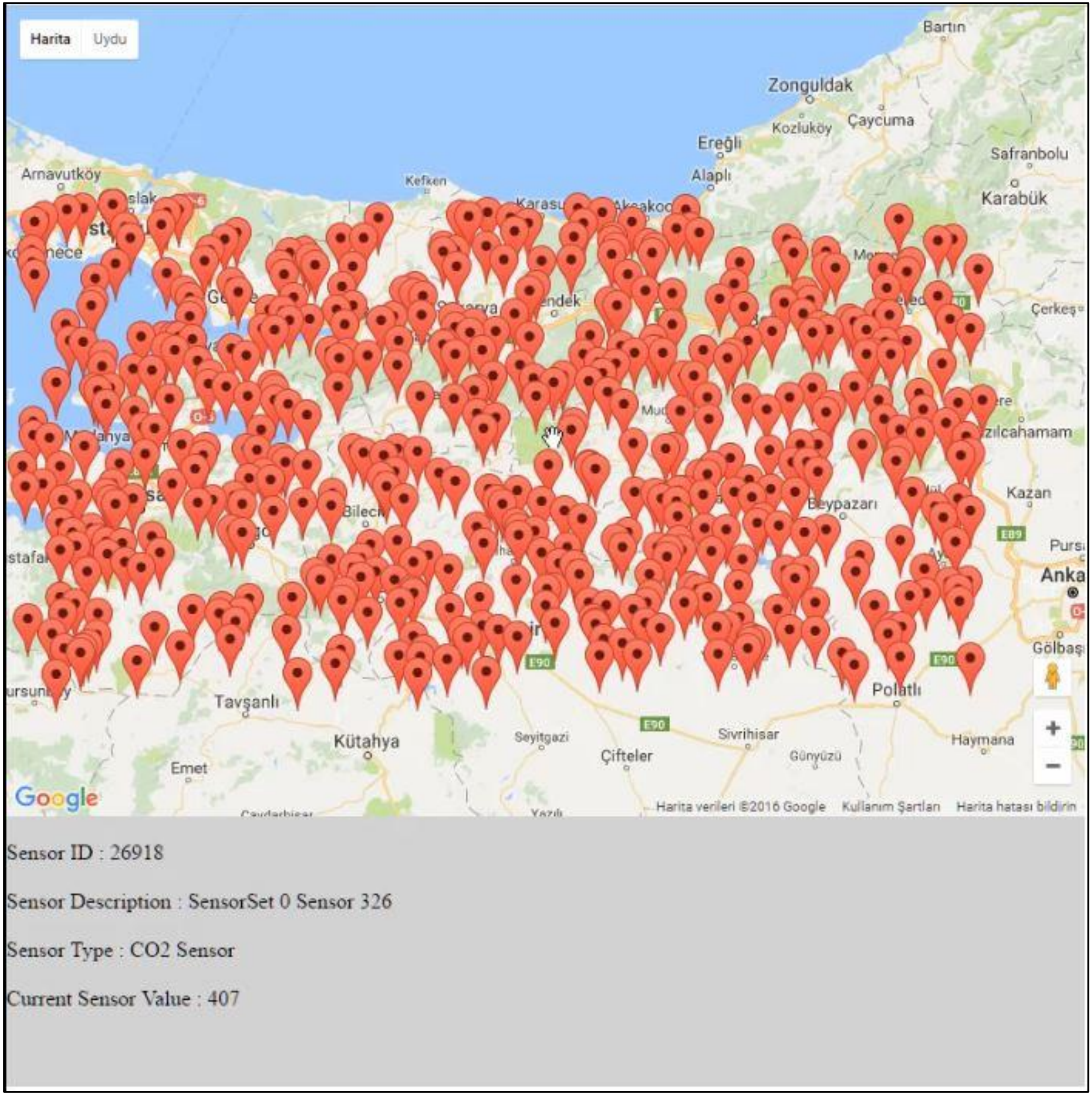
Depending on the user's preference, the contents of the message would also contain GeoInformation (i.e. location of sensors) as well as the semantic information (such as sensor type and value). The developed component also provides the capability of sending a custom MQTT message to an MQTT broker. The message is published through an HTTP GET method, by providing the MQTT Broker's IP and the name of the MQTT channel and Message Content. A URI such as [http://192.168.1.11:8081/mqtsend/192.168.1.25/payload?channel=test/umit&message=1050] would publish a message to the MQTT Broker, e.g. on 192.168.1.25, and to the targeted channel, e.g. test/umit, delivering a payload such as a sensor reading, e.g 1050.

Transient Virtual Sensor (TVS) Generator: The aim of this component is to provide additional set of virtual sensors by utilizing the asynchronous event driven server-side frameworks (such as NodeJS) in order to help the testing of multi-source information integration. The component is developed as a NodeJS service. Unlike with the Persistent Virtual Sensor Generator, the virtual sensors on this tier are in form of GeoJSON, and are transient, i.e. not persistently stored in files or a database. Coordinates of the sensors are generated randomly based on parameter value pool provided by the end-user. The number of sensors to be generated is also defined by a parameter. Once an HTTP GET request is sent through a RESTful URI, the component provides a (server side) response in form of GeoJSON. This response contains a "Feature Collection" of "Points" representing the location and properties (attributes) of the Virtual Sensors.

GeoJSON Transformer: This component is developed to test the data transformation capability in an IoT architecture through utilizing the spatial query capabilities of a Graph Database. The developed component generates a transient server-side response in form of a GeoJSON, as result of a web HTTP GET query sent through a URI. The query parameters include coordinates of a point and a radius value (provided by the user). Once the query is made, the component parses this query, utilizes the RESTful API of the Neo4J to transfer the query to the DB. The Neo4J DB, then makes use of its Spatial extension to respond to this query. As a response, Neo4J sends information related to sensors that can be found within the given radius of a given point. Once the query is responded by the Neo4J, the response is sent to the GeoJSON transformer in form of a JSON. The GeoJSON transformer then parses this JSON file, converts it into GeoJSON and responds back to the end-user. The component operates based on user demand/call.

GeoRSS and GeoJSON Visualizers: These two components are developed with the aim of testing how the integrated GeoInformation provided by multiple SBCs and sensors can be visualised over the web. The developed components are able to visualise both transient and persistent information. The visualizer components are developed in form of Web Maps, by making use of well-known web mapping environment, Google Maps. The API of Google Maps is utilized to visualise the GeoRSS representation of Persistent Virtual Sensors (PVS) through getting sensor information from a set of files and transient real-time information generated by GeoJSON transformer. The second component utilizes GeoJSON Transformer to get the real-time information from sensors. As both visualizers use the same user interface, Figure 3 provides the sensor representation generated by GeoJSON visualizer as an example for sensor representation in these visualizers.

Table View Generator: The component is developed to test how non-graphical views can be generated to present integrated GeoInformation acquired and stored within this architecture. The component generates an HTML page with a table, showing the GeoInformation stored in the Neo4J Graph database. In the first stage an HTTP GET query sent through a URI. The query parameters include coordinates of a point and a radius value (provided by the user). As result of the same steps of interaction explained in GeoJSON transformer, Neo4J sends information related to sensors that are found within the given radius of a point as a response. The Neo4J response is then parsed and an HTML page containing a Table is generated by the component as the final output. The page containing the Table is sent to the end-user as the response.



Sensor ID : 26918

Sensor Description : SensorSet 0 Sensor 326

Sensor Type : CO2 Sensor

Current Sensor Value : 407

Figure 3. Visualisation of Sensor Information using Web Maps

Statistical Data Generator: The end users of IoT applications usually require summary statistics regarding the distribution of sensor types and sensor values. The component is developed to test how such information can be generated based on information presented by the components of this architecture. Statistical Data Generator makes use of information generated by the Transient Virtual Sensor (TVS) Generator. Given information about a number of sensors in form of GeoJSON, the component calculates summary statistics such as mean, standard deviation and variance for each sensor type. The calculated statistics are then presented in form of a JSON file. The component is developed in form of a NodeJS App. The component operates on demand.

The statistical data generator will be extended in the future research in order to cover some machine learning tasks including automatic classification of sensor values furthermore the future research will focus on implementation of LSTM networks which offer a great potential for prediction of the future trends of sensor values.

GIS Based Visualisation: It is also important to test the interaction options for the end-users. Thus, in the final phase of the implementation the information in GeoJSON and GeoRSS formats is acquired directly from i.)the GeoRSS representations and ii.)the transient real-time information in form of GeoJSON generated by the GeoJSON transformer.

Figure 4 provides the representation of this information inside QGIS software. The test revealed that values of sensors along with their type and other semantics attributes can be visualised and queried successfully using a GIS.

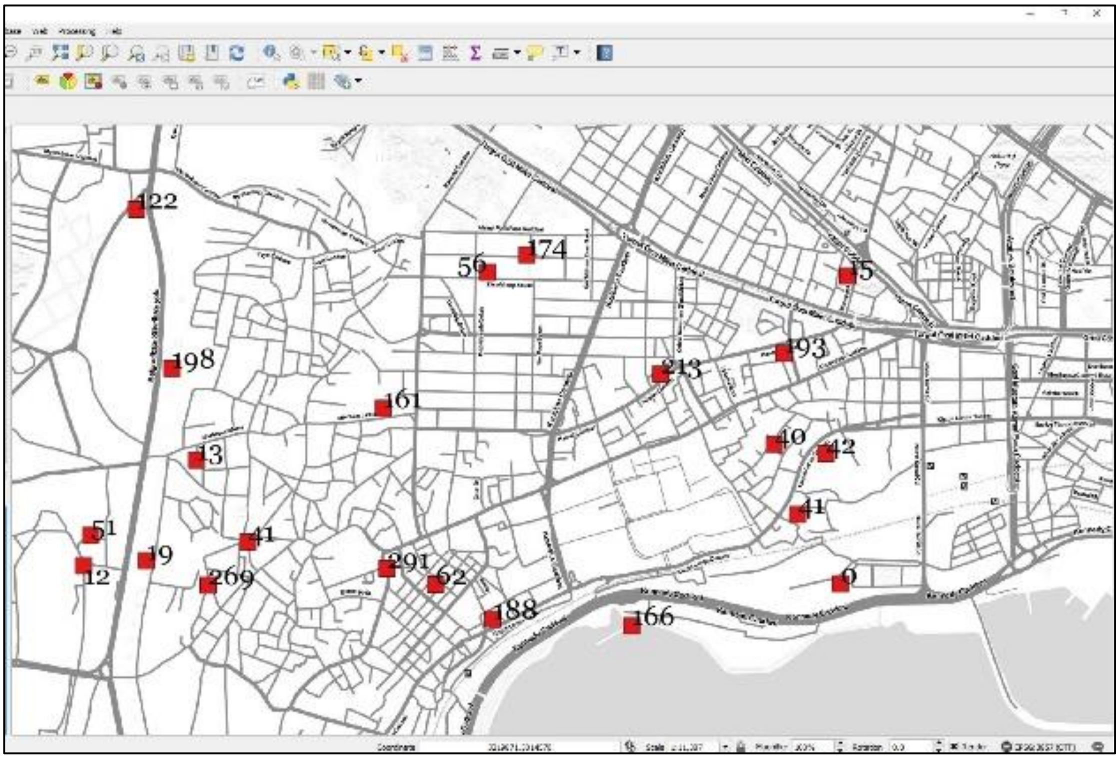


Figure 4. GIS Based Visualisation of Sensor Values

5. CONCLUSION

The research investigated the implementation of various technologies and integration approaches to test the applicability of a GeoInformation integration scenario. The scenario involved the acquisition of real time information from multiple sensors using low-cost hardware (including SBCs), transfer of this information from these devices to traditional data storage mediums such as files or and to modern data storage mediums such as graph databases with spatial information handling capability, development and implementation of RESTful interfaces to the data, and the development of the visualisation components.

The tests also included development of virtual sensor data generator components in order to help the testing of multi-source data integration capability, and mainly the support for non-real-time information fusion. In addition to that two communication models, publish/subscribe and request/response were tested for transferring information between the different IoT nodes.

The implementation results present a proof-of-concept on how multi-source GeoInformation acquired from different type of IoT nodes can be integrated, stored and visualised on different platforms.

Another finding of the research is that the combination of low-cost IoT hardware and graph databases can provide a unique opportunity for storage of large amount of GeoInformation and can respond to different kinds of spatial queries.

The key and unique novelty and scientific contribution of the study is providing a proof of concept architecture for establishing multi source-multi user interfaces, and multiple database/file integration by making use of multiple communication models(i.e. N to N to N. integration).

The research showed that Graph Databases have a unique potential to store and serve real-time GeoInformation. Although not used as commonly as the object-relational databases to store the spatial data, the graph databases have a great potential in dealing with vast amounts of geospatial data. In addition some operations that would be useful in IoT applications such as building up graph based topological data structures (such as space/state or space/event models) and analysis such as shortest path can be facilitated by the use of native graph databases.

Along with the data storage capabilities of the graph databases, this research also tested the automatic generation of interoperable data models such as

GeoJSON on demand, as response to an HTTP request, and the success achieved in this direction provides a proof on the applicability of Graph Databases and RESTful interfaces as a base layer for Geo-IoT web service architectures.

Another novel finding of the research was that publish/subscribe and request/response models of communication can be implemented alongside each other to transfer and interact with real-time GeoInformation.

The architecture can be extended with addition of various types of components and sensors. Other Single Board Computers (SBCs) that can be integrated with this architecture include Onion, Beagle Board, Orange Pi, Banana Pi, Asus Tinker Board, and Latte Panda. The sensors that can be integrated to this architecture is not limited with air quality and there are various types of sensors that can be integrated including traffic congestion detection sensors, water quality monitoring sensors, flood detection sensors, access control sensors, vehicle, human (elderly/child/patient) and product tracking sensors and devices, sensors for smart agriculture and farming, and fire and other hazard detection sensors.

The current visualisation options in the implementation is very limited and this can be expressed as the main weakness of the research. The future research will include new visualisation options using different data 2D data types such as polygons. Furthermore interpolation techniques such as Kriging can be used to estimate values between the sensors, and these estimated values can be visualised using raster visualisation techniques.

The future research will focus on integration of 3D GeoInformation with this architecture as 3D GeoInformation is mostly used 3D City and 3D Building representations, and integration of 3D GeoInformation would provide different query and visualisation opportunities. Visualisation tools such as Virtual Globes can be used for more efficient 3D Visualisations by making use of 3D GeoInformation models.

Due its current focus, this Testbed only tested integration options using standard and commonly used IoT protocols/architectures. On the other hand, in the recent future, implementation of GeoInformation oriented communication protocols and standards such as GeoMQTT, and OGC Sensor Things API, OGC Sensor Observation Service and OGC SensorML will be tested.

REFERENCES

Agoub,A.,Kunde,F.,Kada,M. (2016) Potential of Graph Databases in Representing and Enriching Standardized Geodata, DGPF, Available online at: https://www.dgpf.de/src/tagung/jt2016/proceedings/papers/20_DLT2016_Agoub_et_al.pdf

Akbulut, Z , Özdemir, S , Acar, H , Dihkan, M , Karslı, F . (2018). Automatic Extraction of Building Boundaries from High Resolution Images with Active Contour Segmentation. International Journal of Engineering and Geosciences, 3 (1), pp.37-42. DOI: 10.26833/ijeg.373152

Al-Fuqaha,A., Guizani,M., Mohammadi,M., Aledhari,M., Ayyash,M. (2015) Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications IEEE Communication Surveys & Tutorials, 2015 (17) pp. 2347-2376

Arduino Client for MQTT (2016) Available online: https://pubsubclient.knolleary.net/index.html (accessed on 10-June-2016)

Black, I. and White, G. (2017) Citizen Science, Air Quality, and the Internet of Things. In Internet of Things and Advanced Application in Healthcare, Reis,C.I. Maximiano,M.dS. Eds; IGI Global,USA,2017

Doğan, Y , and Yakar, M . (2018). GIS AND Three-Dimensional Modeling for Cultural Heritages International Journal of Engineering and Geosciences , 3 (2) , pp.50-55 . DOI: 10.26833/ijeg.378257

Elkhodr,M., Shahrestani,S., Cheung.H. (2015) A Smart Home Application based on the Internet of Things Management Platform. Proceedings of the 2015 IEEE International Conference on Data Science and Data Intensive Systems, Dec. 2015. Available online: http://ieeexplore.ieee.org/document/7396548/

Erdem, N and Ince, H . (2016). The Proposal of the Building Application for more Benefiting from Solar Light. International Journal of Engineering and Geosciences , 1 (1) , pp. 7-14 . DOI: 10.26833/ijeg.285215

Gaikwad,P.P., Gabhane,I.P., Golait,S.S. (2015) A Survey based on Smart Homes System Using Internet-of-Things. Proceedings of the International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), Apr. 2015., Available online: http://ieeexplore.ieee.org/ document/7259486/

Göksel, C , Doğru, A . (2019). Analyzing the Urbanization in the Protection Area of the Bosphorus. International Journal of Engineering and Geosciences, 4 (2) , pp.52-57 . DOI: 10.26833/ijeg.446912

Jung,M., Reinisc,C., Kastner,W. (2012) Integrating Building Automation Systems and IPv6 in the Internet of Things. Proceedings of the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, July 2012, Available online: https://ieeexplore.ieee.org/document/6296937

Kamilaris, A., Ostermann, F.O. (2018) Geospatial Analysis and the Internet of Things, International Journal of GeoInformation 7(10), pp.269, Available online: https://doi.org/10.3390/ijgi7070269

Keng, D. and Koo,S.G.M. (2014) Spatial Standards for Internet of Things, Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings 2014), Green Computing and Communications (GreenCom 2014), and Cyber-Physical-Social Computing (CPSCom 2014), Sept. 2014, Available online at: https://ieeexplore.ieee.org/document/7059675

Laska, M., Herle, S., Klamma, R., Blankenbach, J. (2018) A Scalable Architecture for Real-Time Stream Processing of Spatiotemporal IoT Stream Data-

Performance Analysis on the Example of Map Matching, International Journal of GeoInformation 7(10),pp.238. Available online at: https://doi.org/10.3390/ijgi7070238

Libelium Web Site. (2019) Available online: http://www.libelium.com/resources /top_50_iot_sensor_ applications_ranking (accessed on 02-Jan-2019)

Liu,S. and Zhu,G. (2014) The Application of GIS and IoT Technology on Building Fire Evacuation, Procedia Engineering,2014 (71) pp. 577-582

Ming,L.A. and Ling yan,W. (2012) The Study on Geography under Internet of Things, IEEE International Conference on Computer Science and Automation Engineering, June 2012, Available online at: https://ieeexplore.ieee.org/abstract/document/6269452

Mülazimoğlu, E , and Başaraner, M . (2019). User-Centred Design and Evaluation of Multimodal Tourist Maps. International Journal of Engineering and Geosciences , 4 (3) , pp.115-128 . DOI: 10.26833/ijeg.535630

Nguyen, S. H., Yao, Z., Kolbe, T. H. (2017) Spatio-Semantic Comparison of Large 3D City Models in CityGML Using a Graph Database, ISPRS Annals Photogramm. Remote Sens. Spatial Inf. Sci., IV-4/W5,pp. 99-106

Plixer. (2019) Available online : https://www.plixer.com/blog/network-monitoring/network-layers-explained/ (accessed on 02-Feb-2019)

Pozzebon, A. Andreadis, A. Bertoni, D. Bove, C.(2018) A Wireless Sensor Network Framework for Real-Time Monitoring of Height and Volume Variations on Sandy Beaches and Dunes, International Journal of GeoInformation 7(10),pp.141.Available online: https://doi.org/10.3390/ijgi7040141

Rieke, M. Bigagli, L.,Herle, S., Jirka, S. Kotsev, A.,Liebig, T., Malewski, C., Paschke, T., Stasch, C. (2018) Geospatial IoT-The Need for Event-Driven Architectures in Contemporary Spatial Data Infrastructures, International Journal of GeoInformation 7(10) pp.385 Available online: https://doi.org/10.3390/ijgi7100385

Rodriguez,A. (2018) RESTful Web services, IBM Developerworks, Available online : https://developer.ibm.com /articles/ws-restful/ (accessed on 25-July-2018)

Salman,T and Jain, R. (2017) A Survey of Protocols and Standards for Internet of Things, Advanced Computing and Communications, 2017,1, Available online : https://arxiv.org/ftp/arxiv/papers/1903/1903.11549.pdf

Spalazzi,L., Taccari,G., Bernardini,A. (2014) An Internet of Things Ontology for Earthquake Emergency Evaluation and Response. Proceedings of 2014 International Conference on Collaboration Technologies and Systems (CTS), May 2014, Available online: http://ieeexplore.ieee.org/document/6867619/

Technopedia. (2019) Available online: https://www.techopedia.com/definition/8866/physical-layer (accessed on 10-Mar-2019)

Techtarget. (2019) Available online: https://searchnetworking.techtarget.com/definition/Data-Link-layer (accessed on 20-Mar-2019)

Xiaoying,S. and Huanyana,Q. (2011) Design of Wetland Monitoring System Based on the Internet of Things Procedia Environmental Sciences,2011(10),pp. 1046-1051.

Yemenicioglu, C , Kaya, S , Seker, D . (2016). Accuracy of 3D (Three-Dimensional) Terrain Models in Simulations. International Journal of Engineering and Geosciences , 1 (1) , pp.30-33 . DOI: 10.26833/ijeg.28522

Zanella,A, Bui,N., Castellani,A., Vangelista,L., Zorzi,M. (2014) Internet of Things for Smart Cities. IEEE Internet of Things Journal 2014 (1),pp. 22-32

Zhou,Q. and Zhang,J. (2011) Research Prospect of Internet of Things Geography Proceedings of the 19th International Conference on Geoinformatics, Jun. 2011, Available online at: https://ieeexplore.ieee.org/document/5981045

International Journal of Engineering and Geosciences (IJEG),
Vol; 4, Issue; 1, pp. 026-032, February, 2020, ISSN 2548-0960, Turkey,
DOI: 10.26833/ijeg.580373

INVESTIGATION OF THE CENTER COORDINATES OF A CIRCLE WITH UNKNOWN RADIUS USING POLAR MEASUREMENTS

Huseyin Ince [1], Nuri Erdem [2*]

[1]Vocational School of Higher Education for Mapping and Land Survey, Hitit University, Çorum, Turkey, (huseyinince@hitit.edu.tr);**ORCID 0000 0001 6118 5502**

[2*]Engineering Faculty, Department of Geomatics Engineering, Osmaniye Korkut Ata University, Osmaniye, Turkey (nurierdem@osmaniye.edu.tr); **ORCID 0000-0002-1850-4616**

*Corresponding Author, Received: 21/06/2019, Accepted: 28/08/2019

ABSTRACT: Circular pools or buildings may be encountered in production of maps for making development plans, in the first facility cadaster and on land. Geodetic measurements may be made on at least three different points of the circle-shaped pool or building to obtain positions in a coordinate system of measurement points with a geodetic ground-based electronic tachometer after connection to another polygon. It is possible to get a circle from these three points where the coordinates can be calculated, the coordinates of the center of the circle and the radius of the circle with the help of drawing programs (Netcad, Carto CAD, GEOCAD ...). In drawing programs related to cartography, the details of passing the circle from three points, the coordinates of the center of the circle and the radius of the circle cannot be accessed because the details are kept proprietary by software companies. We have been curious about how the coordinates of the center of the circle or the radius of the circle in question are calculated by using the subprograms on a computer. Apart from the information hidden by software companies, two studies which are indirectly close to the subject mentioned in the application have been identified with respect to the solution of the problem on the computer. However, it was found that these studies did not directly illuminate our research topic, so, this issue should be investigated. This study presents theoretical solutions for the topic, makes numerical implementations and provides findings and interpretations.

Keywords: *Polar Coordinates, Coordinates Calculation, Polar Measurements, Circle Center Coordinate Calculation, Radius Calculation*

1. INTRODUCTION

Nowadays, in production of maps for construction of city and town development plans and in the first facility cadastral works, it is possible to encounter pools or buildings in circular form. After connecting to another polygon with an electronic tachometer installed at a polygon point installed in the field, the coordinates of the measurement points may be obtained by making geodetic measurements on at least three different points of the circular pool or building. With the help of drawing programs related to cartography (such as Netcad, Carto CAD, GEOCAD ...), it is possible to obtain the coordinates of the center of a circle and the radius of this circle from these three points, the coordinates of which may also be calculated in the computer environment.

It is not possible to access this information in mapping programs for cartography because the details of passing an area from three points in a coordinate system, coordinates of the center of the circle and the radius of the circle are kept proprietary by software companies. We have been curious about how the coordinates of the center of a circle or the radius of the circle in question are calculated using the subprograms on a computer.

In relation to the solution of this problem, two studies that are indirectly close to the subject mentioned in practice have been identified (Kåsa, 1976; Coppe, 1993). However, it was found that these studies did not directly illuminate our research topic, so this issue should be investigated. Since no study was found in practice regarding the solution of this problem, it is necessary to investigate this problem. This study presents theoretical solutions for the topic, makes numerical implementations and provides findings and interpretations.

A new method is not proposed in this study. As the basic principle of the study, the passage of a circle through three points, the details of obtaining the coordinates and the radius of the center of the circle are intended to be clarified as they are kept secret by software companies.

In field studies related to cartography, it is always possible to encounter a drawing of a circular pool and a building in practice. Therefore, related software is used by practitioners. Since the content of other related software is kept confidential by the related commercial companies, it is not known what kind of problems exist in other software and calculations. In this study, theoretical solutions for the topic are presented, and numerical applications are provided.

The aim of this study is to determine the positions of three points on a circle with polar measurements made from a polygon point and find the coordinates of the circle’s center and radius by taking advantage of geometric positions. After finding the coordinates of the points on the circle, the circle equation can be solved with the help of the coordinates of these points, and the coordinates of the center of the circle can be found in reference to the offset of indirect measures in the least squares method. However, such a solution takes a lot of time in comparison to the method mentioned in this study.

2. CALCULATION OF THE COORDINATES OF MEASURING POINTS FROM A POLYGONAL POINT TO THREE-POINT POLAR MEASUREMENTS ON A CIRCLE

In the vicinity of a circular pool or building, an electronic tachometer is installed at a previously established P_1 polygon site, and a connection to another previously established P_2 point is made. Three separate points (A, B, C) are marked on the tracks near the pool or the floor of the building. Horizontal angles and horizontal distances (β_A, β_B, β_C, S_A, S_B, S_C) are measured by holding the marked point reflector (Figure 1).

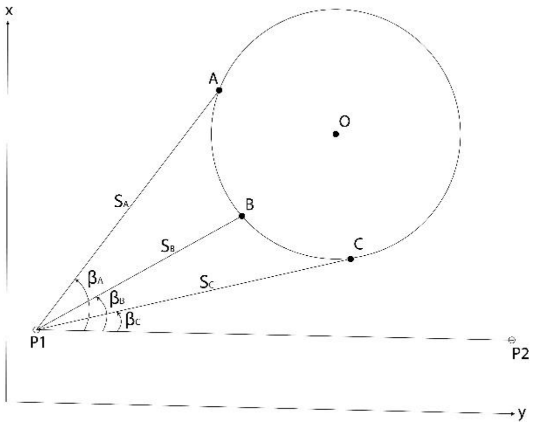


Figure 1: Measurement of points A, B, C with the electronic tachometer installed at point P1.

The coordinates of the points A, B and C are calculated with the following relationships by using the coordinates of the stationary point and the connected points P_1 and P_2 (Y_1, X_1, Y_2, X_2) (Wolf and Ghilani, 2008; Ulvi and Toprak, 2016).

$$(P_1P_2)=\arctan\left(\frac{Y_2-Y_1}{X_2-1}\right) \quad (1)$$

$$(P_1A)=(P_1P_2)-\beta_A \quad (2)$$

$$(P_1B)=(P_1P_2)-\beta_B \quad (3)$$

$$(P_1C)=(P_1P_2)-\beta_C \quad (4)$$

$$Y_A=Y_1+S_A*\sin(P_1A),\ X_A=X_1+S_A*\cos(P_1A) \quad (5)$$

$$Y_B=Y_1+S_B*\sin(P_1B),\ X_B=X_1+S_B*\cos(P_1B) \quad (6)$$

$$Y_C=Y_1+S_C*\sin(P_1C),\ X_C=X_1+S_C*\cos(P_1C) \quad (7)$$

The AB and BC chord lengths are found from the following relations;

$$AB=\sqrt{(Y_B-Y_A)^2+(X_B-X_A)^2} \quad (8)$$

$$BC=\sqrt{(Y_C-Y_B)^2+(X_C-X_B)^2} \quad (9)$$

The fourth point can also be marked and measured to check the traces of the circular pool or building near the ground. In this case, the horizontal angle and horizontal distances measured for the fourth point are used in the relations formed similar to the relations above, the coordinate of the fourth point and the beam length between the nearest measuring point can be calculated.

3. CALCULATION OF THE CENTER COORDINATES AND RADIUS OF A CIRCLE WITH CHORD POINTS

The chord points may be marked on the circle as equal or different chord lengths.

3.1. Calculation of the Center Coordinates and Radius of a Circle with Points Marked on Equal Chord Lengths

If the chord lengths at the points on the circle are equal to each other (AB = BC) (Fig. 2), then the angles of the base at points A, B and C (α) in the isosceles triangles formed in the circle will be equal to each other.

The angles α and ε are obtained from the following relations (Allan et al., 1968; Anderson and Mikhail, 1998; Kavanagh, 2000; Kavanagh, 2003; Kavanagh, 2009; İnce and Erdem, 2019).

$$(BC)=\arctan\left(\frac{Y_C-Y_B}{X_C-X_B}\right) \qquad (10)$$

$$(BA)=\arctan\left(\frac{Y_A-Y_B}{X_A-X_B}\right) \qquad (11)$$

$$\alpha=\frac{(BC)-(BA)}{2} \qquad (12)$$

$$\varepsilon= 200-2\alpha \qquad (13)$$

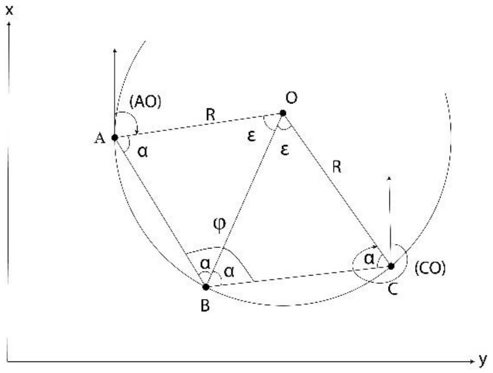


Figure 2: Radius (R) account with the dots marked at equal chord lengths.

In Figure 2, R = AO = BO = CO is found with the following relations;

$$AO = \frac{AB\sin\alpha}{\sin\varepsilon} \qquad (14)$$

$$BO = \frac{BC\sin\alpha}{\sin\varepsilon} \qquad (15)$$

The bearing angles ((AO), (BO), and (CO)) are obtained from the following equations from the points A, B and C to point O;

$(AB)=(BA) \pm 200^g$ (16)
$(CB) = (BC) \pm 200^g$ (17)
$(AO) = (AB) - \alpha$ (18)
$(BO) = (BA) + \alpha = (BC) - \alpha$ (19)
$(CO) = (CB) + \alpha$ (20)

The coordinates of the center of the circle (YO, XO) are derived from the following relations;

A→$Y_O = Y_A + R \sin (AO)$, $X_O = X_A + R \cos (AO)$ (21)
B→$Y_O = Y_B + R \sin (BO)$, $X_O = X_B + R \cos (BO)$ (22)
C→$Y_O = Y_C + R \sin (CO)$, $X_O = X_C + R \cos (CO)$ (23)

(Bannister et al., 1992; Bannister et al., 1998, Erdem and İnce, 2016)

3.2. Calculation of the Center Coordinates and Radius of a Circle with Points Marked on Different Chord Lengths

If the chord lengths between the points marked on the circle are different, then the base angles (α_1, α_2) in the OAB and OBC triangles are not equal to each other. In Fig. 3, the angle (φ) between the BA and BC chords at point B and α_1 are expressed by the following relations (Kavanagh and Bird, 2000; Kavanagh, 2003; Kavanagh, 2009);

$\varphi = (BC) - (BA)$ (24)
$\alpha_2 = \varphi - \alpha_1$ (25)

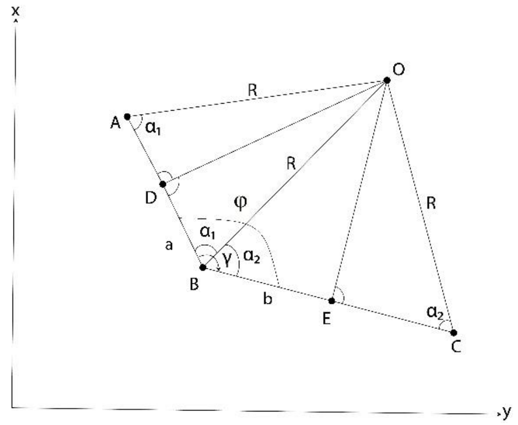


Figure 3: Radius (R) account with the points marked on different chord lengths.

In Fig. 3;
$a = AB/2$ (26)
$b = BC/2$ (27)

D: the midpoint of the AB chord, E: the midpoint of the BC chord, and the following equation is written for R in the right triangles of ODB and OBE.

$$R = \frac{a}{\cos\alpha_1} = \frac{b}{\cos(\varphi-\alpha_1)} \qquad (28)$$

$$a \cos (\varphi-\alpha_1) = b \cos\alpha_1 \qquad (29)$$

The cos (φ-α_1) equation is explained, and as a result of the necessary simplification, α_1 is expressed as follows:

$$\alpha_1=\mathbf{arccos}\left(\frac{a\sin\varphi}{\sqrt{((a\cos\varphi-b)^2+(a\sin\varphi)^2)}}\right) \qquad (30)$$

α_1, which is calculated by the equation (30) is substituted at the bond (28) to obtain R. α_1 is substituted in the relation (25), and α_2 is calculated. (AO), (BO) and (CO), which contain the points O, A and B, and C are obtained from the following equations;

$(AO) = (AB) - \alpha_1$ (31)
$(BO) = (BA) + \alpha_1$ (32)
$(BO) = (BC) - \alpha_2$ (33)
$(CO) = (CB) + \alpha_2$ (34)

The coordinates of the center of the circle (Y_O, X_O) are obtained from the points A, B and C by the relations (21), (22) and (23) (Wilfred, 2001; Wolf and Ghilani, 2008).

The fourth point in equal spacing can also be measured on the traces of the circular pool or building near the ground for control purposes. In this case, the calculated coordinates of the fourth point as defined in the first section and the correlations formed similar to the one above are used, and the coordinates of the circle center O and the radius of the circle can be calculated. If significant differences are observed in the calculation results, the measurements and calculations are repeated. If there are small differences (1-2 cm) between the results, the results that are obtained are averaged.

4. NUMERICAL IMPLEMENTATIONS

Table 1- Coordinates of polar coordinates and polygons made from point P_1

Station number	Number of points of observation	Horizontal Angle	Horizontal Distance	Y	X
P_1				8210.227	7215.475
	P_2	0.0000		8560.735	7225.608
	A	25.9550	83.065		
	B	23.0359	86.219		
	C	21.8019	90.915		

Solution;
ΔY_{P1P2} = 350.508 m., ΔX_{P1P2} = 10.133 m., (P_1P_2) = arctan $(\frac{350.508}{10.133})$ = $98^g.1601$
β_A= $25^g.9550$, β_B = $23^g.0359$, β_C = $21^g.8019$, S_A=83.065 m., S_B = 86.219 m., S_C = 90.915 m.
The angles of the points that are required to obtain the coordinates of the points A, B and C from the point P1 are obtained from the following equations.

$(P_1A) = (P_1P_2) - \beta_A = 72^g.2051$, $(P_1B) = (P_1P_2) – \beta_B = 75^g.1242$ and
$(P_1C) = (P_1P_2) - \beta_C = 76^g.3582$

Using the equations (5), (6) and (7), the coordinates of the points A, B and C are obtained.
Y_A = 8285.500 m., XA = 7250.600 m.
Y_B = 8289.947 m., XB = 7248.314 m.
Y_C =8294.945 m., XC = 7248.467 m.
(BA) = 400 – arctan $(\frac{4.447}{2.286})$ = $330^g.2285$
(BC) = arctan $(\frac{4.998}{0.153})$ = $98^g.0518$

The AB and BC chord lengths are obtained from the following equations (7) and (8).
AB = 5.000 m., BC = 5.000 m.
$\alpha=\frac{(BC)-(BA)}{2}$ = $83^g.9116$, R = $\frac{2.5}{cos\alpha}$ = 9.9986 m. ≈ 10.00 m.

The bearing angles from A, B and C to the center point O are obtained from the following equations.
(AO) = (AB) - α = $46^g.3169$, (BO) = (BC) - α = $14^g.1302$, (CO) = (CB) + α = $381^g.9634$
Using the equations (21), (22) and (23), the coordinates of the center O are obtained as follows.
A→Y_O = 8292.150 m., X_O = 7258.068 m.
B→Y_O = 8292.148 m., X_O = 7258.069 m.
C→Y_O = 8292.150 m., X_O = 7258.068 m.
The exact coordinates of O: Y_O = 8292.149 m., X_O = 7252.068 m.

Numerical Implementation 1:

The points A, B and C are marked on the same radius of 5.00 m chord lengths on a circle with an unknown radius shown in Figure 1. These points are measured by the polar method with connection to the point P_2 by an electronic tachometer installed at the point P_1. Using the given measurements and the coordinates of the polygon points given in Table 1, we calculate the coordinates of the marked points of A, B and C, the radius of the pool and the coordinates of the center of the pool.

Numerical Implementation 2:

The points A, B and C are marked on the chord lengths of 5.00 m. and 8.00 m. on a circle-shaped pool with an unknown radius as shown in Fig. 4. These points are measured by the polar method with connection to the point P_2 by an electronic tachometer installed at the point P_1. Using the given measurements and the coordinates of the polygon points given in Table 2, we calculate the coordinates of the marked points A, B and C, the radius of the pool and the coordinates of the center of the pool.

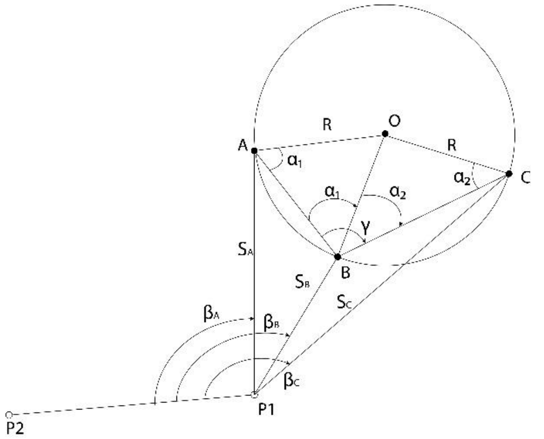


Figure 4: A pool with an unknown radius.

Table 2- Coordinates of polar coordinates and polygons made from point P_1

Station number	Number of points of observation	Horizontal angle	Horizontal Distance	Y	X
P_1				7802.318	5335.972
	P_2	0.0000		7595.612	5230.674
	A	187.8733	83.065	64.192	
	B	191.9292	86.219	66.943	
	C	195.3468	90.915	73.995	

Solution;

$(P_1P_2) = 200 + \arctan\left(\frac{\mathbf{206.706}}{\mathbf{105.298}}\right) = 270^{g}.0059$

$\beta_A = 187^{g}.8733$, $\beta_B = 191\,^{g}.9292$, $\beta_C = 195\,^{g}.3468$, $S_A = 64.192$ m., $S_B = 66.943$ m. and $S_C = 73.995$ m.

The angles of the points that are required to obtain the coordinates of the points A, B and C from the point P1 are obtained from the following equations.

$(P_1A) = (P_1P_2) + \beta_A - 400 = 57^{g}.8792$, $(P_1B) = (P_1P_2) + \beta_B - 400 = 61\,^{g}.9351$,

$(P_1C) = (P_1P_2) + \beta_C - 400 = 65\,^{g}.3527$

Using the equations (5), (6) and (7), the coordinates of the points A, B and C are obtained.

$Y_A = 7852.965$ m., $X_A = 5375.412$ m.

$Y_B = 7857.647$ m., $X_B = 5373.656$ m.

$Y_C = 7865.622$ m., $X_C = 5374.284$ m.

$(AB) = 200 - \arctan\left(\frac{\mathbf{4.682}}{\mathbf{1.756}}\right) = 122\,^{g}.8430$

$(BC) = \arctan\left(\frac{\mathbf{7.975}}{\mathbf{0.628}}\right) = 94\,^{g}.9972$

The AB and BC chord lengths are obtained from the following equations (7) and (8).

AB= 5.00 0m., BC = 8.000 m.

$\varphi = (BC) - (BA) + 400 = 172\,^{g}.1542$,

$\alpha_1 = \arccos\left(\frac{\mathbf{2.5*sin\varphi}}{\sqrt{[(\mathbf{2.5*cos\varphi-4})^2+(\mathbf{2.5*sin\varphi})^2]}}\right) = 89\,^{g}.3394$

$\alpha_2 = \varphi - \alpha_1 = 82\,^{g}.8148$, $R = \frac{\mathbf{2.5}}{\mathbf{cos\alpha1}} = 14.999$ m., $R = \frac{\mathbf{4.0}}{\mathbf{cos\alpha2}} = 14.999$ m., R = 15.00 m.

The bearing angles from A, B and C to the center O are obtained from the following equations.

$(AO) = (AB) - \alpha_1 = 46^{g}.3169$, $(BO) = (BA) + \alpha_1 - 400 = 12\,^{g}.1824$

$(BO) = (BC) - \alpha_2 = 12\,^{g}.1824$, $(CO) = (CB) + \alpha_2 = 377\,^{g}.8120$

Using the equations (21), (22) and (23), the coordinates of the center O are obtained as follows.

A→$Y_O = 7860.499$ m., $X_O = 5388.382$ m.

B→$Y_O = 7860.500$ m., $X_O = 5388.382$ m.

C→$Y_O = 7860.500$ m., $X_O = 5388.381$ m.

The exact coordinates of O: $Y_O = 7860.500$ m., $X_O = 5388.382$ m.

5. ALGORITHM FOR CALCULATION OF THE CENTER AND THE RADIUS

1- Enter the stationary point number and coordinate (P_1, YP_1, and XP_1)

2- Enter the starting point number and coordinate (P_2, YP_2, and XP_2)

3- Enter Measured Horizontal Angles and Horizontal Lengths of the Points A, B and C Measured on the Circle Arch (BTA, BTB, BTC, SA, SB, SC) (Explanation: Horizontal Angles are β_A= BTA, β_B =BTB, β_C=BTC. Horizontal Lengths are SA=S_A, SB=S_B, SC=S_C)

4- The (P_1P_2) bearing angle is calculated based on the differences in the coordinates (ΔYP_1P_2 and ΔXP_1P_2)

$\Delta YP_1P_2=YP_2-YP_1$

$\Delta XP_1P_2=XP_2-XP_1$

The Bearing Angle Calculation Subprogram, by taking into account the expressions of ΔYP_1P_2, ΔXP_1P_2 and (P1P2), calculates the angle of the bearing angle in radian units and converts it to grad units.

5- Calculation of the bearing angles [(P1A), (P1B), (P1C)] for the points A, B and C from the point P_1 in grad and radian units;

Calculation of the bearing angles in grad units from the point P_1 to the points A, B, C

$(P_1A)= (P_1P_2)$-BTA

$(P_1B)= (P_1P_2)$-BTB

$(P_1C)= (P_1P_2)$-BTC

$\rho=200/\pi$

Conversion of the bearing angles into radians

$(P_1A)= (P_1A)/\rho$

$(P_1B)= (P_1B)/\rho$

$(P_1C)= (P_1C)/\rho$

6- Calculating the coordinates of the points A, B and C from the point P_1;

$YA=YP_1+SA*\sin(P_1A)$

$XA=XP_1+SA*\cos(P_1A)$

$YB=YP_1+SB*\sin(P_1B)$

$XB=XP_1+SB*\cos(P_1B)$

$YC=YP_1+SC*\sin(P_1C)$

$XC=XP_1+SC*\cos(P_1C)$

7-Calculating the angle of bearing (BC) and bearing (BA);

$\Delta YBC=YC-YB$

$\Delta XBC=XC-XB$

The subprogram for calculating the bearing, by taking into account the signs of ΔYBC and ΔXBC, calculates the angle of the bearing (BC) in radian units and converts it to grad units.

$\Delta YBA=YA-YB$

$\Delta XBA=XA-XB$

The subprogram for calculating the bearing, by taking into account the signs of ΔYBA and ΔXBA, calculates the angle of the bearing (BA) in radian units and converts it to grad units.

8- Calculation of the chord lengths BA and BC;

BA=$\sqrt{((\Delta YBA^2) + (\Delta XBA^2))}$

BC=$\sqrt{((\Delta YBC^2) + (\Delta XBC^2))}$

9- Comparison of the BA and BC chord lengths
If BA=BC go to step 10
If BA≠BC go to step 14

10- Calculation of base angles and R values of triangles
α= ((BC)-(BA))/2
Conversion of α to radian units
$\alpha=\alpha/\rho$
$\varepsilon=\pi-2\alpha$
Calculation of R, AO, BO and average R
R=AO=BO
AO=AB*sinα/sinε
BO=BC*sinα/sinε
R=(AO+BO)/2
Conversion of the (BA) and (BC) bearing angles to radian units
(BA) = (BA)/ρ
(BC)= (BC)/ρ

11- Calculation of the bearing angles [(BO1), (AO), (BO2), (CO)] from the points A, B and C points to the center of the circle
(BO1) = (BA)-α
If (BA) >π then (AB) = (BA)-π
If (BA) <π then (AB) = (BA) +π
(AO)= (AB)-α
(BO2)= (BC)-α
If (BC) >π then (CB) = (BC)-π
If (BC) <π then (CB) = (BC) +π
(CO)= (CB) +α

12- Calculating the coordinates of the center of the circle from the points A, B and C;
YOA=YA+R*sin(AO)
XOA=XA+R*cos(AO)
YOB=YB+R*sin(BO)
XOB=XB+R*cos(BO)
YOC=YC+R*sin (CO)
XOC=XC+R*cos (CO)

13- Printing the radius R of the circle and the coordinates of the center O

14- Calculation of the angle φ between the chords BA and BC cat the point B and the values a and b
φ=(BC)-(BA)
φ=φ/ρ
a=AB/2
b=BC/2

15- Calculation of the base angles (α_1, α_2) of triangles and R;
α_1=arccos (a*sinφ/$\sqrt{((\boldsymbol{acos\varphi} - \boldsymbol{b})^2 + (\boldsymbol{asin\varphi})^2)}$)
$\alpha_2=\varphi-\alpha_1$
$R_1=a/\cos\alpha_1$
$R_2=b/\cos\alpha_2$
$R=(R_1+R_2)/2$

16- The points A, B and C from the center of the circle for the calculation of the angle;
$(BO_1)=(BA)+\alpha_1$
If (BA)>π then (AB)=(BA)-π
If (BA)<π then (AB)=(BA)+π
$(AO)=(AB)-\alpha_1$
$(BO2)=(BC)-\alpha_2$
If (BC)>π then (CB)=(BC)-π
If (BC)<π than (CB)=(BC)+π
$(CO)=(CB)+\alpha_2$
17- Go to step 13 to calculate the coordinates of the center of the circle at the points A, B and C.

6. DISCUSSION

(1), (10) and (11) in the case of zero in the case of division; In equation (1), XP1 = XP2, in equation (10), XB = XC and (11) must be XB = XA. The equation (1) can be XP1 = XP2. However, considering that the points A, B, C are on the circle arc, it is not possible to have XB = XC, XB = XA. The results of equations (21), (22) and (23) must be equal to each other.

However, due to incidental errors that cannot be avoided during measurement, there may be little differences in cm units in the calculated coordinate results. In this case, it would be more appropriate to obtain the average of the results gathered from the equations. In cartography, one tenth of the circle radius is taken as the best distance between the points to be applied on the circle arc.

In determining the center, the distance to be taken between the points to be marked on the circle arc should be carefully selected to not be less than one tenth of the circle radius length. If a point is taken in a range smaller than the specified value, the lines connecting these points to the center O of the circle intersect at a very narrow angle. This will negatively affect the position accuracy of the point O.

More than three points can be taken on the circle arc. In this case, the coordinate of the center and the length of the radius, will be calculated with the least squares method according to the adjustment calculation.

The positioning of the three points on the circle can be achieved in two ways, as stated in the article. In the first method, the points are positioned on the circle so that they are of equal beam length. In the second method, the points are positioned at different beam lengths. For ease of calculation, it is recommended to position the points on the circle according to the first method.

7. CONCLUSIONS

- The length of a chord and the horizontal distances to be measured to determine the radius and center coordinates of a circular pool of unknown radii; measurements should be made in millimetric precision for control and consistency of results.
- In measurements to determine the radius of a building with a radius of unknown radius and the coordinates of its center; a) The reflector

used to measure the sharpness of the reflector head will not be exactly perpendicular to the projection of the building due to the thickness of the head, b) the distance from the building line to the pointed end of the reflector should be added to the horizontal distance measured with the aid of a ruler in millimetric precision.

- Where the chord length is directly measured, the equal chord length method should be applied in the calculation of the circle radius and the circle center coordinate, because of the ease of calculation.
- Where the chord length cannot be measured directly, three points on the circle arc should be marked very precisely so as to cover half the circumference of the entire circle.
- At places where the beam length cannot be measured directly, at least three separate points should be marked very precisely on the arc of the circle to cover half of the circumference of the entire circle and for control purposes.
- The reflector used in polar measurements must be as sharp as possible in the tip that comes into contact with the floor.

On the subject of the Type Reconstruction Regulation;

- Width of road in 3-storey buildings (excluding basement): 7.00 - 10.00 m
- Width of road in 4-storey buildings (excluding basement): 10.00 - 12.00 m.
- Width of road in 5-storey buildings (excluding basement): 12.00 - 15.00 m, this cannot be less, and provisions are included.

REFERENCES

Allan, A.L., Hollwey, J.R., and Maynes, J.H.B. 1968. Practical Field Surveying and Computations. Heinnemann, London.

Anderson, J.M. and Mikhail, E.M., 1998. Surveying theory and practice. 7th ed. Boston, MA: Mc Graw Hill.

Bannister, A., Baker, R., and Raymond, S., 1992. Surveying. John Willey & Sons. Inc., New York, pp. 186–188.

Bannister, A., Raymond, S. and Baker, R., 1998. Surveying. 7th ed. Harlow, UK: Addison Wesley Longman Limited Edinburg Gate, 81–82.

Coope, I. D. , 1993. Circle fitting by linear and nonlinear least squares. Journal of Optimization Theory and Applications, 76(2), 381-388.

Erdem, N., and İnce, H., 2016. The Proposal Of The Building Application For More Benefiting From Solar Light, International Journal of Engineering and Geosciences, 1 (1), 7-14. DOI: 10.26833/ijeg.285215

İnce, H, Erdem, N., 2019. Positioning Buildings on a Zoning Island to Provide Maximum Shading: A Case Study. International Journal of Engineering and Geosciences, 4 (3), 129-140. DOI: 10.26833/ijeg.525020

Kåsa, I., 1976. A circle fitting procedure and its error analysis. IEEE Transactions on instrumentation and measurement, (1), 8-14.

Kavanagh, F. B., 2003. Geomatics. Pearson Education ˙Inc. Prentice Hall, Upper Saddle River, NJ pp. 159–160.

Kavanagh, F. B., and Bird, S.J.G., 2000. Surveying Principles and Applications. Pearson Education ˙Inc. Prentice Hall, Upper Saddle River, NJ

Kavanagh, F.B., 2009. Surveying principles and application. 8th ed. Columbus: Pearson Education Inc., 525-526.

Ulvi, A , Toprak, A., 2016. Investigation of Three-Dimensional Modelling Availability Taken Photograph of the Unmanned Aerial Vehicle; Sample of Kanlidivane Church. International Journal of Engineering and Geosciences, 1 (1), 1-7. DOI: 10.26833/ijeg.285216

Wilfred, S., 2001. Engineering surveying. 5th ed. Oxford, UK: Butterworth Heinemann, 395-397.

Wolf, P. R., and Ghilani, C. D., 2008. Elementary Surveying an Introduction to Geomatics. 12th Edition, Upper Saddle River, New Jersey: Pearson Prentice-Hall. pp 271-274, 716-717, 720-721

International Journal of Engineering and Geosciences (IJEG),
Vol; 5, Issue; 1, pp. 033-041, February, 2020, ISSN 2548-0960, Turkey,
DOI: 10.26833/ijeg.580510

INVESTIGATION OF BLACK SEA MEAN SEA LEVEL VARIABILITY BY SINGULAR SPECTRUM ANALYSIS

Cansu Beşel[1*], Emine Tanır Kayıkçı [1]

[1]Karadeniz Technical University, Engineering Faculty, Department of Geomatic Engineering, Trabzon, Turkey
(cansubesel/etanir@ktu.edu.tr); **ORCID 0000-0003-3434-6483, ORCID 0000-0001-8259-5543**

*Corresponding Author, Received: 21/06/2019, Accepted: 22/07/2019

ABSTRACT: The mean sea level has been continuously increasing since the end of the 19th century and will continue to increase in the 21st century. The Intergovernmental Panel on Climate Change (IPCC) states that the sea level will rise by 40-60 cm until 2100. This situation will lead to social and economic problems, especially in coastal areas. For this reason, studies on sea level determination have great importance in our country. In this paper, we used the singular spectrum analysis (SSA) to investigate mean sea level variability along the coasts of the Black Sea, which is an intercontinental inland sea. This study aimed to determine the trend in sea level change along the coasts of the Black Sea over time. The mean sea level data from 10 tide gauge stations (Amasra, Batumi, Bourgas, Constantza, Igneada, Poti, Sevastapol, Trabzon II, Tuapse and Varna) are analyzed in this study. The mean sea level data were obtained from the Permanent Service for Mean Sea Level (PSMSL). SSA was applied to the mean sea level observations at tide gauges stations, and the results were interpreted. According to the analysis results, there are increasing trends at the Batumi, Poti, Tuapse, Constantza, Sevastopol and Varna stations. The obtained trend of Bourgas station is not significant. There is The results of the Amasra, Igneada and Trabzon II tide gauge stations were inadequate in interpreting any change. There were no trends at these stations. Close eigenvalues were computed from the mean sea level at the tide gauge stations. This situation shows that there is a dominant seasonal component in the time series.

Keywords: *Black Sea, Mean Sea Level, Tide Gauge, Singular Spectrum Analysis*

1. INTRODUCTION

The climate depends on the correlation between the atmosphere, hydrosphere, cryosphere and geosphere (Nacef et al.,2016). Sea level rise, which is predicted to significantly affect coastal areas, is one of the most important pieces of evidence for climate change. Identifying and understanding the causes of sea level change are important in climate change studies at global and regional scales. Sea-level change occurs at different rates over a broad time scale depending on the location. Global mean rate which is estimated satellite altimetry-based has reported a ± 3 mm/year change over a few decades (Cazenave et al., 2014). However, this rate varies across the Earth. For this reason, it is important to predict sea level changes, determine the areas that will be affected by these changes and take precautions. Observations of sea level change are also important for geodesy. In geodesy, the determination of sea level change is crucial in terms of identifying the vertical datum and determining the geoid. In the most applications require a datum for determining ocean depths (Yilmaz et al.,2016).

In our country, which is surrounded by the sea on three sides, sea level monitoring and forecasting studies are extremely important. Karaca and Ünal (2003) noted that the sea level increase would cause a loss in our country's national income of approximately 10%.

The estimates of sea level change are computed by satellite altimetry and tide gauge data. Related to sea level, established sea level monitoring networks and data centers have been established at global, regional and local scales. A few of these institutions include the Permanent Service for Mean Sea Level (PSMSL) and the European Global Ocean Observing System (EuroGOOS). The PSMSL serves to provide hundreds of tide gauge data points at monthly and yearly scales, controlling and presenting. In our country, tide gauge data are presented by the Turkish Sea Level Monitoring System (TUDES; URL 1).

Time series models are widely used in the analysis and estimation of climate data. Singular spectrum analysis (SSA) is a powerful technique in time series analysis. This technique is applied to many problems (see, Ghil et al.,2002). It is particularly advantageous in complex seasonal component estimation and the analysis of non-stationary time series (Hassani et al., 2009). In the literature, there are many studies on SSA theory and applications (Golyandina 2001; Hassani et al., 2009; Golyandina 2010; Golyandina ve Zhigljavsky 2013).

In this study, the mean sea level data from 10 tide gauge stations (Amasra, Batumi, Bourgas, Constantza, Igneada, Poti, Sevastapol, Trabzon II, Tuapse and Varna) were analyzed using SSA. With SSA, the change in sea level along the coasts of the Black Sea, as well as harmonic oscillations at these stations and seasonal effects, will be determined. Concurrently, the sea level change along the coasts of the Black Sea will be interpreted through the obtained results.

2. DATA AND METHODS

2.1 Study Area and Tide Gauge Data

In this paper, we used the mean sea level data from tide gauge records along the Black Sea coast. The Black Sea is a unique location, as it is the most isolated inland sea in the Atlantic Ocean system (Goriacikin and Ivanov 2006). There are several studies on sea level change in this region. In these studies, the sea level of the Black Sea coast increased rapidly, see Kubryakov and Stanichyni 2013; Vigo et al., 2005; Avşar et al., 2015.

The mean sea level data from 10 tide gauge stations (Amasra, Batumi, Bourgas, Constantza, Igneada, Poti, Sevastapol, Trabzon II, Tuapse and Varna) located along the coasts of the Black Sea are used in this study. The Amasra, Igneada and Trabzon II tide gauges are in Turkey; the Batumi and Poti stations are in Georgia; the Bourgas and Varna stations are in Bulgaria; the Constantza station is in Romania; and the Sevastopol station is in the Ukraine (Figure 1).

We used monthly mean sea level time series from the PSMSL, and the data sets are from the Revised Local Reference (RLR) in the PSMSL (URL 2). The data spans for each station are different from each other, and the data have gaps of 5% and 14% (Table 1).

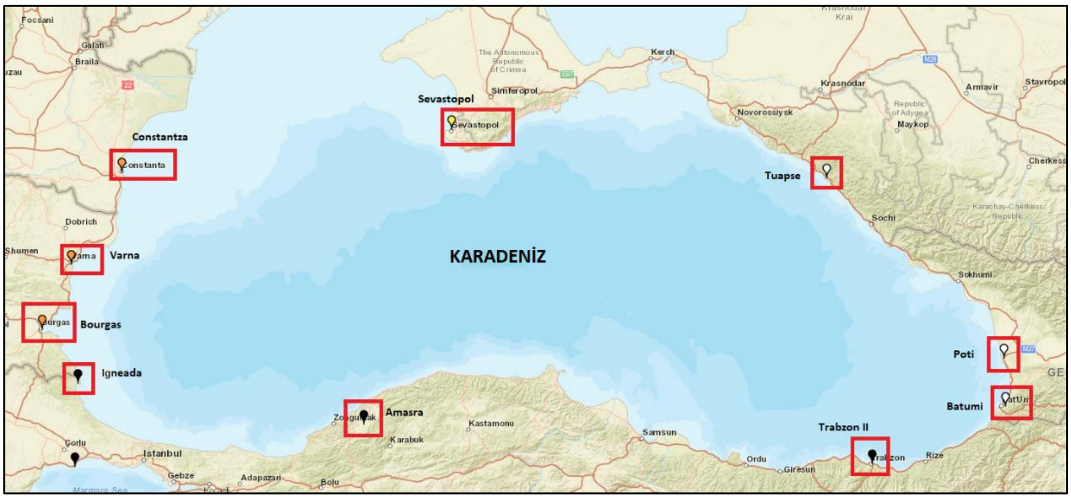


Figure.1 Tide gauge stations used in this study (https://www.psmsl.org/data/obtaining/map.html)

Table 1 Tide gauge stations and geographic coordinates

Station	Country	Latitude (Degree)	Longitude (Degree)	Time Span
Amasra	Turkey	41.4333	32.2333	2001 – 2009
Batumi	Georgia	41.6333	41.7000	1882 – 2015
Bourgas	Bulgaria	42.4833	27.4833	1929 – 1996
Constantza	Romania	44.1666	28.6666	1933 – 1997
Igneada	Turkey	41.8833	28.0166	2002 – 2009
Poti	Georgia	42.1666	41.6833	1874 – 2015
Sevastapol	Ukraine	44.6166	33.5333	1910 – 1994
Trabzon II	Turkey	41.0000	39.7333	2002 – 2009
Tuapse	Russia	44.1000	39.0666	1917 – 2017
Varna	Bulgaria	43.1833	27.9166	1929 – 1996

2.2 Singular Spectrum Analysis

Singular spectrum analysis is a powerful filtration technique. This method has a wide range of applications in the fields of hydrology, oceanography, medicine, economy and earth sciences. SSA is a nonparametric approach that is widely used in time series analysis (Hassani et al. 2009). This method extracts periodic and quasi-periodic signals in a time series. A spectrum of eigenvalues is used to determine these signals.

The SSA technique consists of four steps: embedding, singular value decomposition, grouping and diagonal averaging. In the first stage, the original time series is decomposed, and in the second stage, the original time series is reconstructed. Embedding and singular value decomposition belong to the decomposition stage, and the grouping and diagonal averaging belong to the reconstruction stage. The steps of this technique are shown in Figure 2. The main components selected for reconstruction include information about the trend and harmonic component. Spectral decomposition and reconstruction provide a susceptible determination of trends, seasonal fluctuations, and low frequency components.

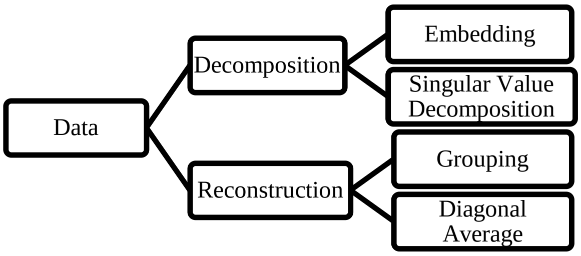


Figure. 2 Singular spectrum analysis steps

The first step in the SSA technique is embedding. In this step, a one-dimensional series is transferred into a multidimensional series. Therefore, the Hankel matrix (trajectory matrix) is taken from the original time series.

Time series $\mathbf{X} = (\mathbf{x_1}, \dots, \mathbf{x_N})$, with length N, is equalized to an L-series vector, as shown in Eq.(1).

$$\mathbf{X_i} = (\mathbf{x_i}, \dots, \mathbf{x_{i+L-1}})^{\mathbf{T}} \tag{1}$$

where L is the window length, or embedding dimension; $\mathbf{1 < L < N}$; and $\mathbf{1 \leq i \leq K}$, where

$$\mathbf{K = N - L + 1} \tag{2}$$

The Hankel matrix of series X,

$$\mathbf{X} = [\mathbf{X_1}, \dots, \mathbf{X_N}] = (\mathbf{x_{ij}})_{\mathbf{i,j=1}}^{\mathbf{L,K}} = \begin{bmatrix} \mathbf{x_1} & \dots & \mathbf{x_K} \\ \mathbf{x_2} & \dots & \mathbf{x_{K+1}} \\ \vdots & \dots & \vdots \\ \mathbf{x_L} & \dots & \mathbf{x_N} \end{bmatrix} \tag{3}$$

The second step in SSA is the singular value decomposition (SVD). Then, the computed eigenvalues and eigenvectors of the matrix $\mathbf{S = XX^T}$ and SVD are applied to the Hankel matrix X.

$$\mathbf{X} = \sum_{\mathbf{i=1}}^{\mathbf{d}} \sqrt{\boldsymbol{\lambda}_\mathbf{i}}\, \mathbf{U_i V_i^T} \tag{4}$$

where $\boldsymbol{\lambda}_\mathbf{i}$, is the eigenvalue of the S matrix ($\boldsymbol{\lambda}_\mathbf{1} \geq \boldsymbol{\lambda}_\mathbf{2} \dots \geq \boldsymbol{\lambda}_\mathbf{L} \geq \mathbf{0}$). According to SSA theory, close eigenvalues indicate the existence of seasonal components in the time series (Khelifa vd.,2016). In Eq. 4, the ($\sqrt{\boldsymbol{\lambda}_\mathbf{i}}, \mathbf{U_i}, \mathbf{V_i}$) components are referred to as the eigentriple of the X matrix, where $\mathbf{U_i}$ is the normalised eigenvector corresponding to the eigenvalues.

When the eigenvalues are found, the decomposition stage is completed and proceeds to the reconstruction stage. The grouping step is first applied in this stage. The X matrix is split into several groups, and the matrices in each group are summed. Let,

$$\mathbf{X_i} = \sqrt{\boldsymbol{\lambda}_\mathbf{i}}\, \mathbf{U_i V_i^T} \tag{5}$$

$$\mathbf{X_I} = \mathbf{X_{i_1}} + \mathbf{X_{i_2}} + \dots + \mathbf{X_{i_m}}, \quad \mathbf{I} = \{\mathbf{i_1}, \dots, \mathbf{i_m}\} \tag{6}$$

Here, $\mathbf{I} = \{\mathbf{i_1}, \dots, \mathbf{i_m}\}$ represents eigentriple grouping. Then, each matrix $\mathbf{X_{I_j}}$ $(\mathbf{j = 1, \dots, d})$ is transformed into a new series with length N, where d is the rank of a matrix. This step is referred to as diagonal averaging.

Let the Y matrix be computed. Y is the length $\mathbf{L} \times \mathbf{K}$ matrix, and $\mathbf{y_{ij}}$ is an element of $\mathbf{Y}(\mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_N})$. As a result, Y is converted to series $(\mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_N})$, and the reconstruction elements are written as:

$$\mathbf{y_k} = \begin{cases} \frac{1}{k}\sum_{m=1}^{k} \mathbf{y}^*_{\mathbf{m,k-m+1}} & \mathbf{0 \le k < L^*}, \\ \frac{1}{L^*}\sum_{m=1}^{L^*} \mathbf{y}^*_{\mathbf{m,k-m+1}} & \mathbf{L^* \le k < K^*}, \\ \frac{1}{N-k+1}\sum_{m=k-K^*+1}^{N-K^*+1} \mathbf{y}^*_{\mathbf{m,k-m+1}} & \mathbf{K^* \le k < N} \end{cases} \quad (7)$$

where $\mathbf{1 \le i < L}$, $\mathbf{1 \le j \le K, i + j = k + 1, L^* = min(L, K)}$ and $\mathbf{K^* = max(L, K)}$. If $\mathbf{L < K}$, $\mathbf{y^*_{ij} = y_{ij}}$ and $\mathbf{y^*_{ij} = y_{ji}}$; otherwise, the choice $\mathbf{k = 1}$ gives $\mathbf{y_1 = y_{1,1}}$, and for $\mathbf{k = 2}$, $\mathbf{y_2 = (y_{1,2} + y_{2,1})/2}$ (Moreno and Coelho 2018; Osmanzade 2017; Hassani and Thomakos 2010; Golyandina et al., 2001).

3. APPLICATION

In this paper, we aim to determine sea level change along the coasts of the Black Sea using SSA. With the SSA, the trend and seasonal effect on the mean sea level can be determined. The mean sea level time series recorded at the tide gauge stations used in this study are presented in Figure 3. At the 10 tide gauge locations along the coasts of the Black Sea, mean sea level data in the period 1930-2017 showed similar changes.

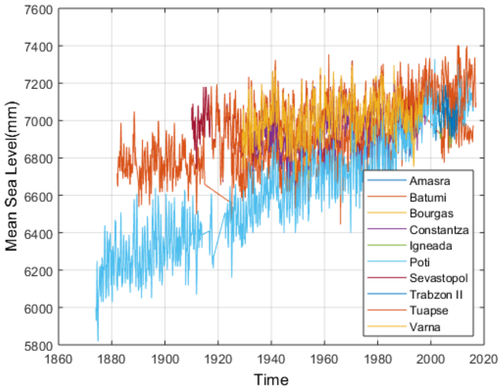


Figure. 3 Tide gauge station time series graphics

The window length (L) is determined for the decomposition stage of the SSA. Determining the window or embedding dimension (L) is one of the most critical steps of this method. The main components decompose better when the L value is large (Hassani et al., 2009). However, there is no strict rule for determining window length. Hassani et al. (2009) suggested that L is selected as the median of the 1,2,…,N values. Golyandina (2010) proposed that L≤N / 2 be selected. In this study, the window length L for each station was selected as N / 2. Thus, we computed the L×L Hankel matrix and L eigentriples.

According to the eigenvalues, we observed that there are close eigenvalue groups. The close eigenvalues of the stations are identified as harmonic components. At the same time, the eigenvalues are close to each other, showing the presence of a seasonal component in the time series. There are close eigenvalues at the Batumi, Constantza, Poti, Sevastapol, Trabzon II, Tuapse and Varna stations, as presented in Figure 4. The first harmonic eigenvectors are shown in Table 2.

Table 2 Harmonic eigenvectors

Station Name	Harmonic eigenvectors
Batumi	6-7
Constantza	5-6, 15-16,17-18,21-22
Poti	3-4
Sevastapol	1-2, 7-8,9-10
Trabzon II	7-8, 10-11
Tuapse	5-6
Varna	21-22, 24-25, 28-29

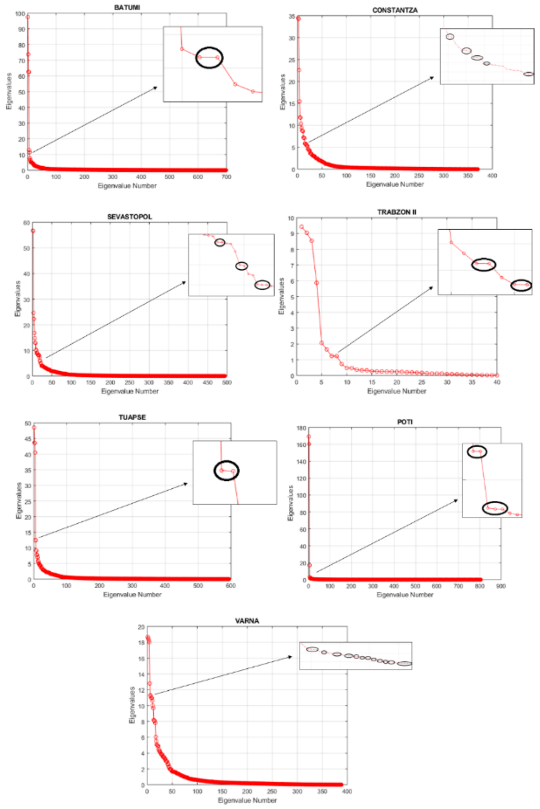


Figure. 4 Eigenvalues of Batumi, Constantza, Poti, Sevastapol, Trabzon II, Tuapse and Varna stations.

Then, the reconstruction stage is achieved using Eq. (5), Eq. (6) and Eq. (7). The difference trend solutions are computed as a result of different combinations of eigentriples. For this, the eigentriples must be selected, as they are the best representation of the data. In this study, the first five reconstruction components (RCs) were selected for each station. Because, according to the RCs graphics, the first five RCs contain practically all trend and seasonal components of time series. RCs are shown in the time series graphics (Figure 5). The RCs that best represented the trend were found in the time series for each station.

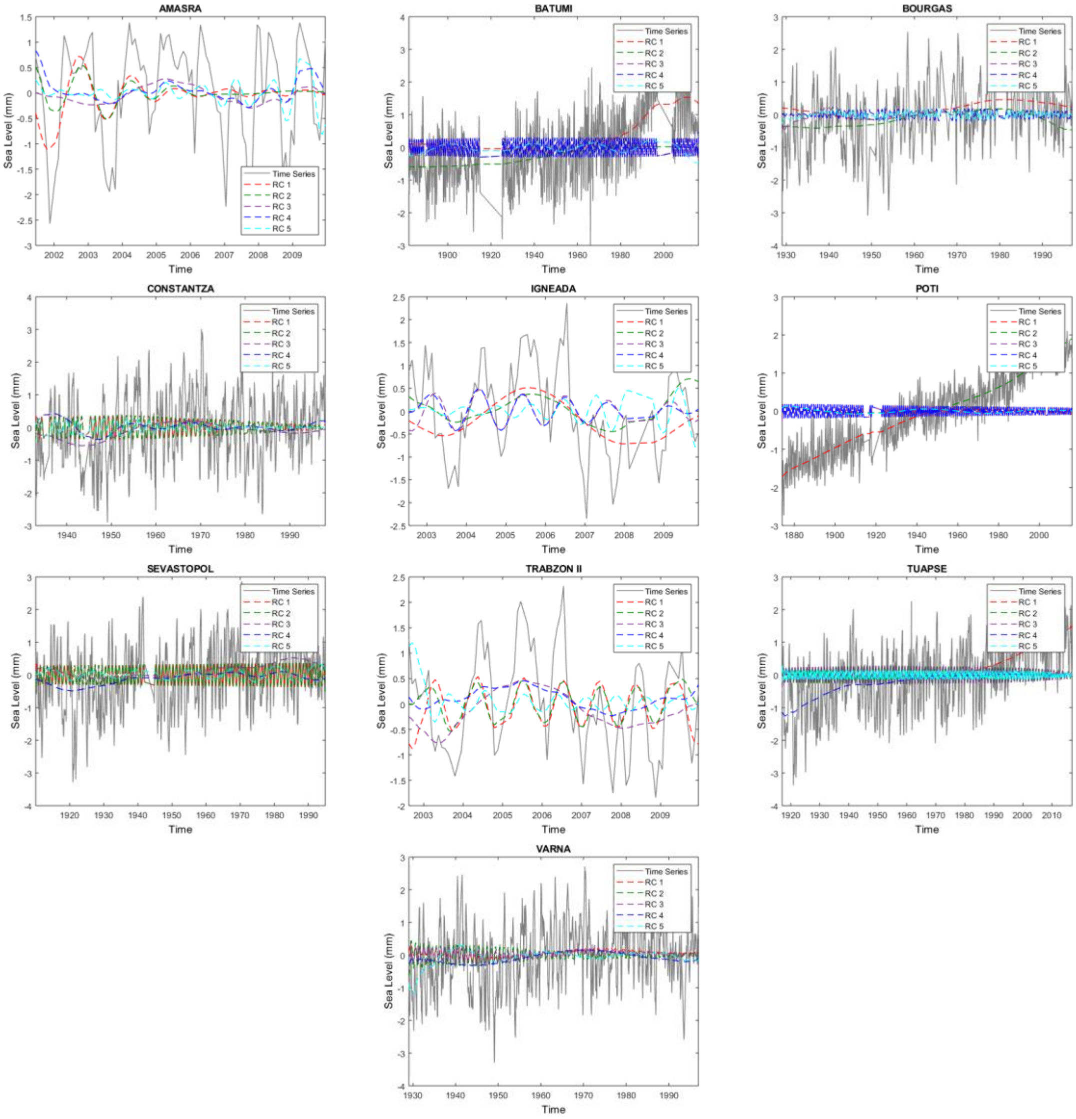


Figure. 5 Reconstruction components of the tide gauges

The slowly varying eigenvalues present the trend (Hassani, 2007).

It is realized that the sum of the reconstructed components present initial time series. The sum of the reconstruction components that best represent the trend for each station is shown on the time series graphics (Figure 6, Figure 7 and Figure 9).

At the Batumi, Poti and Tuapse stations, most data records have increasing trends in the mean sea level. The trends show that the mean sea level at Batumi station increased from -0.11 mm to 0.85 mm during the period of 1882-2015, that at the Poti station increased from -1.62 mm to 1.82 mm during the period of 1874-2015, and that at the Tuapse station increased from -1.09 mm to 1.39 mm during the period of 1917-2017. The results are presented in Figure 6.

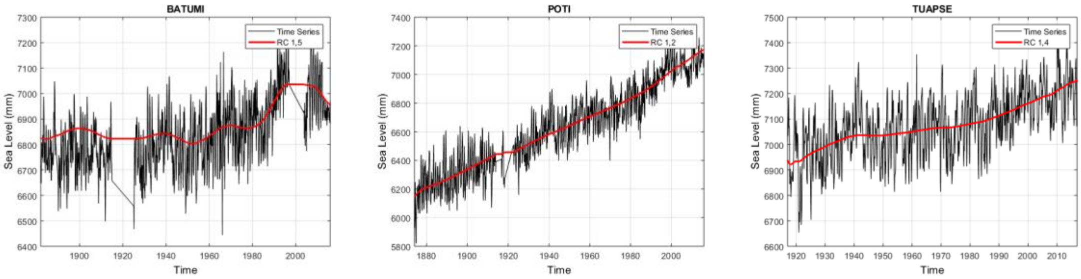


Figure. 6 Batumi, Poti and Tuapse station time series and reconstruction components

The mean sea level at the Bourgas station has no significant change from -0.13 mm to -0.21 mm during the period of 1929-1996. That at the Constantza station increased from -0.06 mm to 0.14 mm during the period of 1933-1997, and that at the Sevastopol station increased from 0.002 mm to 0.17 mm during the period of 1910-1994. The mean sea level at the Varna station changed from -1.10 mm to -0.16 mm during the period of 1929-1996. The results are presented in Figure 7. The increasing trend stations are shown in Figure 8.

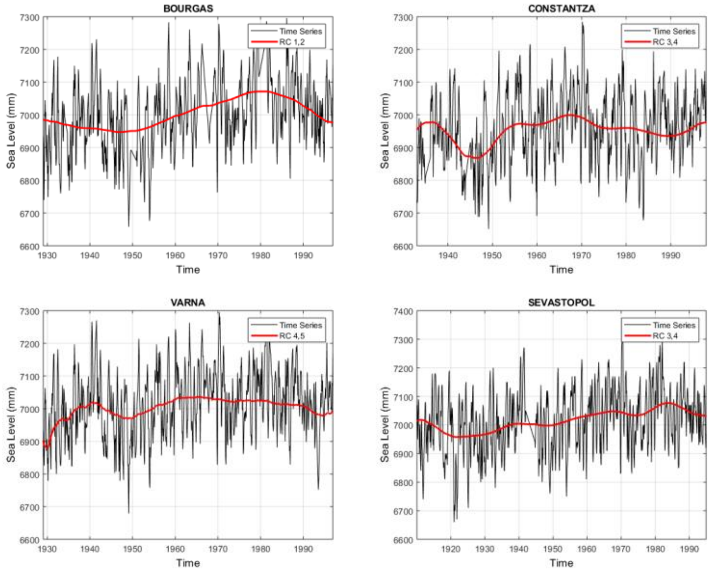


Figure. 7 Bourgas, Constantza, Sevastopol and Varna station time series and reconstruction components

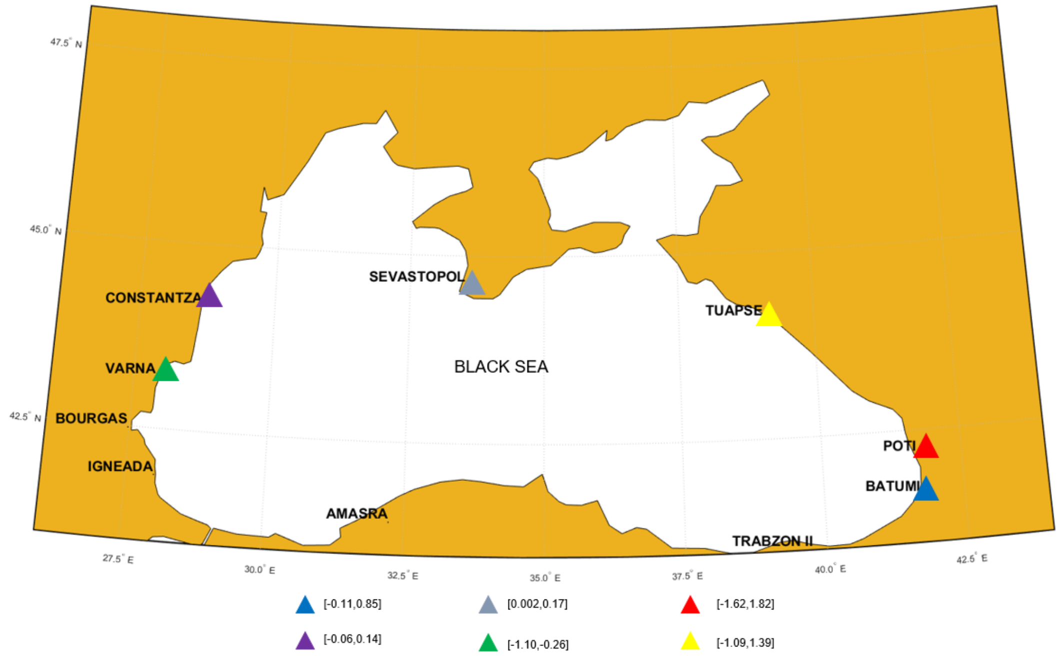


Figure. 8 Increasing trend stations in the Black Sea coasts (Δ denotes increasing trends)

The Amasra, Igneada and Trabzon II tide gauge stations have inadequate data records. These stations have short time series. Thus, the mean sea level change could not be clearly determined. The mean sea level at the Amasra station changed from 0.44 mm to 0.02 mm during the period of 2001-2009, that at the Igneada station changed from 0.11 mm to 0.54 mm during the period of 2002-2009, and that at the Trabzon-II station changed from -0.10 mm to 0.37 mm during the period of 2002-2009 (Figure 9).

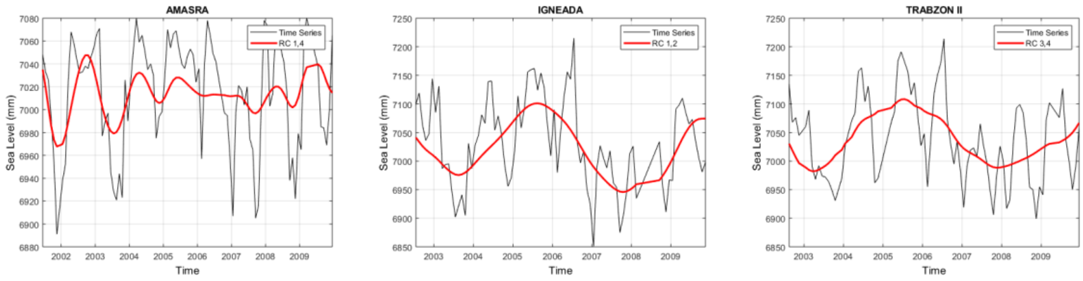


Figure. 9 Amasra, Igneada and Trabzon II station time series and reconstruction components

4. CONCLUSION

In this study, we used the SSA method to detect mean sea level change at 10 tide gauge stations along the coasts of the Black Sea. In addition, we detected the seasonal influence in the time series. To this end, we discuss the results of the analysis.

In the analysis results, there are increasing trends at the Batumi (Georgia), Poti (Georgia), Tuapse (Russia), Constantza (Romania), Sevastapol (Ukraine) and Varna (Bulgaria) stations.

The Amasra, Igneada and Trabzon II tide gauge stations in Turkey were inadequate in interpreting this change. The analysis results are shown in Table 3.

Table 3 Trends at the tide gauge stations

Station	Trend (mm)	Mean	Mean Square Error (mm)	Data Record Span (year)
Batumi	[-0.11, 0.85]	0.26	0.50	133
Constantza	[-0.06, 0.14]	-0.08	0.30	64
Sevastopol	[0.02, 0.17]	0.02	0.31	64
Varna	[-1.10, -0.26]	-0.14	0.28	67
Poti	[-1.62, 1.82]	0.005	0.93	141
Tuapse	[-1.09, 1.39]	0.004	0.62	100
Igneada	[0.11, 0.54]	-0.12	0.60	7
Amasra	[0.44, 0.02]	-0.02	0.35	8
Trabzon II	[-0.10, 0.37]	-0.04	0.51	7
Bourgas	[-0.13, -0.21]	0.002	0.37	67

The results showed that for the Batumi (Georgia) station, the station values of the trend in the 133-year period are from -0.11 mm to 0.85 mm, and for the Poti (Georgia) station, the station values of the trend in the 141-year period are from -1.62 mm to 1.82 mm. For the Bourgas (Bulgaria) and Varna (Bulgaria) stations, the trend values for the same period are from -0.13 mm to -0.21 and from -1.10 mm to -0.26 mm, respectively.

If the eigenvalues are close to each other, this situation showing the presence of a seasonal component in the time series. According to the eigenvalue results, the Batumi, Constantza, Sevastopol, Poti, Tuapse, Trabzon II and Varna stations have close eigenvalues. This situation indicates the presence of a seasonal component in the time series of related stations. No seasonal components found at the other stations (Amasra, Igneada and Bourgas).

ACKNOWLEDGEMENTS

The authors are grateful to Permanent Service for Mean Sea Level (PSMSL) for providing the mean sea level data.

REFERENCES

Avşar N.B., Kutoglu S.H., Erol B., Jin S.G. (2015). Sea level changes in the Black Sea using satellite altimetry and tide-gauge observations. Proceedings of the 26th IUGG General Assembly, Prag, Czech Republic.

Cazenave A., Dieng H.B., Meyssignac B., von Schuckmann K., Decharme B. (2014). The rate of sea-level rise. Nature Climate Change, 4(5), pp.358–361.

Ghil M., Allen R. M., Dettinger M.D., Ide K., Kondrashov D., Mann M.E., Robertson A.W., Saunders A., Tian Y., Varadi F. and Yiou P. (2002). Advanced spectral methods for climatic time series. Reviews of Geophysics, 40, pp. 1-41.

Golyandina N. (2010). On the choice of parameters in singular spectrum analysis and related subspace-based methods. Statistics and Its Interface, 3,pp. 259-279.

Golyandina N., Nekrutkin V., Zhigljavsky A. (2001). Analysis of time series structure: SSA and related techniques. Chapman & Hall/CRC, Boca Raton.

Golyandina N., Zhigljavsky, A. (2013). Singular spectrum analysis for time series. Springer, doi: 10.1007/978-3-642-34913-3.

Goriacikin I.N., Ivanov V.A. (2006). Black Sea Level: Past, Present, Future, (in Russian). EKOCI-Gidrofizica, Sevastopol,210.

Hassani H. (2007). Singular spectrum analysis: methodology and comparison. Journal of Data Science, 5(2), pp.239-257.

Hassani H., Heravi S., Zhigljavsky A. (2009). Forecasting European industrial production with singular spectrum analysis. International Journal of Forecasting, 25(1),pp. 103-118.

Hassani H., Thomakos D. (2010). A review on singular spectrum analysis for economic and financial time series. Statistics and Its Interface, 3, pp. 377-397.

Karaca M., Ünal Y.S. (2003). Potential ımplications of accelerated sea-level rise for Turkey. Journal Coastal Research, 24, pp. 288-298.

Khelifa S., Gourine B., Rami A., Taibi H. (2016). Assessment of nonlinear trends and seasonal variations in global sea level using singular spectrum analysis and wavelet multiresolution analysis. Arabian Journal of Geosciences, 9:560, doi: 10.1007/s12517-016-2584-6.

Kubryakov A.A., Stanichny S. (2013). The Black Sea level trends from tide gages and satellite altimetry. Russian Meteorology and Hydrology, 38(5), doi: 10.3103/S1068373913050051.

Moreno S.R., Coelho L.D.S. (2018). Wind speed forecasting approach based on singular spectrum analysis and adaptive neuro fuzzy ınference system. Renewable Energy, 126, pp. 736-754.

Nacef L., Bachari N.E.I., Bouda A. and Boubnia R. (2016). Variability and decadal evolution of temperature and salinity in the Mediterranean sea surface. International Journal of Engineering and Geosciences, 1, pp. 24-33.

Osmanzade A. (2017). Singular spectrum analysis forecasting for financial time series. Master Thesis, University of Tartu, Tartu.

Yilmaz M., Turgut B., Gullu M., Yilmaz I. (2016). Evaluation of recent global geopotential models by GNSS/Levelling data: internal Aegean region. International Journal of Engineering and Geosciences,1, pp.18-223.

URL-1
https://tudes.harita.gov.tr/tudesportal/Hakkında.aspc.

URL 2- https://www.psmsl.org/data/

Vigo I., Garcia D., Chao B.F. (2005). Change of sea level trend in the Mediterranean and Black seas. Journal of Marine Research, 63, pp. 1085–1100.

International Journal of Engineering and Geosciences (IJEG),
Vol; 5, Issue; 1, pp. 042-048, February, 2020, ISSN 2548-0960, Turkey,
DOI: 10.26833/ijeg.581568

THE GLOBAL AND LOCAL ROBUSTNESS ANALYSIS IN GEODETIC NETWORKS

Pakize Küreç Nehbit [1]*, Haluk Konak[2]

[1]Kocaeli University, Faculty of Engineering, Department of Geomatic Engineering, Kocaeli, Turkey
(pkurec80@yahoo.com); **ORCID 0000-0003-2669-0401**
[2]Kocaeli University, Institutes of Natural and Sciences, Department of Geodesy end Geoinformation, Kocaeli, Turkey
(konak_haluk@yahoo.com); **ORCID 0000-0003-2554-8905**

*Corresponding Author, Received: 24/06/2019, Accepted: 20/07/2019

ABSTRACT: *The geodetic networks should be able to detect the possible earth crust movements caused by active tectonic movements in Turkey. Geodetic networks should also be able to determine the crust movements accurately as well as provide precision and reliability requests. The capacity of geodetic networks to determine the crustal movements can be determined by sensitivity analysis. Robustness analysis consists of strengthening internal reliability analysis with strain techniques. Robustness is defined as the deformation strength induced by the maximum undetectable errors with the internal reliability analysis. The robustness of a geodetic network is determined by the global initial condition, which aims at minimizing the total displacement value in the network. In this study, the local initial condition that aims to minimize the total displacement value at the uniform polyhedron, which consists of observations from each station points. The displacement values obtained according to the local and the global initial conditions are compared with threshold values. The results have also been interpreted.*

Keywords: *Sensitivity, Strain, Reliability, Geodetic Networks, Robustness.*

1. INTRODUCTION

Geodetic networks established for scientific or engineering purposes are expected to be robust enough to detect local deformations or tectonic movements of the region. For this reason, it is highly important that the sensitivity levels and the robustness are integrally queried when the quality of the geodetic networks is investigated. In geodetic networks, robustness is obtained from a function of reliability criteria. Even if the reliability criteria is sufficiently within the required limit values, in the geodetic networks, an outlier at an observation affects coordinate unknowns of each station point and leads to a different magnitude of deformation. Also, the station points are stretched in a different direction and ratio with the effect of an outlier at any observations. Therefore, investigation of the reliability of the geodetic networks can be considered as research of the external reliability vector which corresponds to the observation leading to the greatest strain at a station point (Vanicek et al., 1990; Berber, 2006; Konak, 2018). The external reliability vector depends on the number of datum points in free network solutions and distribution in the network. For this reason, the external reliability vector which causes the greatest strain can be defined as the vector having a maximum vector norm. In this case, the magnitude of the largest strain components is obtained independently of the datum using the strain models representing the surface formed by observation links of each station point. The estimated strain components for each station point are independent of the translation components of the strain area. However, while determining the displacement value of station points in the network, the effects of the translation components must also be eliminated. This process is accomplished with a global initial condition that aims to have a minimum total displacement magnitude. The corrected global displacement magnitudes are compared with the threshold value estimated from the confidence ellipsoids.

The robustness analysis has been first defined by Vanicek et al. (1990). In the study, an alternative geodetic network analysis to standard statistical analysis techniques has been developed to investigate the sensitivity of geodetic networks against outliers. The fundamental part of this alternative analysis, called Reliability Analysis, was introduced by Baarda (1976). This investigation process which was developed by a group of researchers at the University of New Brunswick was published as The Geometrical Strength Analysis. The differences between the reliability analysis and the benefits of the geometrical strength analysis are the subject of the study. Then, it was discovered that both of the analysis techniques are complementary to each other. The combined analysis method which consists of the reliability analysis and the geometrical strength analysis is called Robustness Analysis.

Basic deformation parameters are defined as relative translation, relative rotation, strain tensor and differential rotation components by Kuang (1991). Different strain models are explained according to the deformation model defined on the surface represented by any geodetic network with this study. The different mathematical models are proposed according to the homogeneous deformation of a whole surface and to the heterogeneous deformation explained with the movement of a surface relative to different centers of gravity by Kuang.

The displacement magnitude derived from the effects of the outliers on the coordinate unknowns is obtained independently according to the translation by Berber (2006). Therefore, it is aimed to determine a global initial condition for all station points of the network. These initial conditions are computed separately for 3D, 2D, and 1D geodetic networks. Displacement magnitudes are also obtained as 3D, 2D, and 1D.

In this study, local initial conditions aimed at minimizing the total displacement and developed for the polyhedron represented by each network point are proposed.

Global displacement magnitudes represent all station points of the network; however, local displacement magnitudes represent a surface formed by observation links of each neighbouring station point. The displacement vectors are comparable to the minimum undetectable displacement value (the sensitivity level) due to representing the effects of the undetectable outliers on the coordinate unknowns. For this reason, it is recommended that the local and the global displacement magnitudes computed for each station point should be compared with both the threshold values estimated from the confidence ellipsoid and the sensitivity levels suggested as a different threshold value (Küreç Nehbit, 2018).

2. STRAIN IN GEODETIC NETWORKS

Any tectonic plate or surface area can be identified as a kind of material or as a homogeneous object, despite its natural structures. When a force is applied to the surface, the resistance of the surface to this force can be determined mathematically and defined according to a coordinate system. The determination of the deformations on a surface in a coordinate system, independent from the datum and the geometric interpretation are explained by the concept of strain (Chou and Pagano, 1992; Konak, 2018; Küreç Nehbit, 2018).

Strain in the geodetic networks is defined as the ratio of the change in coordinate axies to the initial coordinate system. In other words, the strain is a geodetic/geometric interpretation of the deformation on a surface. Strains in the geodetic networks can be caused by the internal structure of the networks or by external factors. The strains arising from the internal structure of the network are affected by observation weights and observation plan. The strains due to external factors occur because of tectonic movements and local deformations (Küreç Nehbit, 2018).

Different mathematical methods are used to compute the strain components depending on the homogenous or heterogeneous deformation models defined on the surface. In the homogeneous deformation models, the deformation on a surface is equal in each region of the surface. On the other hand, in the heterogeneous deformation models, the deformations on a surface are different in every region of the surface. In this case, if the surface has a homogeneous deformation, the strain is described as a homogeneous strain. Also, if the surface has a heterogeneous deformation, the strain is identified as a heterogeneous strain (Vanicek et al., 1990; Kuang, 1991; Poyraz, 2009; Küreç Nehbit, 2018).

In geodetic networks measured at different times, the deformation vectors can be established according to the predicted deformation model for the displacement vector between two periods. The deformation model is selected according to a priori knowledge (Kuang, 1991; Küreç Nehbit, 2018).

At any station points, the displacement vector as a strain relationship can be written with the following equation;

$$\Delta x_i = E_i x_i + c_0 \tag{1}$$

If the displacement vector is obtained in three dimensions, the strain matrix (Ei) is identified as;

$$E_i = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} = \begin{bmatrix} e_{xx} & e_{xy} & e_{xz} \\ e_{yx} & e_{yy} & e_{yz} \\ e_{zx} & e_{zy} & e_{zz} \end{bmatrix} \tag{2}$$

(Vanicek et al., 1990; Vanicek et al., 2001; Konak, 2010; Küreç 2010; Küreç Nehbit, 2018).

2.1 Surface Model Approach

Strain components could be obtained from a function of coordinate differences or from velocity information. The mathematical models representing triangular or polygonal surfaces could be defined with Affine or the extended Helmert transformation (Konak et al., 2017; Küreç Nehbit, 2018; Öcalan, 2018).

The transformed coordinates according to the reference point are obtained as;

$$\Delta x_i = X_i - X_0 \tag{3a}$$
$$\Delta y_i = Y_i - Y_0 \tag{3b}$$
$$\Delta z_i = Z_i - Z_0 \tag{3c}$$

and using this transformed coordinates' differences, translation equations are written for each network point on a surface, where the strain components to be computed,

$$v_{x_{k+1}} = x_{k+1} - x_k = t_x + \frac{\partial u}{\partial x}\Delta x_i + \frac{\partial u}{\partial y}\Delta y_i + \frac{\partial u}{\partial z}\Delta z_i \tag{4a}$$
$$v_{y_{k+1}} = y_{k+1} - y_k = t_y + \frac{\partial v}{\partial x}\Delta x_i + \frac{\partial v}{\partial y}\Delta y_i + \frac{\partial v}{\partial z}\Delta z_i \tag{4b}$$
$$v_{z_{k+1}} = z_{k+1} - z_k = t_z + \frac{\partial w}{\partial x}\Delta x_i + \frac{\partial w}{\partial y}\Delta y_i + \frac{\partial w}{\partial z}\Delta z_i \tag{4b}$$
$$d_i = H_i . g \tag{5}$$

where (k) is epoch number, (P0) is reference points, (di) is displacement vector, (Hi) is the design matrix, (g) is the vector of the strain components. 3D affine transformation matrix representing homogeneous strain properties is defined as;

$$H_i = \begin{bmatrix} 1 & 0 & 0 & \Delta x_i & \Delta y_i & \Delta z_i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \Delta x_i & \Delta y_i & \Delta z_i & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta x_i & \Delta y_i & \Delta z_i \end{bmatrix} \tag{6}$$

and the displacement vector is obtained as;

$$d_i^T = [v_x \quad v_y \quad v_z] \tag{7}$$

In this way, Affine Model is formulated (Kuang, 1991). In this case, strain components with 3D are determined with the following equations;

$$g = (H^T H)^{-1} H^T d \tag{8}$$

$$g^T = [t_x \quad t_y \quad t_z \; e_{xx} \quad e_{yx} \quad e_{zx} \; e_{xy} \quad e_{yy} \quad e_{zy} \; e_{xz} \quad e_{yz} \quad e_{zz}] \tag{9}$$

3. ROBUSTNESS ANALYSES FOR THE GEODETIC NETWORKS

Robustness is defined as the deformation strength caused by the undetectable possible model error with the internal reliability analyses. Robustness analyses consist of strengthening internal reliability analyses with strain techniques. In any observation, the effect of outliers on the coordinate unknowns is obtained as;

$$\Delta X = Q A^T P \Delta_{0i} \tag{10}$$

$$\delta^{\,2}_{0i} = \Delta X^T K_{xx}^{-1} \Delta X \tag{11}$$

and is determined for experimental observation as;

$$\delta^{\,2}_{0i} = m_0^2 \frac{\delta_0}{e_i^T P Q_{vv} P e_i} \tag{12}$$

This squared magnitude called external reliability is a criterion independent from the selection of datum points in free networks. On the other hand, the effect of the outliers, which are interpreted as the displacement vector, on the coordinate unknowns are identified with the following equations;

$$\Delta X^T = [\Delta X_1; \quad \Delta X_2; \quad \ldots \quad \ldots \quad \ldots; \quad \Delta X_u] \tag{13}$$

$$\Delta X_{\,i} = \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} \tag{14}$$

The vector magnitude, which can be computed as much as the number of observations for any coordinate unknown, is stretched in different magnitude and directions depending on the observation weights. In this case, this vector magnitude ($\mathbf{\Delta X_{0i}}$);

$$\boldsymbol{\Delta X_{0i}} = \boldsymbol{max}\{|\boldsymbol{\Delta X_i}|\} \tag{15}$$

causing the greatest strain must be queried. The observation that causes the greatest strain is assumed to be the observation with the largest vector norm. To determine the largest vector norm, either L1 norm, Euclidean (L2) norm or Weighted Euclidean norms could be used. The most appropriate vector norm that could be compared with the external reliability criterion is the Euclidean (L2) norm.

$$\|\Delta x\| = \sqrt{(\Delta X_1)^2 + (\Delta X_2)^2 + \cdots \ldots \ldots . + (\Delta X_u)^2} \tag{16}$$

When determining the largest vector norms; if there is any equality between the Euclidean norms of the observations, the observation which has the largest external reliability value should be chosen (δ_{0i}; Weighted Euclidean norms).

The strain resulting from the effect of undetectable errors on the coordinate unknowns is obtained using the Affine or the extended Helmert transformation model for the surface representing each station point.

The strains are independent of the location of the surface (initial conditions, X0, Y0, Z0) in a coordinate system. In this case, the strain tensor matrix Ei with respect to a selected reference point P0 on the surface is obtained by the following equation;

$$\begin{bmatrix}\Delta x_i\\ \Delta y_i\\ \Delta z_i\end{bmatrix}=\begin{bmatrix}e_{xx} & e_{xy} & e_{xz}\\ e_{yx} & e_{yy} & e_{yz}\\ e_{zx} & e_{zy} & e_{zz}\end{bmatrix}\begin{bmatrix}X_i-X_0\\ Y_i-Y_0\\ Z_i-Z_0\end{bmatrix} \qquad (17)$$

3.1 Determination of Deformation Vectors

Robustness analysis procedures are based on determining the magnitude of the deformation vector leading to strain at any station point and investigating the significance level. In this case, an initial condition (P0) representing the network is required in order to calculate a deformation vector at each point according to equation (17). This value is the estimation value that "makes minimum squares of corrections brought to the center of gravity coordinates of the polyhedron representing each point" (Berber, 2006; Küreç Nehbit, 2018).

To determine the initial conditions/translation parameters, the norm of the displacement vector elements at all network points is intended to be minimum. The objective function is arranged as following;

$$min\{\textstyle\sum_{i=1}^{n}\|\Delta\vec{r}\|_i\}=min\{\textstyle\sum_{i=1}^{n}\left(u_i^2+v_i^2+w_i^2\right)\} \qquad (18)$$

where (n) is the number of station points in the network. The objective function is linearized separately according to the initial conditions (X0, Y0, Z0). Linearization of the objective function according to (X0);

$$\textstyle\sum_{i=1}^{n}\begin{bmatrix}\left(-\frac{\partial u_i}{\partial x}\frac{\partial u_i}{\partial x}-\frac{\partial v_i}{\partial x}\frac{\partial v_i}{\partial x}-\frac{\partial w_i}{\partial x}\frac{\partial w_i}{\partial x}\right)X_0+\left(-\frac{\partial u_i}{\partial y}\frac{\partial u_i}{\partial x}-\frac{\partial v_i}{\partial y}\frac{\partial v_i}{\partial x}-\frac{\partial w_i}{\partial y}\frac{\partial w_i}{\partial x}\right)Y_0+\\ \left(-\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial x}-\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial x}-\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial x}\right)Z_0+\left(\frac{\partial u_i}{\partial x}\frac{\partial u_i}{\partial x}+\frac{\partial v_i}{\partial x}\frac{\partial v_i}{\partial x}+\frac{\partial w_i}{\partial x}\frac{\partial w_i}{\partial x}\right)X_i+\\ \left(\frac{\partial u_i}{\partial y}\frac{\partial u_i}{\partial x}+\frac{\partial v_i}{\partial y}\frac{\partial v_i}{\partial x}+\frac{\partial w_i}{\partial y}\frac{\partial w_i}{\partial x}\right)Y_i+\left(\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial x}+\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial x}+\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial x}\right)Z_i\end{bmatrix} \qquad (19a)$$

$$\frac{\partial\sum_{i=1}^{n}\|\Delta\vec{r}\|_i}{\partial X_0}=\textstyle\sum_{i=1}^{n}[a_1X_0+b_1Y_0+c_1Z_0+d_1]=0 \qquad (19b)$$

linearization of the objective function according to (Y0);

$$\textstyle\sum_{i=1}^{n}\begin{bmatrix}\left(-\frac{\partial u_i}{\partial y}\frac{\partial u_i}{\partial x}-\frac{\partial v_i}{\partial y}\frac{\partial v_i}{\partial x}-\frac{\partial w_i}{\partial y}\frac{\partial w_i}{\partial x}\right)X_0+\left(-\frac{\partial u_i}{\partial y}\frac{\partial u_i}{\partial y}-\frac{\partial v_i}{\partial y}\frac{\partial v_i}{\partial y}-\frac{\partial w_i}{\partial y}\frac{\partial w_i}{\partial y}\right)Y_0+\\ \left(-\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial y}-\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial y}-\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial y}\right)Z_0+\left(\frac{\partial u_i}{\partial x}\frac{\partial u_i}{\partial y}+\frac{\partial v_i}{\partial x}\frac{\partial v_i}{\partial y}+\frac{\partial w_i}{\partial x}\frac{\partial w_i}{\partial y}\right)X_i+\\ \left(\frac{\partial u_i}{\partial y}\frac{\partial u_i}{\partial y}+\frac{\partial v_i}{\partial y}\frac{\partial v_i}{\partial y}+\frac{\partial w_i}{\partial y}\frac{\partial w_i}{\partial y}\right)Y_i+\left(\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial y}+\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial y}+\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial y}\right)Z_i\end{bmatrix} \qquad (20a)$$

$$\frac{\partial\sum_{i=1}^{n}\|\Delta\vec{r}\|_i}{\partial Y_0}=\textstyle\sum_{i=1}^{n}[a_2X_0+b_2Y_0+c_2Z_0+d_2]=0 \qquad (20b)$$

and linearization of the objective function according to (Z0);

$$\textstyle\sum_{i=1}^{n}\begin{bmatrix}\left(-\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial x}-\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial x}-\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial x}\right)X_0+\left(-\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial y}-\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial y}-\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial y}\right)Y_0+\\ \left(-\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial z}-\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial z}-\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial z}\right)Z_0+\left(\frac{\partial u_i}{\partial x}\frac{\partial u_i}{\partial z}+\frac{\partial v_i}{\partial x}\frac{\partial v_i}{\partial z}+\frac{\partial w_i}{\partial x}\frac{\partial w_i}{\partial z}\right)X_i+\\ \left(\frac{\partial u_i}{\partial y}\frac{\partial u_i}{\partial z}+\frac{\partial v_i}{\partial y}\frac{\partial v_i}{\partial z}+\frac{\partial w_i}{\partial y}\frac{\partial w_i}{\partial z}\right)Y_i+\left(\frac{\partial u_i}{\partial z}\frac{\partial u_i}{\partial z}+\frac{\partial v_i}{\partial z}\frac{\partial v_i}{\partial z}+\frac{\partial w_i}{\partial z}\frac{\partial w_i}{\partial z}\right)Z_i\end{bmatrix} \qquad (21a)$$

$$\frac{\partial\sum_{i=1}^{n}\|\Delta\vec{r}\|_i}{\partial Z_0}=\textstyle\sum_{i=1}^{n}[a_3X_0+b_3Y_0+c_3Z_0+d_3]=0 \qquad (21b)$$

The objective function given by equation (18) is arranged with the following equation;

$$\boldsymbol{\textstyle\sum_{i=1}^{n}}(\Delta \boldsymbol{r}^{\boldsymbol{T}}\Delta r)_i=\boldsymbol{\textstyle\sum_{i=1}^{n}}(\boldsymbol{X_i}-\boldsymbol{X_0})^{\boldsymbol{T}}\boldsymbol{E_i^T}\,\boldsymbol{E_i}(\boldsymbol{X_i}-\boldsymbol{X_0})\Rightarrow \boldsymbol{min} \qquad (22)$$

The objective function is rewritten as;

$$\textstyle\sum_{i=1}^{n}(\Delta r^T\Delta r)_i=\sum_{i=1}^{n}(\Delta x)^TE_i^TE_i(\Delta x)\Rightarrow min \qquad (23)$$

Also, it is linearized according to the initial conditions. In this case, normal equations are obtained by linearizing;

$$\boldsymbol{d\{\textstyle\sum_{i=1}^{n}(\Delta r^T\Delta r)_i\}=0} \qquad (24)$$

$$\boldsymbol{-\textstyle\sum_{i=1}^{n}E_i^TE_iX_0+\sum_{i=1}^{n}E_i^TE_iX_i=0} \qquad (25)$$

The initial conditions $\boldsymbol{X_0^T}=[\boldsymbol{X_0},\boldsymbol{Y_0},\boldsymbol{Z_0}]$ are computed with the following equation;

$$\boldsymbol{X_0}=\left[\boldsymbol{\textstyle\sum_{i=1}^{n}E_i^TE_i}\right]^{\boldsymbol{-1}}\boldsymbol{\textstyle\sum_{i=1}^{n}E_i^TE_iX_i} \qquad (26)$$

(Berber, 2006; Konak, 2018; Küreç Nehbit, 2018). The solution vector in Eq. (26) could be shown as follows;

$$\begin{bmatrix}X_0\\ Y_0\\ Z_0\end{bmatrix}=\begin{bmatrix}[a_1] & [b_1] & [c_1]\\ [a_2] & [b_2] & [c_2]\\ [a_3] & [b_3] & [c_3]\end{bmatrix}^{-1}\begin{bmatrix}[d_1]\\ [d_2]\\ [d_3]\end{bmatrix} \qquad (27)$$

If the obtained initial conditions are written in the equation (17), the corrected global displacement vector is obtained.

Translation value of the global displacement vector (corrected global displacement vector) relative to the gravity center is obtained as;

$$\boldsymbol{d_{Gi}}=\sqrt{\boldsymbol{u_{Gi}^2}+\boldsymbol{v_{Gi}^2}+\boldsymbol{w_{Gi}^2}} \qquad (28)$$

The corrected global displacement vector could also be estimated from the surface represented by each station point instead of the whole network. In this context, local initial conditions can be estimated which aim at minimizing the total displacement for the polyhedron represented by each network station point, with a new approach developed. The strains computed for each station point represent the surface which consists of observations of each station point. In this case, corrected local initial conditions are determined for each surface area with the following equations;

$$\boldsymbol{-m(E^TE)_iX_{L0}+m(E^TE)_iX_i=0} \qquad (29)$$

$$\boldsymbol{X_{L0}=(E^TE)_i^{-1}E_i^TE_i\textstyle\sum_{i=1}^{m}X_i} \qquad (30)$$

where (m) is station points number on surface. Using the computed local initial conditions;

$$\begin{bmatrix}\boldsymbol{u}\\ \boldsymbol{v}\\ \boldsymbol{w}\end{bmatrix}_{\boldsymbol{Li}}=\boldsymbol{E_i}\begin{bmatrix}\boldsymbol{X-X_{L0}}\\ \boldsymbol{Y-Y_{L0}}\\ \boldsymbol{Z-Z_{L0}}\end{bmatrix}_{\boldsymbol{i}} \qquad (31)$$

the local displacement vector magnitude (d_{Li}) is computed as;

$$d_{Li}=\sqrt{u_{Li}^2+v_{Li}^2+w_{Li}^2} \qquad (32)$$

The displacement magnitudes obtained as the local and the global are compared with the threshold value (δ_i) estimated from the confidence ellipsoid;

$$\boldsymbol{\delta_i = m_0\sqrt{3.F_{h,f,1-\alpha}.iz(Q_{xx})}} \tag{33}$$

In the case of $d_{Gi} > \delta_i$ and of $d_{Li} > \delta_i$, it can be said that the network station point discussed is not robust (not sufficiently reliable) in respect of the global and the local (Berber, 2006; Küreç Nehbit, 2018).

It is expected that a network is insensitive to possible outliers and the disruptive effect of the outliers (strains, deformation vectors) on the coordinate unknowns are as low as possible. On the other hand, the networks should be able to detect the negative influence of these effects on the displacement vector sufficiently. In other words, the more robust a network is, the more sensitive it is to outliers in observations or changes over time.

Therefore, the sensitivity and robustness distributions should be evaluated and interpreted together. Because of the displacement vectors obtained at any station point in the network represent the external reliability values, these are also comparable with the sensitivity values.

In this case, the inequation of non-centrality distribution is recommended as a threshold value instead of the equation (34) for the displacement vectors at each network point;

$$\frac{d^T Q_{dd}^{-1} d}{\sigma^2} \leq \delta_0^2 \tag{34}$$

where d is the displacement vector, Q_{dd} is cofactor matrix of the displacement vector, δ_0 is the threshold value of the non-centrality parameter, σ^2 is a priori variance of the average error of the unit observation. In equation (34), the quadratic value of the displacement vector ($d^T Q_{dd}^{-1} d$) is rearranged according to the eigenvalue and eigenvector separation and orthogonality conditions, and the sensitivity value of each station point is obtained;

$$\|d\|_{min} = \frac{\delta_0 \sigma}{\sqrt{\lambda_{max}}} \tag{35}$$

(Hsu and Hsiao, 2002; Küreç, 2010; Küreç and Konak, 2011 and 2014; Küreç Nehbit, 2018; Kirici Yildirim and Sisman, 2019).

4. NUMERICAL APPLICATION

In this study, the data of IZDOGAP Densification Global Positioning System (GPS) Network established for the Monitoring of IZGAZ Natural Gas Infrastructure with Geodetic Networks and Information System Project (IZDOGAP) is used. Observation plan of the densificated IZDOGAP GPS network has been obtained using the second order weight optimization (Figure 1). The network measured in 2009 and 2010 has been evaluated with obtained GPS observations respectively. Also, the robustness level of the IZDOGAP network has been queried.

Robustness investigations are performed using displacement vectors resulting from the effect of the outliers on the coordinate unknowns. In this investigation process, the displacement magnitude can be obtained as much as the number of observations at each station point. Determining the deformation resistance or strain caused by the observation having the greatest effect on the coordinate unknowns is very important in terms of robustness analysis. Therefore, the observation with the greatest effect on coordinate unknowns should be estimated independently from the datum. Various vector norms have been tested for the estimation processes and it has been decided to use Euclidean norm (L2 norm) which gives an unbiased result. Strain components are computed for Densificated IZDOGAP GPS Network by using displacement vector having the greatest effect on coordinate unknowns determined according to Euclidean norm.

Strain components are obtained in 3D using the adjusted Affine transformation process for the surface representing each station points. The deformation vectors resulting from the strain are estimated for each station point with the obtained strain information. Initial conditions are required for deformation vectors to be estimated as independent of translation and consistent. In this study, initial conditions are obtained by two different approaches: local and global. The corrected displacement vectors for each station point are estimated separately using the local and global initial conditions. In other words, the robustness level of each station point is determined both locally and globally. The deformation values obtained have been compared with the threshold value computed using equation (33) and the significance of the deformation vector is tested.

The displacement vectors obtained for any station point represent the effect of undetectable errors on coordinate unknowns. Therefore, the displacement vector can be interpreted as a value comparable to the sensitivity level (d_{min}). For this reason, computed local and global deformation vectors are compared with the sensitivity levels of each station point (d_{min}) obtained with equation (35) (Table 1).

5. CONCLUSION

When the corrected displacement magnitudes computed using the 2009 and 2010 epoch observations of densified IZDOGAP GPS Network for all network points are examined, it is seen that;

- The magnitude of the displacement vector estimated according to the local approach is generally smaller than estimated values according to the global approach.
- On the other hand, in epoch 2009.370, local displacement vectors at station points 103 and 38 are higher than global displacement vectors. If the locations of these station points are examined in Figure 1, it is observed that they are located in external zone points. In the 2010.496 epoch, the magnitudes of local deformation vectors at station points 73 and 42 are higher than global deformation vectors.
- In the epochs, 2009.370 and 2010.496, the local and global displacement vectors obtained for each station point do not exceed threshold value obtained from the confidence ellipse components.
- As a result, the IZDOGAP GPS Network is robust at all points and the sensitivity values are reliable.

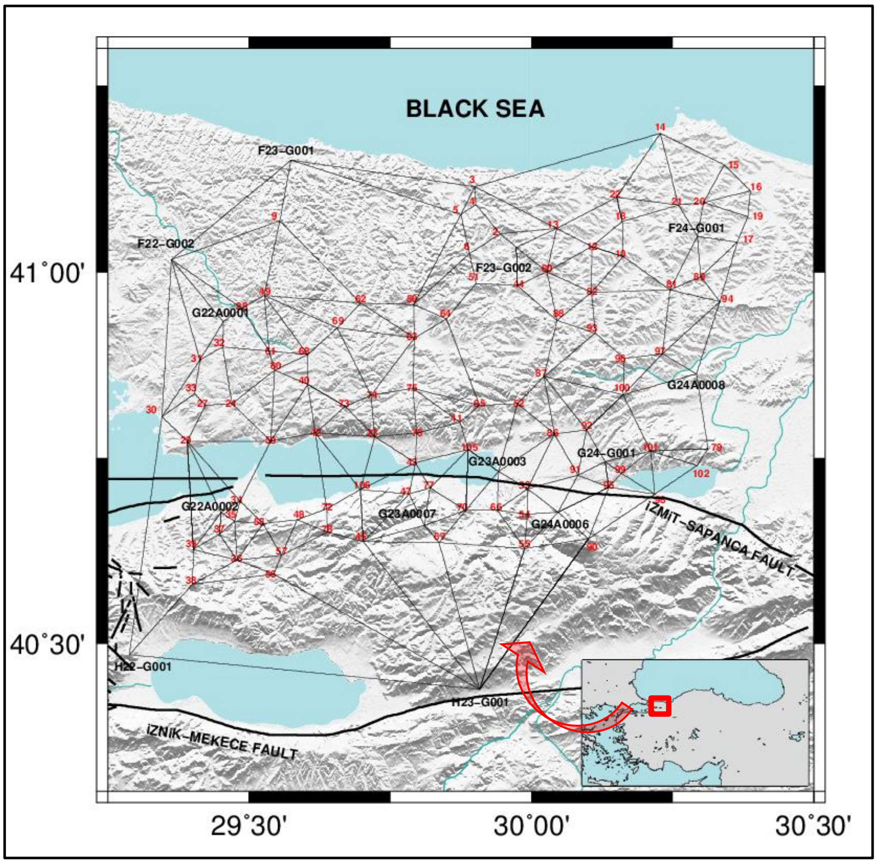


Figure 1. The densified Kocaeli IZDOGAP GPS network

Table 1. A Priori Robustness Synthesis (cm)

	Epoch 2009.370				Epoch 2010.496			
P. Num.	Deformation Vec.		Thresholds		Deformation Vec.		Thresholds	
	Local	Global	$\boldsymbol{\delta_i}$	d_{min} (m_0)	Local	Global	$\boldsymbol{\delta_i}$	d_{min} (m_0)
3	0.0006	0.0133	3.12	1.15	0.0028	0.0216	5.55	1.97
10	0.0005	0.0187	2.98	1.07	0.0018	0.0380	5.69	2.06
14	0.0015	0.1059	4.17	1.57	0.0052	0.2707	8.34	2.61
15	0.0001	0.0070	4.12	1.54	0.0014	0.0355	7.66	2.61
16	0.0002	0.0122	4.04	1.47	0.0003	0.0238	7.22	2.47
19	0.0001	0.0103	3.69	1.37	0.0003	0.0185	7.03	2.42
20	0.0001	0.0255	3.49	1.29	0.0001	0.0386	6.58	2.26
21	0.0004	0.0131	3.32	1.20	0.0015	0.0134	6.48	2.24
30	1.0206	0.0530	3.61	1.18	0.0122	0.0485	7.67	2.73
33	0.0192	0.2043	3.23	1.18	0.0041	0.1323	7.96	2.71
38	1.5879	0.0939	3.63	1.36	0.0097	0.0592	13.94	4.16
56	0.0362	0.3403	4.16	1.05	0.0089	0.1552	8.48	2.90
64	0.0013	0.0818	2.69	0.95	0.0330	0.2971	4.96	1.67
73	0.0021	0.0838	2.34	0.80	7.6174	0.8845	13.09	3.55
74	0.0025	0.0343	2.36	0.79	2.0290	2.4175	6.66	2.24
79	0.0023	0.0411	3.95	1.39	0.0081	0.0454	8.51	3.29
98	0.0048	0.0130	3.37	1.19	0.0020	0.0354	6.30	2.33
99	0.0002	0.0238	2.97	1.04	0.0011	0.0577	6.12	2.21
101	0.0010	0.0100	3.18	1.10	0.0034	0.0174	6.30	2.34
103	5.4968	0.2282	6.49	2.31	0.0143	0.0452	14.46	4.44

External reliability vectors consist of a plurality of sub-vector components close to and equal to zero according to the observation plan of the network. On the other hand, a global displacement vector is estimated for the robustness analysis. As a result, the magnitude of the local and global displacement vector changes according to the selection of initial conditions. In other words, the local displacement vector at a point represents a common surface consisting of neighboring station points. Therefore, it is very sensitive to the weights of the observations and the location in the observation plan. In this case, the local displacement vector recommended as a local comparison criterion can be used as a local query detector.

- When the local displacement vectors have been examined in the 2009.370 epoch, it is observed that displacement vector values (d_{min}) at station points 103 and 38 exceed the limit value. In the 2009.370 epoch, the external reliability value of the 103-38 base observation disrupts the reliability distribution. However, in both 2009.370 and 2010.496 epochs, sensitivity values of stations 103 and 38 are obtained at a high level (Figure 1, Table 1).
- In the 2010.496 epoch, at station points of 74 and 201 (23), global displacement vectors have exceeded the (d_{min}) threshold value. At station points 73 and 42, only local displacement vectors have exceeded the (d_{min}) threshold value. At these points, the sensitivity values are also relatively weak (Table 1).

When the findings have been analyzed, it is observed that the global displacement magnitudes have more optimistic results. On the other hand, the local displacement vector at a point represents the common surface formed by neighboring points. Therefore, it is very sensitive to the observation weights and its locations in the network. As a result of this feature, the local displacement vectors can detect the possible outliers on the surface.

As a result; in this study, the displacement vector used as a local comparison criterion is suggested to be used as a local query detector.

REFERENCES

Berber M., (2006), Robustness analysis of geodetic networks, Ph.D. Thesis, University of Brunswick, Canada.

Hsu R., Hsiao K., (2002), Pre-computing the sensitivity of a GPS station for crustal deformation monitoring, Journal of the Chinese Institute of Engineers, 25(6), 715-722.

Kirici Yildirim, U., Sisman, Y., (2019), The deformation Analysis using hypothesis tests, International Journal of Engineering and Geosciences (IJEG), 4(2), 88-93.

Konak H., (2010), Jeodezik ağlar ve uygulamaları ders notları, KOÜ Mühendislik Fakültesi, Kocaeli.

Konak H., Küreç Nehbit P., İnce C. D., (2017), Yaşamsal altyapı sistemlerinin GPS ağlarıyla izlenmesi ve deformasyon sonuçlarının yorumlanması: Kocaeli İZDOGAP GPS ağı uygulaması, Geomatik Dergisi, 2(1), 40-52.

Konak H., (2018), Jeodezik ağlarda sağlamlık ders notları, KOÜ, FBE, Joedezi ve Jeoinformasyon Mühendisliği Anabilim Dalı.

Kuang, S., (1991), Optimization and design of deformation monitoring schemes, Ph.D. Dissertation, Department of Surveying Engineering Technical Report, University of New Brunswick, 157, 1991.

Küreç P., (2010), Yüzey ağlarının değerlendirilmesi, Yüksek Lisans Tezi, KOÜ Fen Bilimleri Enstitüsü, Kocaeli.

Küreç P., Konak H., (2011), Sıklaştırma ağlarının yer kabuğu hareketlerine karşı duyarlıkları ve olası algılayabilirlik düzeyleri, hkm Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi, Özel Sayı, 84-90.

Küreç P., Konak H., (2014), A priori sensitivity analysis for densification GPS networks and their capacities of crustal deformation monitoring: a real GPS network application, Nat. Hazards Earth Syst. Sci., 14, 1299-1308.

Küreç Nehbit P., (2018), GPS/GNSS ağları için sürekli bir gerinim izleme ve kalite değerlendirme stratejisi, Doktora Tezi, KOÜ Fen Bilimleri Enstitüsü, Kocaeli.

Öcalan, T., (2018), Investihation on the effects of number of common points in 2D transformation problem, International Journal of Engineering and Geosciences (IJEG), 4(2), 58-62.

Poyraz F., (2009), Kuzey Anadolu fay zonu doğu kesiminde yatay yerkabuğu hareketleri ve gerilme birikiminin araştırılması, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Istanbul.

Vanicek P., Craymer M. R., Krakiwsky E. J., (2001), Robustness Analysis of Geodetic Horizontal Networks, Journal of Geodesy, 75(4), 199-209.

Vanicek P., Krakiwsky E. J., Craymer M. R., Gao Y., Ong P. S., (1990), Robustness Analysis, Final contract report, Department of Surveying Engineering Technical Report, University of New Brunswick, 156.

Chou P.C., Pagano, N.J., (1992), Elasticity Elasticity. Tensor, Dyadic and Engineering Approaches, ISBN 0486669580.

International Journal of Engineering and Geosciences (IJEG),
Vol; 5, Issue; 1, pp. 049-059, February, 2020, ISSN 2548-0960, Turkey,
DOI: 10.26833/ijeg.581584

INTERPRETING DEFORMATION RESULTS OF GEODETIC NETWORK POINTS USING THE STRAIN MODELS BASED ON DIFFERENT ESTIMATION METHODS

Haluk Konak[1*], Pakize Küreç Nehbit[2], Aslıhan karaöz[1], Fazilet Cerit[1]

[1]Kocaeli University, Institutes of Natural and Sciences, Department of Geodesy and Geoinformation, Kocaeli, Turkey
(hkonak@kocaeli.edu.tr, karaozaslhan@gmail.com, fazilet.cerit2010@gmail.com); **ORCID 0000-0003-2554-8905, ORCID 0000-0003-2371-8958, ORCID 0000-0002-0524-4237**
[2]Kocaeli University, Engineering Faculty, Department of Geodesy and Geoinformation, Kocaeli, Turkey
(pkurec80@yahoo.com); **ORCID 0000-0003-2669-0401**

*Corresponding Author, Received: 24/06/2019, Accepted: 02/09/2019

ABSTRACT: Geodetic Networks designed as Deformation Networks or Continuous Networks are observed in different epochs/ sessions and evaluated as a function of time. Those can be design as global GNSS networks for aim monitoring active tectonic movements or as regional densification geodetic and deformation networks for monitoring local earthquakes and surface movements. The areas covered geodetic networks are assumed as any surface on ellipsoid or sphere. Characteristics of surfaces are analyzed with Geometric Strain Models using deformation data on surface points. In this case, effect rates on geodetic network area are determined from local surface movements or regional active earthquakes and interpreted as experimental. On the other hand, undetermined outliers by model hypothesis test affect coordinate-unknowns separately. Outliers cause deformations in certain magnitude on networks points. Therefore, network points strain in different rates and directions. Query of maximum affects caused by these strain rates is a referenced reliability method called "Robustness Analysis in Geodetic Networks". Mentioned strain rates are modelled by various estimation methods. Thus, deformation results could be interpreted together by the obtained strain components and deformation vector.

In this paper, possible strain components belonging to network points are determined with methods of L1 Norm, Least Median Squares (LMS) and Least Squares Estimation (LSE). These estimation methods are tested on KOUSAGA (Kocaeli University Permanent GPS Network). Strain components are estimated by use polyhedrons covered by network points. Obtained results are compared and analyzed according to weakness and strengths of proposed estimation methods.

Keywords: *Deformation, Strain, Estimation Methods, KOUSAGA*

1. INTRODUCTION

Geodetic networks can be designed in different shape and sizes according to their functions and purpose as global, regional or local. Point-position coordinates of these networks can be defined using geocentric or local coordinate systems and then evaluated together with classical and satellite-based observations. Geodetic networks can be continuously measured by advanced satellite measurement techniques in different epochs and sessions, and evaluated with appropriate mathematical statistics methods; their geometric structures and point-positions can be monitored periodically. Accordingly, in areas covered by geodetic networks, possible tectonic plate deformations, crustal movements, large and important engineering infrastructures and time-dependent regular periodic changes in their environment can be modeled and possible local-block movements can be detected. The areas covered by geodetic networks are assumed as any surface on ellipsoid or sphere. The displacement ratios of the network points on this surface have deformation and strain information representing the surface. Surfaces characteristics are analyzed with geometric strain models using deformation data on surface points, in any coordinate system and datum free. As a result of this situation, affect rates of the areas covered by geodetic networks from local surface movements or regional active earthquakes are determined and interpreted as experimental.

On the other hand, undetermined outliers by model hypothesis test affect coordinate-unknowns separately. Outliers cause deformations in certain magnitude on network points and these points strain in different rates and directions. Query of maximum affects caused by these strain rates is a referenced reliability method called "*Robustness Analysis in Geodetic Networks*". A priori or a posteriori strain rates are modelled by various estimation methods. Thus, deformation results could be interpreted together by the obtained strain components and deformation vectors.

Possible strain components belonging to network points on the any surface are determined with different estimation methods. Depending on the nature of the surface movements, the strain components related to any surface can be modeled and interpreted by geometric, geodetic and/or geophysical data. The estimates that having minimum variance, consistently, efficiently, sufficiently and at least asymptotically unbiased should be used for the strain models. In this study, the possible strain in the network points is determined by Least Absolute Value (LAVM, L1-Norm), Least Median Squares (LMS) and Least Squares (LS, L2-Norm) estimation methods.

In an estimation process; it is aimed to obtain the most consistent unknown parameters and the measurement corrections having the greatest probability without removing any measurements (Öztürk and Şerbetçi, 1992). The LAV estimation method is a Maximum Likelihood Estimation (MLE) method, which minimizes the sum of absolute values of corrections by using any combinations of measurements as many as the unknown parameters number. With LAV method, it is aimed to minimize the effect of the data in contradiction with normal distribution. This objective function cannot always be guaranteed by the LAV methods. In this case, LMS method is recommended as another MLE method which controls the LAV method (Niemeier, 2002).

The MLE method gives successful results when the distribution of measurements is well known. On the other hand, LS estimation method provides minimum variance and unbiased results for the sets of measures which are known to have normal distribution at the beginning or not. However, the LS method can spread the effects of possible outliers with higher weights to the other observations (Konak et al, 2005). At the same time, these outliers, which occur differently according to the measurement plan, observation types and weights, are called leverage point effect in the robust statistic (Hekimoğlu, 2005). This situation is the most criticized and query aspect of the LS method. Both the LS and the LAV estimation are adversely affected by the leverage point effect.

On the other hand, normal distribution laws can be applied easily and successfully to LS results (Öztürk and Şerbetçi 1992). As a result of this feature; LSM methods are strengthened by the LS estimation method and integrated estimation methods called robust estimation methods are developed and implemented. If the measurements with sufficient degree of freedom are in normal distribution, the LAV, LMS and LS estimation magnitudes (sample means) are the same and their variances also exhibit asymptotically unbiased behavior (Dilaver et al., 1998; Konak and Dilaver, 1998).

The purpose of this paper is to perform experimental strain or robustness analysis using polyhedrons defined for any network point, by an appropriate estimation method. For this purpose, Affine Transformation is used as a mathematical model, and the strain tensors/components of symmetric and anti-symmetric characteristics are estimated by LAV, LMS and LS estimation methods separately. In the last step, an integrated solution algorithm is proposed by utilizing the useful properties of these methods.

2. MATHEMATICAL METHOD

Any surface strain model is written in the form of the Affine Transformation Model using deformation/velocity vectors at the neighbor network points where any station point is connected (Table 1a). Strain elements representing the surface are obtained accordingly to the objective functions of LAV, LMS or LS estimation (Niemeier, 2002).

If the strain elements tensor matrix (E) are arranged in the form of symmetric (S) and anti-symmetric (A) matrices, random strain rates can be obtained. If the symmetric (S) matrix is separated into eigenvalues and eigenvectors; the principal strain axes representing any polyhedron and the strain invariants (dilation, differential rotation, maximum shear strain) in any coordinate system are obtained (Table 1b). The non-diagonal element of the anti-symmetric matrix (A) is called the differential rotation strain (Vanicek et al., 1990; Berber, 2006; Yetkin, 2012; Küreç Nehbit, 2018).

Table 1. Strain components and Mathematical Model

a. **Mathematical Model as Affine Transformation**	b. ***Strain components***
$u=\begin{bmatrix}V_x\\V_y\end{bmatrix};t_0=\begin{bmatrix}t_x\\t_y\end{bmatrix};E=\begin{bmatrix}e_{xx}&e_{xy}\\e_{yx}&e_{yy}\end{bmatrix};\Delta x=\begin{bmatrix}x_i-x_0\\y_i-y_0\end{bmatrix}$ (1) $P_0(x,y)$ Coordinates of reference point $u=t_0+E\Delta x$ Deformation vector (2) $H=\begin{bmatrix}1&0&x_0-x_0&y_0-y_0&0&0\\0&1&0&0&x_0-x_0&y_0-y_0\\1&0&x_1-x_0&y_1-y_0&0&0\\0&1&0&0&x_1-x_0&y_1-y_0\\.&.&.&.&.&.\\.&.&.&.&.&.\\1&0&x_p-x_0&y_p-y_0&0&0\\0&1&0&0&x_p-x_0&y_p-y_0\end{bmatrix}$ Design Matrix (3) $v=Ht-l$ Correction Equations (4) $l^T=\left[V_{x0}\quad V_{y0};\quad V_{x1}\quad V_{y1};\quad \ldots\quad \ldots.;\quad V_{xp}\quad V_{yp}\right]$ (5) a vector of observations; Annual velocities	$t^T=\left[t_x\quad t_y\quad e_{xx}\quad e_{xy}\quad e_{yx}\quad e_{yy}\right]$ (6) Strain components $S=\frac{1}{2}\left(E+E^T\right)=\begin{bmatrix}\varepsilon_{xx}&\varepsilon_{xy}\\\varepsilon_{xy}&\varepsilon_{yy}\end{bmatrix}$ (7) Symmetric strain components $A=\frac{1}{2}\left(E-E^T\right)=\begin{bmatrix}0&-w\\w&0\end{bmatrix}$ (8) Anti-Symmetric strain components $\lambda_1=\frac{1}{2}\left\{\varepsilon_{xx}+\varepsilon_{yy}+\sqrt{(\varepsilon_{xx}-\varepsilon_{yy})^2+4\varepsilon_{xy}^2}\right\}$ (9) the first principal strain axe: maximum eigenvalue $\lambda_2=\frac{1}{2}\left\{\varepsilon_{xx}+\varepsilon_{yy}-\sqrt{(\varepsilon_{xx}-\varepsilon_{yy})^2+4\varepsilon_{xy}^2}\right\}$ (10) the second principal strain axe: minimum eigenvalue $\theta=\frac{1}{2}\arctan\frac{2\varepsilon_{xy}}{\varepsilon_{xx}-\varepsilon_{yy}}$ (11) the direction of principal strain axe *Invariants in 2D and 3D* $\sigma=\frac{1}{2}iz\{S\}$ Dilation (12) $\mu=\lambda_1-\lambda_2$ Maximum shear strain (13) $\Omega=w$ Differential rotation (14)

3. ESTIMATION METHODS

The LAV estimation method corresponds to the solution of the Affine Transformation problem in equation (4) with the inequality system described in Table 1a. This inequality system, called linear programming, can be solved by using Simplex Algorithms. In this solution method, for model parameters, there shouldn't be any sign constraint. In this case, the measurement corrections, having equal number to the number of unknown parameters, are equal to "zero". As a Maximum Likelihood Estimation (MLE), in this method the sum of absolute values of corrections is going to be minimal (Bektaş and Şişman, 2010). Thus, a special equation solution is obtained where the number of measurements is equal to the number of unknown (Table 2a).

In LAV method, it is aimed to localize the effect of the data which has abnormal distribution in the measurement sets and to minimize the sum of the absolute values of the corrections as much as possible. However, such a purpose function cannot always be guaranteed. In other words, in case there are error-free data in some subsets of measurements, LAV estimation is not an efficient solution. On the other hand, this estimation process can be used as an a priori approach/step to be able to determine random gross errors in the transformation problems. As a result of this situation; the LMS method is suggested as another MLE method to control the LAV method (Hekimoğlu, 2005; Niemeier, 2002).

In Regression and Transformation problems, measurements separated from the barycenter of network points or from geometric distribution of the observations called "Leverage Point Effect" are highly effective on the parameters. In this case, the leverage point effect can be explained as the spreading effect of the random errors in poor controllable observations onto the other observations. The LMS estimator can determine initially the most appropriate observation set purged of outliers. Therefore, it is recommended as an estimator to control the LAV estimation (Niemeier, 2002).

The LMS method is solved by an algorithm similar to the LAV method. For example; in affine transformation problem, the design matrices are composed form the (u + 2) transformation equation row, where u is the number of unknowns and q is the number of coordinate pairs. In this case, the different solution combinations $\begin{pmatrix}p\\4\end{pmatrix}$ arise. The optimal solution is the least value of Median Squares in all of these combinations, as $\left(\Omega=median\left|v_i^2\right|\Rightarrow\min.\right)$ (Table 2b).

LAV and LMS methods, which are called MLE methods, provide successful results when the distribution of measurements is well known. On the other hand, LS estimation method is an estimation type that gives minimum variance and unbiased results for the sets of measurement which are known to normal distribution at the beginning or not. In this case, LS method can be used as an estimation method to control the MLE (Table 2c).

As a result of LS aim-function, outliers in the observation set are able to disrupt the estimation values and also have a negative effect on other observations which have a normal distribution. This is the most criticized and questioned aspect of the LS method. However, in the LS method, the higher the numbers of abnormal observations are the more extend is the unit variance of model. This property of LS method allows for the localization and elimination of outliers in observations having gross-error using mathematical-statistical methods (Öcalan 2019; Yıldırım and Şişman, 2019).

However, in strain analysis; the number of common control points is quite limited and, in this case, it is difficult to determine the abnormal observations (outliers). On the other hand, normal distribution laws can be applied easily and successfully to the LS results. As a result of this situation, taking advantage of the features of the LS squares estimation, integrated approaches/algorithms that can be controlled by LSM estimations can be developed (Table 2d).

Table 2. Estimation Methods for Strain Elements

Strain Model: 2D Affine Transformation	
a. The Least Absolute Value (LAV) Estimation	**b. The Least Median Squares (LMS) Estimation**
$\Omega = \sum \lvert v \rvert \Rightarrow \min$ $Ht - v \leq l$ $H(t^{+} + t^{-}) - (v^{+} - v^{-}) \leq l$ $t^{+} \geq 0; t^{-} \geq 0; v^{+} \geq 0; v^{-} \geq 0$ (15) Special Case The Number of Common control point p = 3 $\Omega = \sum \lvert v \rvert \Rightarrow \min \equiv \min\left\{\Omega_i ; i = 1,2,...., \binom{p}{3}\right\}$ $(p = 3; \quad n = 2 * p = 6; \quad u = 6; \quad n = u)$ $H_i^{-} = (H^T H)^{-1} H_{(u,u)}^{T}$ (16) $t_i = H_i^{-} l$	The Number of Common control point p = 4 $\Omega = median \lvert v_i^2 \rvert \Rightarrow \min \equiv \min\left\{\Omega_i ; i = 1,2,...., \binom{p}{4}\right\}$ $(p = 4; \quad m = 2 * p = 8, \quad ; u = 6; \quad m > u)$ $H_i^{-} = (H^T H)^{-1} H_{(u,m)}^{T}$ (17) $t_i = H_i^{-} l$
c. The Least Squares (LS) Estimation	**d. The Least Squares-Robust (LS-R) Estimation**
The Number of Common control point p = p $\Omega = v^T v \Rightarrow \min$ $(n = 2 * p; \quad u = 6; \quad n > u)$ $H^{-1} = (H^T H)^{-1} H^T$ (18) $t = H^{-1} l$	The Number of Common control point p = p $\Omega = v^T W_k v \Rightarrow \min$ $(n = 2 * p; \quad u = 6)$ $n > u$ $v_k = (H_t - l)_k$ $H^{-1} = (H^T W_k H)^{-1} H^T W_k$ $t_{k+1} = H^{-1} l,$ $v_{k+1} = (Ht - l)_{k+1}$ $\lvert t_{k+1} - t_k \rvert \leq \Delta$ $W_k = \begin{cases} w_i = 1 & ; \quad (x;y)_i \in LAV : GROUP\ 1 \\ w_i = e^{-\left(\frac{v_x^2 + v_y^2}{Z^2 2m_v^2}\right)_i} & ; \quad (x;y)_i \in LMS : GROUP\ 2 \\ w_i = e^{-\left(\frac{v_x^2 + v_y^2}{C^2 2m_v^2}\right)_i} & ; \quad (x;y)_i \in LS : GROUP\ 3 \end{cases}$ (19) $C = \tau_{f,1-\alpha_0} = \sqrt{\frac{fF_{h,f-h,1-\alpha/n}}{f - h + hF_{h,f-h,1-\alpha/n}}}; h = 2$ *Tau Distribution Threshold value* (20) $F_{h,f-h,1-\alpha/n}$: *Fisher Distribution Threshold value ;* $Qvv = W_{ll}^{-1} - H(H^T W H)^{-1} H^T$ $Z = \sqrt{F_{h,\propto,1-\propto/n}}$ *Normal Distribution Threshold value ;* $m_{vi} = \pm m_0 \sqrt{(Qvv)_i}$
e. The Damped Least Squares (D-LS: Levenberg- Marquard,)	
$\Omega = v^T v + 2\vartheta(t^T t - t_0)^2 \Rightarrow \min$ Objective function (21)	$t = (H^T H + \vartheta I)^{-1} H^T l$ Solution set (22)

3.1 Solution Algorithm

The main purpose of this paper is to develop a feasible and usable solution algorithm for experimental strain or theoretical robustness analysis-processes using polyhedrons defined according to any network point. In this solution algorithm, LS estimation method is controlled by LSM methods and strengthened by a robust estimation method which we recommend in necessary and obligate situations. This solution algorithm can be summarized in five steps.

In the first step, by using the LAV estimation method ***which is to minimize the sum of absolute values of the corrections***, a priori values of the strain components are estimated, for the polyhedron in any station point (Niemeier, 2002). With the help of LS estimation, the pair of common control points that provide the objective function and equal to the number of unknowns are classified as first group observations (Table 2a).

In the second step, by the LSM estimation method ***where the median square is minimum***, the results of LAV estimation are controlled. In this step, the design matrices are composed to form the transformation equation row (u + 2), where u is the number of unknowns and q is the number of coordinate pairs. The transformation unknowns are obtained. These results are compared with LAV solutions. In this comparison, the pair of common control points that provide the objective function and also aren't there in the first group are classified as second group observations (Table 2b).

In the third step, the LS estimation is used ***which minimizes the sum of squares of the corrections*** by using all the common control points for the polyhedron determined in each station point. In this step, the set of network points not included in the LAV and LMS estimation are classified as third control network points (Table 2c).

In the fourth step, the experimental strain invariants and their distribution on the network are compared. For example, if the comparison operations are based on a priori-robustness analysis, the location, direction and magnitude of the dilatations, maximum shear and rotation invariants can be questioned.

As a result of the LS estimation, observations not conforming to the distribution of corrections may occur. As this case, a fifth step, separately robust weighting functions are defined for observations disrupting the distribution in three different groups. In this regard, corrections and their cofactors obtained from LS estimation are used. However, the weights of the common control point's pairs determined by using the LAV estimation are not altered and these weights are saved as original weights. The robust weighting procedure occurs according to the threshold of the normal distribution for the second group's control points and according to the threshold of the Tau-distribution for the third group's control points (Table 2d). Under these conditions, LS-R estimation is performed by using all of the control points and the optimal strain invariants are obtained.

In the developed solution algorithm, the selected number of common control points for any polyhedron may be limited or insufficient in the outer zone points of the network. This leads to some rang or ill-conditioned defects in design matrix H. Therefore, such a polyhedron can't be included in the stress/robustness analysis (Vanicek et al., 1996; Berber, 2006; Yetkin, 2012). On the other hand, Damped LS method (D-LS) is a useful common method for problems which lead to uncertainties in the solution set and their eigenvalues are equal to zero or too close (Canıtez, 1996).

In the D-LS estimation, the constant (ϑ) added to the diagonal element of the normal equations is a damping constant which can be selected in the calculation accuracy and does not affect the solution results. These constants can be added to the eigenvalues of normal equations and consistent generalized inverses called "Lancsoz Inverse" are obtained (Table 2e).

Therefore, the D-LS method has a feature that can be successfully adapted to the LAV, LMS and LS methods especially at the outer zone points of the network especially.

LAV, LMS and LS solutions give identical results if the number of common control point pairs for any polyhedron is equal to the unknown number. In cases where the number of common controls point is equal to the number of points (u+2), LMS and LS solutions are identical.

4. NUMERICAL APPLICATION

In this study, the annual velocities for Kocaeli University Permanent GPS Test Network (KOUSAGA) points are used for the numerical application data. The KOUSAGA network is established in the process of the Project of Monitoring of İzmit-Kocaeli Natural Gas Infrastructure (IZDOGAP) with the Geodetic Networks and Geographical Information Systems. This project is a scientific research project (Figure 1).

By evaluating the 30 sec rinex data of this network points with GAMIT/GLOBK software, 3 years of regular coordinate and velocity area information of KOUSAGA network are obtained (Herring et al., 2015a and 2015b). The first group of these is selected from point of 10 IGS (NOT1, MATE, ORID, BUCU, ISTA, ANKR, TUBI, NICO, CRAO and ZECK). The second group is selected from point of 11 IGS (TEKR, BAND, BURS, ISTN, SLEE, IZMT, BILE, HEND, NAHA, BOLU and ZONG). KOUSAGA network covers İZDOGAP project network and is established to represent the velocity area of this network. KOUSAGA consist of 2 IGS (TUBI and ISTA) and 11 TUSAGA-Active (HEND, IZMT, ISTN, ZONG, BOLU, NAHA, BILE, BURS, BAND, TEKR and SLEE) stations.

The observation plan of the KOUSAGA network is designed with the aid of delaunay triangulation (Küreç Nehbit, 2018). For the strain calculations, the coordinate and velocity information defined in the ITRF 2008 reference system is used. Strain surfaces are determined as regular polyhedrons formed according to the observation plan designed for every station point. Strain components and symmetrical strain ratio tensors are obtained by LAV, LMS and LS estimation methods.

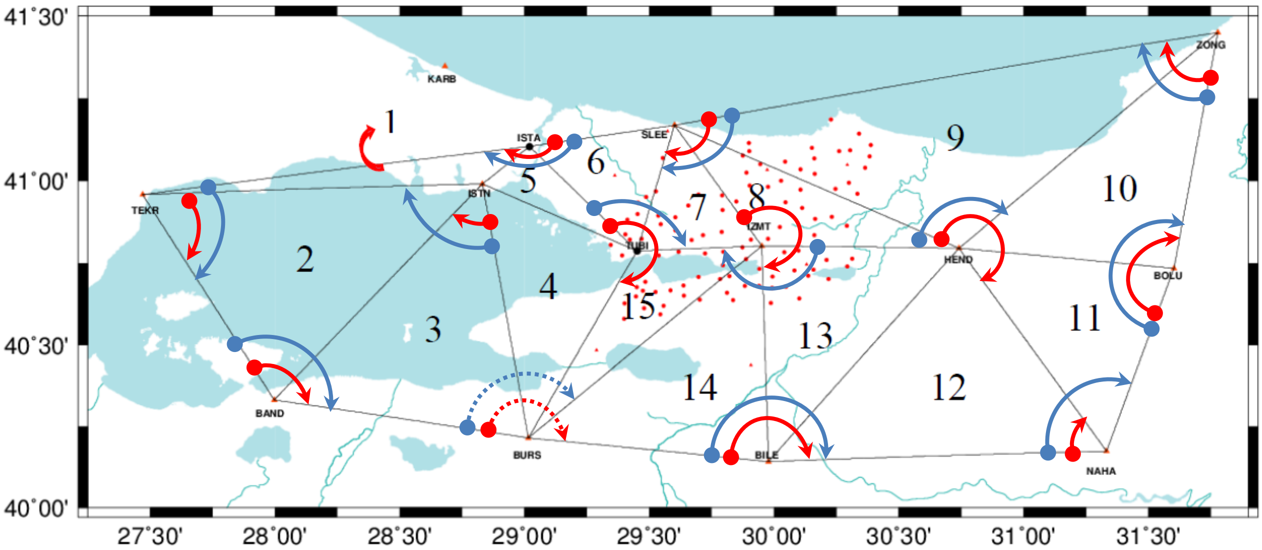


Figure 1. Kocaeli University Permanent GPS Test Network (KOUSAGA) and ranking direction of selected common points for the Strain model (Red: Red: according to LAV solution, Blue: according to LMS solution, Dashed line: barycenter point of network which is eliminated in both methods, 1, 2, …. 14, 15: number of triangles)

Experimental results are compared in terms of optimal deformation surfaces represented by any of the polyhedrons. According to these results;

- According to the results of LAV and LMS, the common points entering the set of strain model are scattered as to cover the widest possible area over the surface of any polyhedron, and these points are located around the barycenter of the polyhedron in both methods.
- As can be seen in equation (3), the design matrix H related to the surface of any polyhedron strain is constructed using the shifted coordinate differences according to this surface's barycenter point. As a result of this situation, according to the results of the LAV estimation method, barycenter points of polyhedrons could often not be included in the optimal common point combinations (Table 3).
- In the LMS estimator, common points that do not contribute positively to the geometric shape have been excluded. In LMS estimation, the barycenter point of any polyhedron is included in all combinations, except for the BURS point, which is located at the same line approximately with the BAND and BILE points (Figure 1, Table 3).

When the results are compared in terms of object functions and estimation types, in polyhedrons consisting of four (4) common network points, the results of LAV and LMS estimation remain the same. Theoretically, in the case of normal distribution of observations, the median value is expected to be the smallest and the standard deviation is to be the largest.

- According to both biased median and absolute error and unbiased standard deviation, this ranking is provided except the points of ZONG, BOLU, NAHA, BAND and TEKR. Mentioned points are the outer zone points at both ends of the KOUSAGA network (Tables 4, 5 and 6).

Table 3. A Comparison of Polyhedron Gravity Centers

Inner zone network points						
Barycenter	***p***	***LAV*** ***First group observation***	***LMS*** ***Second group observation***	***Third group observation***	***Polyhedron***	***Excluded triangles***
HEND: 2009	**7**	SLEE, ZONG, NAHA	**HEND,** IZMT, SLEE, ZONG	BOLU, BILE **(S)**	Area with Widest Angle	12 -13
IZMT: 2007	**6**	SLEE, HEND, BILE	**IZMT,** HEND, BURS, TUBI	----	Entire area	--
TUBI: 1006	**6**	ISTA, IZMT, BURS	**TUBI,** ISTA, SLEE, IZMT	ISTN **(K)**	Area with Widest Angle	4
ISTN: 2004	**6**	**ISTN,** BURS, BAND	**ISTN,** BURS, BAND, TEKR	ISTA, TUBI **(N)**	Area with Widest Angle	1 - 5
Outer zone network points						
Barycenter	***p***	***LAV*** ***First group observation***	***LMS*** ***Second group observation***	***Third group observation***	***Polyhedron***	***Excluded triangles***
ZONG: 2012	**4**	**ZONG,** BOLU, SLEE	**ZONG,** BOLU, HEND, SLEE	---	Entire area	--
BOLU: 2011	**4**	NAHA, HEND, ZONG	**BOLU,** NAHA, HEND, ZONG	----	Entire area	---
NAHA: 2010	**4**	NAHA, BILE, HEND	**NAHA,** BILE, HEND, BOLU	----	Entire area	---
BILE: 2008	**5**	BURS, HEND, NAHA	**BILE,** BURS, HEND, NAHA	IZMT **(N)**	Entire area	---
BURS: 2003	**6**	BAND, TUBI, BILE	BAND, ISTN, TUBI, BILE	**BURS**	Entire area	---
BAND: 2002	**4**	TEKR, ISTN, BURS	**BAND,** TEKR, ISTN, BURS	----	Entire area	---
TEKR: 2001	**4**	**TEKR,** ISTN, BAND	**TEKR,** ISTA, ISTN, BAND	----	Entire area	---
ISTA: 1005	**5**	SLEE, TUBI, TEKR	**ISTA,** SLEE, TUBI, TEKR	ISTN **(S)**	Entire area	---
SLEE: 2006	**6**	ZONG, HEND, TUBI	**SLEE,** ZONG, HEND, TUBI	IZMT, ISTA **(S)**	Area with Widest Angle	6
(N): in North of Barycenter				***(S): in South of Barycenter***		

Table 4. Strain Elements and Results for LS and LS-R Solutions

KOUSAGA Net Points	**p**	**Biased Standard deviation** $s_0 = \pm\sqrt{\frac{v^T P v}{2p}}$	**Unbiased Standard deviation** $m_0 = \pm\sqrt{\frac{v^T P v}{2p-6}}$	**LS** $\Omega = v^T P v \Rightarrow \min$	**Strain Components**			**Principal Strain Invariants**		
					λ_1 **(the first principal axis: maximum eigenvalue)**	**λ_2** **(the second principal axis: maximum eigenvalue)**	**θ** **(the direction of principal axis)**	**σ (Dilation)**	**μ (Maximum shear strain)**	**Ω (Differential rotation)**
HEND	7	0,00146	0,00193	$2,99.10^{-5}$	$8,19.10^{-8}$	$-1,02.10^{-7}$	$6,62.10^{-1}$	$-9,82.10^{-9}$	$1,83.10^{-7}$	$8,25.10^{-8}$
IZMT	6	0,00110	0,00156	$1,47.10^{-5}$	$1,32.10^{-7}$	$-9,70.10^{-8}$	$7,37.10^{-1}$	$1,75.10^{-8}$	$2,29.10^{-7}$	$1,10.10^{-7}$
TUBI	**6**	**0,00127**	**0,00180**	$1,95.10^{-5}$	$1,38.10^{-7}$	$-8,78.10^{-8}$	$7,46.10^{-1}$	$2,51.10^{-8}$	$2,26.10^{-7}$	$1,11.10^{-7}$
ISTN	**6**	**0,00109**	**0,00154**	$1,42.10^{-5}$	$1,53.10^{-7}$	$-1,07.10^{-7}$	$6,94.10^{-1}$	$2,27.10^{-8}$	$2,60.10^{-7}$	$1,13.10^{-7}$
ZONG	**4**	**0,00111**	**0,00222**	$9,83.10^{-6}$	$4,98.10^{-8}$	$-1,00.10^{-7}$	$7,27.10^{-1}$	$-2,52.10^{-8}$	$1,50.10^{-7}$	$6,87.10^{-8}$
BOLU	4	0,00097	0,00193	$7,48.10^{-6}$	$6,52.10^{-8}$	$-1,27.10^{-7}$	$5,86.10^{-1}$	$-3,10.10^{-8}$	$1,92.10^{-7}$	$6,51.10^{-8}$
NAHA	4	0,00118	**0,00235**	$1,11.10^{-5}$	$1,15.10^{-7}$	$-1,10.10^{-7}$	$6,69.10^{-1}$	$2,73.10^{-9}$	$2,25.10^{-7}$	$8,82.10^{-8}$
BILE	5	0,00069	**0,00109**	$4,77.10^{-6}$	$1,39.10^{-7}$	$-1,15.10^{-7}$	$7,41.10^{-1}$	$1,17.10^{-8}$	$2,54.10^{-7}$	$1,17.10^{-7}$
BURS	6	0,00051	**0,00072**	$3,12.10^{-6}$	$1,68.10^{-7}$	$-1,07.10^{-7}$	$7,17.10^{-1}$	$3,07.10^{-8}$	$2,75.10^{-7}$	$1,21.10^{-7}$
BAND	**4**	**0,00008**	**0,00016**	$4,99.10^{-8}$	$1,76.10^{-7}$	$-1,12.10^{-7}$	$6,73.10^{-1}$	$3,19.10^{-8}$	$2,89.10^{-7}$	$1,16.10^{-7}$
TEKR	4	0,00099	**0,00199**	$7,90.10^{-6}$	$1,48.10^{-7}$	$-1,15.10^{-7}$	$6,58.10^{-1}$	$1,68.10^{-8}$	$2,63.10^{-7}$	$1,15.10^{-7}$
ISTA	**5**	**0,00064**	0,00102	$4,13.10^{-6}$	$6,74.10^{-8}$	$-6,52.10^{-8}$	$7,19.10^{-1}$	$1,11.10^{-9}$	$1,33.10^{-7}$	$5,55.10^{-8}$
SLEE	**6**	**0,00042**	**0,00059**	$2,11.10^{-6}$	$4,88.10^{-8}$	$-6,01.10^{-8}$	$6,71.10^{-1}$	$-5,65.10^{-9}$	$1,09.10^{-7}$	$5,47.10^{-8}$

Table 5. Strain Elements and Results for LAV

KOUSAGA Net Points	**p**	**Biased Standard deviation** $s_0 = \pm\frac{\sum\lvert v\rvert}{2p}$	**Unbiased Standard deviation** $m_0 = \pm\frac{\sum\lvert v\rvert}{2p-6}$	**LMS** $\Omega = \sum\lvert v\rvert \Rightarrow \min$	**Strain Components**			**Principal Strain Invariants**		
					λ_1 **(the first principal axis: maximum eigenvalue)**	**λ_2** **(the second principal axis: maximum eigenvalue)**	**θ** **(the direction of principal axis)**	**σ (Dilation)**	**μ (Maximum shear strain)**	**Ω (Differential rotation)**
HEND	7	0,00102	0,00179	0,0143	$6,58.10^{-8}$	$-9,15.10^{-8}$	$6,68.10^{-1}$	$-1,28.10^{-8}$	$1,57.10^{-7}$	$8,08.10^{-8}$
IZMT	6	0,00076	0,00152	0,0091	$1,19.10^{-7}$	$-8,89.10^{-8}$	$7,70.10^{-1}$	$1,50.10^{-8}$	$2,08.10^{-7}$	$1,08.10^{-7}$
TUBI	**6**	0,00093	**0,00185**	0,0111	$1,64.10^{-7}$	$-8,76.10^{-8}$	$-7,62.10^{-1}$	$3,84.10^{-8}$	$2,52.10^{-7}$	$1,10.10^{-7}$
ISTN	**6**	0,00078	**0,00157**	0,0094	$1,77.10^{-7}$	$-1,10.10^{-7}$	$6,77.10^{-1}$	$3,33.10^{-8}$	$2,87.10^{-7}$	$1,16.10^{-7}$
ZONG	**4**	**0,00074**	**0,00295**	0,0059	$4,78.10^{-8}$	$-1,17.10^{-7}$	$-7,69.10^{-1}$	$-3,47.10^{-8}$	$1,65.10^{-7}$	$7,90.10^{-8}$
BOLU	4	**0,00045**	0,00180	0,0036	$5,59.10^{-8}$	$-1,17.10^{-7}$	$5,65.10^{-1}$	$-3,07.10^{-8}$	$1,73.10^{-7}$	$7,54.10^{-8}$
NAHA	**4**	**0,00094**	**0,00375**	0,0075	$1,18.10^{-7}$	$-1,12.10^{-7}$	$7,49.10^{-1}$	$3,22.10^{-9}$	$2,30.10^{-7}$	$1,07.10^{-7}$
BILE	5	0,00055	**0,00138**	0,0055	$1,32.10^{-7}$	$-1,12.10^{-7}$	$7,63.10^{-1}$	$1,00.10^{-8}$	$2,44.10^{-7}$	$1,10.10^{-7}$
BURS	6	0,00040	**0,00080**	0,0048	$1,69.10^{-7}$	$-1,14.10^{-7}$	$7,29.10^{-1}$	$2,77.10^{-8}$	$2,83.10^{-7}$	$1,20.10^{-7}$
BAND	**4**	**0,00005**	**0,00020**	0,0004	$1,74.10^{-7}$	$-1,12.10^{-7}$	$6,69.10^{-1}$	$3,13.10^{-8}$	$2,86.10^{-7}$	$1,15.10^{-7}$
TEKR	4	**0,00065**	**0,00260**	0,0052	$1,34.10^{-7}$	$-1,16.10^{-7}$	$6,42.10^{-1}$	$9,09.10^{-9}$	$2,50.10^{-7}$	$1,16.10^{-7}$
ISTA	**5**	0,00038	0,00095	0,0038	$7,47.10^{-8}$	$-6,64.10^{-8}$	$7,04.10^{-1}$	$4,12.10^{-9}$	$1,41.10^{-7}$	$5,90.10^{-8}$
SLEE	**6**	0,00030	**0,00060**	0,0036	$5,33.10^{-8}$	$-5,97.10^{-8}$	$6,38.10^{-1}$	$-3,19.10^{-9}$	$1,13.10^{-7}$	$5,94.10^{-8}$

Table 6. Strain Elements and Results for LMS Solutions

KOUSAGA Net Points	**n**	**Median**	**LMS** $\Omega = med\lvert v^2\rvert \Rightarrow \min$	**Strain Components**			**Principal Strain Invariants**		
				λ_1 **(the first principal axis: maximum eigenvalue)**	**λ_2** **(the second principal axis: maximum eigenvalue)**	**θ** **((the direction of principal axis)**	**σ (Dilation)**	**μ (Maximum shear strain)**	**Ω (Differential rotation)**
HEND	7	0,0004	$1,50.10^{-7}$	$5,01.10^{-8}$	$-6,24.10^{-8}$	$6,19.10^{-1}$	$-6,18.10^{-9}$	$1,12.10^{-7}$	$5,92.10^{-8}$
IZMT	6	0,0004	$1,84.10^{-7}$	$1,59.10^{-7}$	$-1,31.10^{-7}$	$6,59.10^{-1}$	$1,39.10^{-8}$	$2,90.10^{-7}$	$1,46.10^{-7}$
TUBI	6	0,0002	$4,73.10^{-8}$	$4,35.10^{-8}$	$-2,88.10^{-8}$	$7,62.10^{-1}$	$1,40.10^{-8}$	$7,23.10^{-8}$	$1,23.10^{-7}$
ISTN	6	0,0001	$1,64.10^{-8}$	$1,76.10^{-7}$	$-1,12.10^{-7}$	$6,73.10^{-1}$	$3,19.10^{-8}$	$2,89.10^{-7}$	$1,16.10^{-7}$
ZONG	4	**0,0014**	$2,05.10^{-6}$	$4,98.10^{-8}$	$-1,00.10^{-7}$	$7,27.10^{-1}$	$-2,52.10^{-8}$	$1,50.10^{-7}$	$6,87.10^{-8}$
BOLU	4	**0,0012**	$1,41.10^{-6}$	$6,52.10^{-8}$	$-1,27.10^{-7}$	$5,86.10^{-1}$	$-3,10.10^{-8}$	$1,92.10^{-7}$	$6,51.10^{-8}$
NAHA	4	**0,0016**	$2,69.10^{-6}$	$1,15.10^{-7}$	$-1,10.10^{-7}$	$6,69.10^{-1}$	$2,73.10^{-9}$	$2,25.10^{-7}$	$8,82.10^{-8}$
BILE	5	0,0005	$2,16.10^{-7}$	$1,27.10^{-7}$	$-1,10.10^{-7}$	$7,66.10^{-1}$	$8,55.10^{-9}$	$2,37.10^{-7}$	$1,07.10^{-7}$
BURS	6	0,0003	$1,04.10^{-7}$	$1,76.10^{-7}$	$-8,47.10^{-8}$	$6,60.10^{-1}$	$4,56.10^{-8}$	$2,61.10^{-7}$	$1,23.10^{-7}$
BAND	4	**0,0001**	$1,20.10^{-8}$	$1,76.10^{-7}$	$-1,12.10^{-7}$	$6,73.10^{-1}$	$3,19.10^{-8}$	$2,89.10^{-7}$	$1,16.10^{-7}$
TEKR	4	**0,0013**	$1,64.10^{-6}$	$1,48.10^{-7}$	$-1,15.10^{-7}$	$6,58.10^{-1}$	$1,68.10^{-8}$	$2,63.10^{-7}$	$1,15.10^{-7}$
ISTA	5	0,0003	$9,56.10^{-8}$	$6,94.10^{-8}$	$-6,50.10^{-8}$	$7,14.10^{-1}$	$2,20.10^{-9}$	$1,34.10^{-7}$	$5,60.10^{-8}$
SLEE	6	0,0002	$3,35.10^{-8}$	$5,53.10^{-8}$	$-6,43.10^{-8}$	$6,29.10^{-1}$	$-4,49.10^{-9}$	$1,20.10^{-7}$	$6,26.10^{-8}$

- According to the values computed by using degrees of freedom as a mixed unbiased comparison, estimation values of LAV unbiased absolute error are obtained as smaller than LS unbiased error estimation values except for HEND, IZMT, BOLU and ISTA points (Tables 4 and 5).

Principal strain invariants obtained by LAV, LMS and LS methods are compared by drawing surface maps (Figures 3, 4 and 5) and the values obtained are summarized as tables (Tables 4, 5 and 6). In this study, differential rotation maps are emphasized as an interesting sample. These maps and table data are examined; the surface maps of LAV and LS estimations are generally similar. Significant changes according to LMS estimation are observed at the inner zone points of the network. This is due to the fact that the number of common pairs in the outer zone of the network is equal to or very close to the number of strain parameters. Therefore, all three estimators are produced similar results in the outer zone.

On the other hand, the behavior of the LMS estimation is clearly recognizable at the inner zone points of the network (TUBI and IZMT). In the LMS estimation, the points representing the polyhedron according to the TUBI point are selected from the northern part of the polyhedron. The points representing the polyhedron according to the IZMT point are selected from the southern part of the polyhedron (Figure 1). The northern part of the TUBI polyhedron belongs to the EURASIA Plate and the southern part of the IZMT polyhedron belongs to the ANADOLU Plate. We would like to emphasize here as authors, the main purpose of this paper is not to question plate movements, but to examine the behavior of LAV, LMS and LS estimators. However, these estimation results are quite interesting. In other words; the distribution and number of these points on the polyhedrons are the most optimal solutions which provide the objective function of the LMS estimation. The results mentioned here should be considered as the leverage point and at the same time the common effects of observations which do not conform to the annual velocity distribution of the common points. Similar results are observed in the behaviors of the principal strain components (Figure 2 and 3).

Strain surfaces according to the dilatation maps are generally similar. However, due to the lever point effect, the Maximum Shear Strain Map of the LMS estimator is exhibited a different distribution according to the maps of other estimators (Figure 4 and 5).

In this study; according to the LS estimation results, there has not arisen any observation group that passed the threshold value of the normal distribution or Tau-distribution and excluded from the common point sets of LAV or LMS. Therefore, these have not conformed to the normal distribution. For this reason, the results of LS and LS-R estimation are identical (Tables 3,4 and 5 and Figures 2, 3 and 4).

5. CONCLUSION

When numerical results and graphs are analyzed together; LMS estimation shows that besides the velocities to related the network points, the barycenter point is also very effective in eliminating the leverage point effect.

During the computing of the strain components for any network point, the reference point selected for a polyhedron is determined as the barycenter of polyhedron. In this case, due to the objective function, LAV estimation method uses the common network points representing velocity information consistent with the barycenter. Therefore, the estimated values are the best approximate values representing the strain components. According to our experimental results, a polyhedron or deformation surface determined by LMS estimation is a shape improved and supported by barycenter point of any surface determined by use of the LAV estimation.

As a result, in the geodetic network, during the interpreting of the deformations points with experimental strain or robustness analyzes, the common network points and velocity information determined by the LMS and LAV method should be preserved as much as possible. For this reason, if the LS estimation method is used, the surface properties estimated by LMS and LAV methods should be considered and interpreted together.

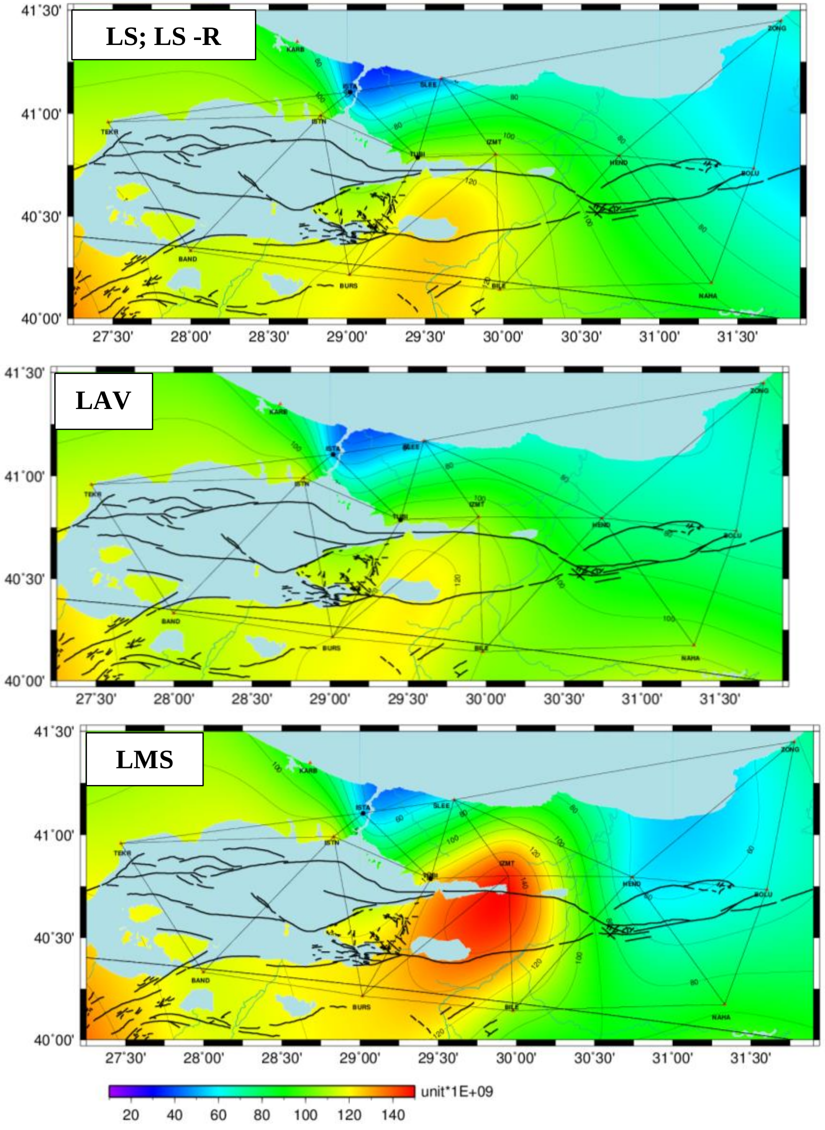


Figure 2. The Maps of Differential Rotation

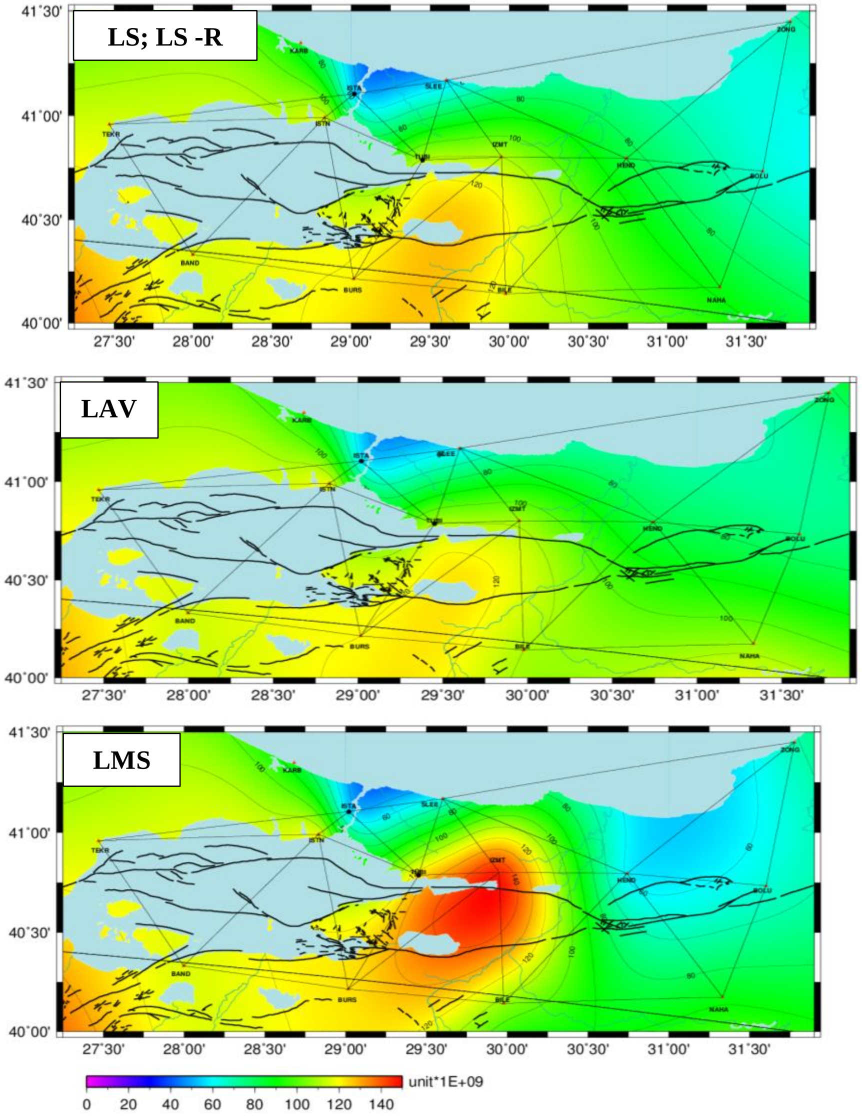


Figure 3. The Maps of The First Principal Strain Axe: Maximum Eigenvector (λ max)

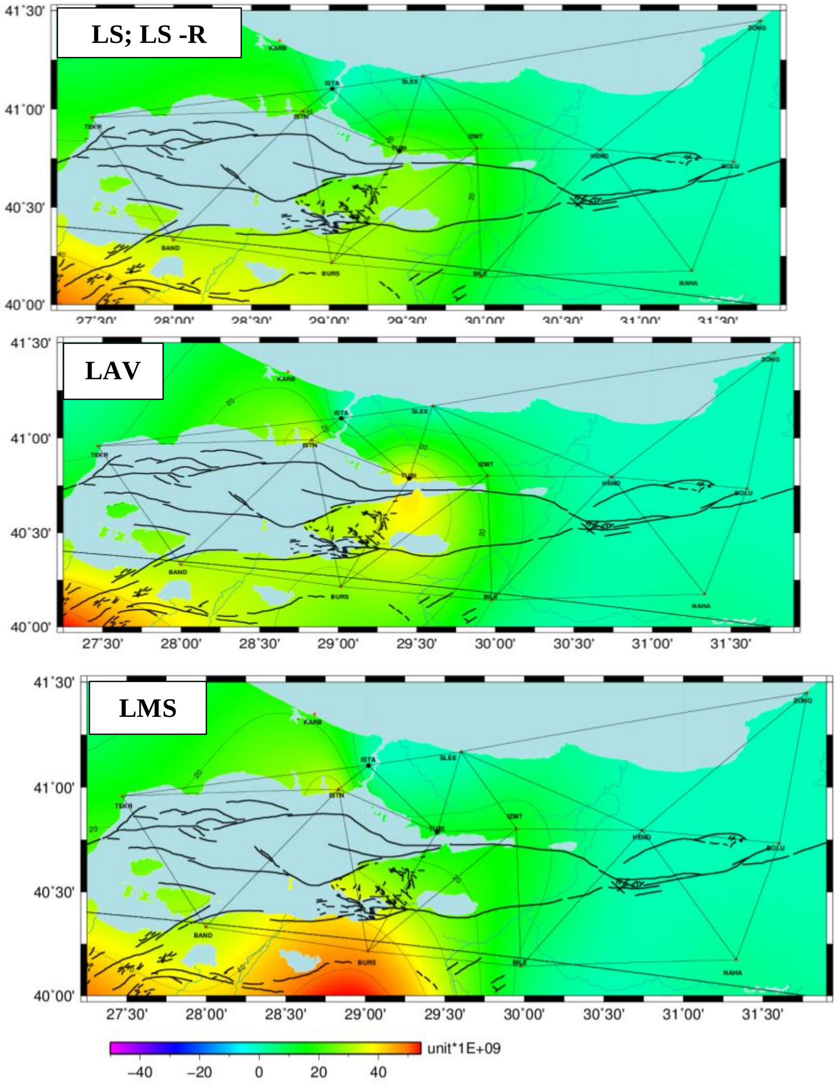


Figure 4. The Maps of Dilatation

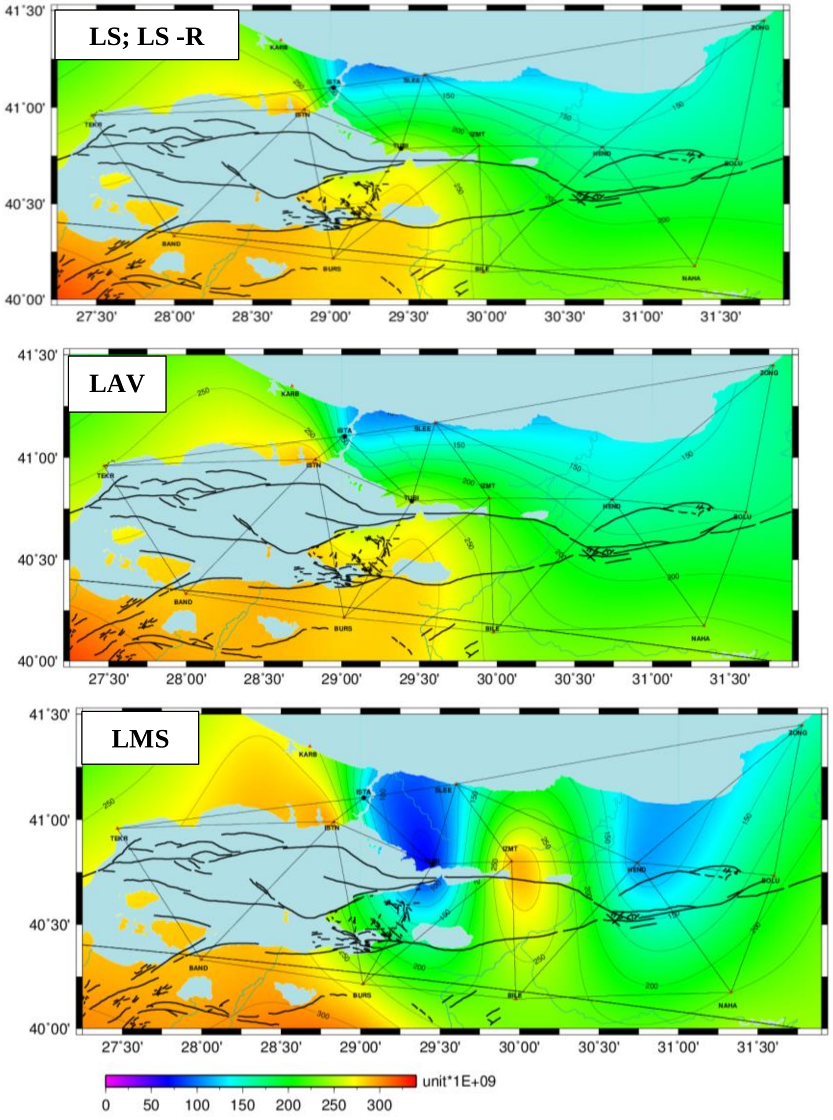


Figure 5. The Maps of Maximum Shear Strain

REFERENCES

Berber M. (2006), *Robustness Analysis of Geodetic Networks* Technical Report, University of New Brunswick, New Brunswick, Canada, 121 ss.

Bektaş S., Şişman Y. (2010) *The Comparison of L1 ve L2-norm minimization methods, International,* Journal of Physical Sciences, 5 (11), 1721-1727..

Canıtez N., (1997), *Jeofizikte Modelleme,* Literatür Yayıncılık, İstanbul, Türkiye, 368 ss.

Dilaver A, Konak H., Çepni M. S., (1998), *Jeodezik Ağlarda Uyuşumsuz Ölçülerin Yerelleştirilmesinde Kullanılan Yöntemlerin Davranışları,* Harita ve Kadastro Mühendisliği Dergisi, Ankara, 84 (1), 17-34.

Hekimoğlu, Ş. (2005), *Increasing Reliability of the Tests for Outliers Whose Magnitude is Small,* Survey Review, 38 (298), 274-285.

Herring T. A., King R. W., Floyd M. A., McClusky S. C., GAMIT Reference Manual, http://www-gpsg.mit.edu/~simon/gtgk/GAMIT_Ref.pdf, 2015.

Herring T. A., King R. W., Floyd M. A., McClusky S. C., Introduction to GAMIT/ GLOBK, http://www-gpsg.mit.edu/~simon/gtgk/Intro_GG.pdf, 2015.

Konak H., Dilaver A (1998), *Jeodezik Ağlarda Uyuşumsuz Ölçülerin Yerelleştirilmesinde Kullanılan Yöntemlerin Davranışları -II, -Fuzzy Logic (Bulanık Mantık) Yaklaşımı,* Harita ve Kadastro Mühendisliği Dergisi, Ankara, 85(1), 91-109.

Konak, H., Dilaver, A., Öztürk, E., (2005), *Effects of Observation Plan and Precision on the Duration of Outliers Detection and Fuzzy Logic: A Real Network Application,* Survey Review, (38) 298, 331–341.

Niemeier W., (2002), *Ausgleichungsrechnung,* Walter de Gruyter Kehrbuch, Berlin, Germany – New York, USA, 407 ss.

Öcalan T., (2019) *Investigation Common Points in 2D Transformation Problem,* International Journal of Engineering and Geosciences, 4 (2), 58-62.

Öztürk E., Şerbetçi M, (1992), *Dengeleme Hesabı Cilt III,* KTÜ Yayınları, 2. Baskı, Yayın No: 144/40, Trabzon, Türkiye, 558 ss.

Küreç Nehbit P., (2018), *GPS/GNSS Ağları için Sürekli Bir Gerinim İzleme ve Kalite Değerlendirme Stratejisi,* Doktora Tezi, Kocaeli Üniversitesi, Kocaeli, Türkiye.

Vanicek P., Krakiwsky E. J., Craymer M. R. Gao Y., Ong P. S. (1990), *Robustness Analysis,* Technical Report, University of New Brunswick, New Brunswick, Canada.

Yetkin M., (2012), *GNSS Gözlemlerinin Robust Kestirimi ve Robustluk Analizi Yöntemleriyle Değerlendirilmesi Üzerine Bir İnceleme,* Selçuk Üniversitesi, Konya, Türkiye.

Yıldırım. Ü. K., Şişman Y. (2019) *The Deformation Analysis Using Hypothesis Tests, International,* Journal of Engineering and Geosciences, 4 (2), 88-93.