An open-source Python library developed for GNSS & InSAR integration

Damian TONDAŚ[1,*]

[1] *KGHM CUPRUM Research and Development Centre, Wrocław, Poland, (damian.tondas@kghmcuprum.com)*

[*] *corresponding author*

Abstract

The unification of GNSS (Global Navigation Satellite System) and InSAR (Interferometric Synthetic Aperture Radar) observations is a sophisticated procedure due to their different spatio-temporal characteristics, including discrepancies in spatial and temporal resolution, measurement geometries, environmental constraints, and sensitivity to deformation components. Effective long-term ground deformation monitoring requires a comprehensive three-dimensional displacement vector, particularly addressing the north-south component, where InSAR has lower sensitivity. Moreover, the integration algorithm must be scalable and capable of ingesting non-linear ground movements to support an automated and robust deformation monitoring system. Hence, the integration procedure demands advanced methodologies that reconcile discrepancies and ensure consistency while leveraging the complementary strengths of GNSS and InSAR.

This paper introduces an original, open-source Python library designed for the seamless integration of GNSS and InSAR data. The library addresses all the aforementioned challenges, enabling the fusion of permanent GNSS observations with InSAR radar data for reliable and long-term ground deformation monitoring. The library extends the existing approach of integrating geodetic techniques by supporting simultaneous application of three InSAR computational techniques: DInSAR (Differential InSAR), SBAS (Small BAseline Subset), and PSI (Persistent Scatterer Interferometry). Furthermore, the radar observations can be derived from three independent orbits, with the flexibility to incorporate up to nine elements (pixels or PSI points) per computational method. The core of the library relies on the Kalman filter algorithm, which generates a time series of six calculated parameters (three North, East, Up components and their velocities).

Keywords: GNSS, InSAR, ground deformations, data integration, open-source Python library

1 Introduction

On-ground surface deformations are caused by natural or anthropogenic factors. In the long-term domain, we can distinguish influences like seismicity (Steinberg et al., 2020), volcanic activity (Parker et al., 2020), landslides (Wasowski and Pisano, 2020), glacial movements (Liu et al., 2019), mining industry (Yang et al., 2020), urban development (Strozzi et al., 2017), damming and reservoir construction, and others. These long-term impacts require the establishment of a consistent monitoring service to provide detailed information about ongoing deformation movements. Nowadays, the modern and well-available geodetic techniques, like GNSS (Global Navigation Satellite System) and InSAR (Interferometric Synthetic Aperture Radar), are widely adopted for monitoring of long-lasting surface motions. The widespread use of GNSS and InSAR methods is primarily due to their ability to achieve millimeter-level precision and their capacity to provide continuous data at regular intervals. The regular and dense frequency of observations can provide information about potential non-linear ground motions, which is barely possible for the other geodetic methods. Utilising levelling, total station, LiDAR (Light Detection and Ranging) or photogrammetry provides the opportunity to collect data only during intended measurement campaigns. Additionally, GNSS and InSAR techniques are relatively cost-effective and can operate in diverse environmental conditions.

$$x = [N \quad vN \quad E \quad vE \quad U \quad vU] \tag{1}$$

$$z_t = [N_{GNSS} \quad E_{GNSS} \quad U_{GNSS} \quad d_{LOS}^{DInSAR} \quad D_{LOS}^{SBAS} \quad D_{LOS}^{PSI}]^T \tag{2}$$

$$d_{LOS}^{DInSAR} = D_{LOS}^{DInSAR}/\Delta T \tag{3}$$

$$A_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & N_{LOS}^{DInSAR} & 0 & E_{LOS}^{DInSAR} & 0 & U_{LOS}^{DInSAR} \\ N_{LOS}^{SBAS} & 0 & E_{LOS}^{SBAS} & 0 & U_{LOS}^{SBAS} & 0 \\ N_{LOS}^{PSI} & 0 & E_{LOS}^{PSI} & 0 & U_{LOS}^{PSI} & 0 \end{bmatrix} \tag{4}$$

$$N_{LOS} = \sin(\theta)\sin(\alpha)\,;\; E_{LOS} = -\sin(\theta)\cos(\alpha)\,;\; U_{LOS} = \cos(\theta) \tag{5}$$

$$R_t = \begin{bmatrix} \sigma_{N_{GNSS}}^2 & \sigma_{NE_{GNSS}} & \sigma_{NU_{GNSS}} & 0 & 0 & 0 \\ \sigma_{NE_{GNSS}} & \sigma_{E_{GNSS}}^2 & \sigma_{EU_{GNSS}} & 0 & 0 & 0 \\ \sigma_{NU_{GNSS}} & \sigma_{EU_{GNSS}} & \sigma_{U_{GNSS}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{d_{LOS}^{DInSAR}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{D_{LOS}^{SBAS}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{D_{LOS}^{PSI}}^2 \end{bmatrix} \tag{6}$$

Nevertheless, the unification of GNSS and InSAR observations is a sophisticated and multi-stage procedure due to their inherent characteristics, such as different spatial and temporal resolution, various sensitivity to deformation components, discrepancy of measurement geometries, irregular latency of products, accuracy of results, and environmental or other obstacles. The comprehensive monitoring system of natural or anthropogenic long-term ground deformations requires the determination of three-dimensional (3-D) displacement vector: North, East, Up (NEU) components. Moreover, the integration algorithm must be scalable and capable of ingesting non-linear ground movements to support an automated and robust system for deformation monitoring. Hence, the integration procedure requires advanced algorithms to reconcile discrepancies and ensure consistency while exploiting their complementary strengths.

The methodology fulfilling these requirements was developed by Tondaś et al. (2023). The authors created the integration methodology based on the Kalman filter approach. The algorithm designed by authors used only GNSS and DInSAR (Differential InSAR) methods of observations, while the presented open-source library extends the previous solution by including also Multi-temporal InSAR (MT-InSAR) methods: SBAS (Small BAseline Subset) and PSI (Persistent Scatterer Interferometry). It should be highlighted that in the proposed method of integration, GNSS, SBAS and PSI observations describe absolute positions, while the DInSAR data are considered as velocity measurements. Therefore, the fusion procedure allows to preserve observations in their original domains (3-D NEU vector for GNSS and one-dimensional line-of-sight (LOS) imaginary for InSAR techniques).

2 Mathematical model of the library

The software integrates permanent GNSS data and radar observations, incorporating specific InSAR computational methods such as DInSAR, SBAS, and PSI. The model of fusion consists of the Kalman filter (Kalman, 1960) algorithm. The Kalman Filter consists of two main steps in a recursive estimation process: *time-update* and *measurement-update*.

The *time-update* stage predicts the current state and uncertainty of the system. The core of the library aligns with the methodology proposed by Tondaś et al. (2023), where the state vector - x in eq. (1) comprises six parameters: three components of the displacements (NEU) and their corresponding velocities (vN, vE, vU).

To obtain the filtered system state with the variances, in the *measurement-update* step, the prediction is combined with the actual observations in terms of minimum mean square error. The library

extends on the *measurement-update* part outlined in the article Tondaś et al. (2023) by adding two MT-InSAR data sources: SBAS and PSI. In the vector of the observations - z_t defined in eq. (2) three elements N_{GNSS}, E_{GNSS}, U_{GNSS}) refer to the topocentric position of the permanent GNSS station, received by converting Cartesian coordinates. The three last elements (d_{LOS}^{DInSAR}, D_{LOS}^{SBAS}, D_{LOS}^{PSI}) are the LOS displacements, obtained from the DInSAR, SBAS and PSI methods, respectively.

In contrast to the MT-InSAR observations, the DInSAR displacement signature is signed by lowercase „d" due to different nature of InSAR images processing. Both SBAS and PSI methods estimate the time series of displacements in the absolute domain whereas, the DInSAR solution provides the relative deformations referring to the acquisition time of the primary image of the pair forming each interferogram. Therefore, the DInSAR original observations - D_{LOS}^{DInSAR} in eq. (3) were divided by the time factor (ΔT), which corresponds to the period of the interferogram time span acquisition.

In the projection matrix - A_t in eq. (4) the number of rows is equal to the number of measurements (z_t), while the number of columns determines the number of parameters (x). The parameters N_{LOS}, E_{LOS}, U_{LOS} comprise part of 1-D LOS vector of DInSAR, SBAS and PSI observations. The exact definition of these parameters is presented in eq. (5), where θ is a value of the incidence viewing angle and α is an azimuth/heading angle of the InSAR satellite.

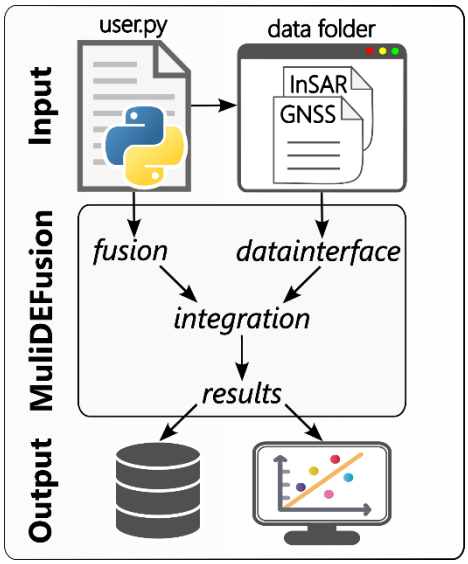

Figure 1. The library architecture

The measurement noise matrix - R_t in eq. (6) contains the correlated variances of the GNSS topocentric coordinates ($\sigma^2_{N_{GNSS}}$, $\sigma^2_{E_{GNSS}}$, $\sigma^2_{U_{GNSS}}$, $\sigma_{NE_{GNSS}}$, $\sigma_{NU_{GNSS}}$, $\sigma_{EU_{GNSS}}$) whereas InSAR estimates are assumed to be uncorrelated ($\sigma^2_{d_{LOS}^{DInSAR}}$, $\sigma^2_{d_{LOS}^{SBAS}}$, $\sigma^2_{d_{LOS}^{PSI}}$).

The overall performance scheme of the integration model outlined in eqs. (2-6) considers three different InSAR calculation methods conducted for a single orbit. However, the introduced algorithm can be easily extended by more InSAR data. The created library allows for simultaneous processing of the data registered from three various orbits. Moreover, each InSAR technique is able to ingest even nine elements (pixels or PSI points) for every introduced orbit. Within the particular method, the naming style of elements can be the same or be different for subsequent orbits. The maximum number of distinct InSAR element signatures within a particular method is equal to 10.

3 The library architecture

3.1 Installation

The software is an open-source Python library, available on Python Package Index (PyPI) repository. The documentation of the programme is published at: https://damiantondas.github.io/multidefusion, whereas the source code is publicly open at: https://github.com/damiantondas/multidefusion. To install the library, the following command in a user's terminal should be executed:

pip install multidefusion

To start the integration process in a Python Integrated Development Environments (IDE) like *Spyder*, *PyCharm*, *Visual Studio Code*, or *Jupyter Notebook*, the user must create a Python script and prepare the input GNSS and InSAR data files. The complete library architecture is shown on Figure 1.

3.2 Input files structure

The library is based on geodetic observations and the integration procedure is provided for the point domain scenario. Hence, for the each point involved into the fusion procedure, the separate folder containing input data files has to be created by the user. The input data ASCII files have a specified structure of data storage and the pattern of naming. The pattern for GNSS file is *GNSS.txt* and for InSAR: *Type_Orbit_Element.txt*, where:

- *Type* - a mandatory signature of the InSAR calculation method with allowed values: *DInSAR*, *SBAS* or *PSI*,
- *Orbit* - a mandatory signature of InSAR orbit with allowed types: string or integer, e.g., *Asc*, *Dsc*, *51*, *3a*, etc.,
- *Element* - a non-mandatory signature of a particular pixel or PSI point with allowed types: string or integer, e.g., *1*, *102fa*, *main*, etc.

Maintenance of the specified structure for all input files is necessary to ensure the successful completion of the integration procedure. The number of files allowed per point is defined by the following terms:

- The *GNSS.txt* is a mandatory file.
- The InSAR *Type* can be realised only by the *DInSAR*, *SBAS* or *PSI* signature.
- The number of distinct InSAR *Orbit* signatures must be less than or equal to 3.
- The distinct InSAR *Element* signatures within particular *Orbit* must be less than or equal to 9.
- The distinct InSAR *Element* signatures within particular *Type* must be less than or equal to 10.

The exact structure of the GNSS and InSAR time-series are described on the website in the full library documentation. Moreover, the trial dataset of input files is available on the GitHub and Zenodo platform (https://doi.org/10.5281/zenodo.11064301).

3.3 Launching the fusion procedure

After the input files preparation, to launch the fusion algorithm, user has to create his individual Python script. In the script, the library environment is loaded using the formula:

```
from multidefusion.fusion import run_fusion
```

In the next line of code, the user is obliged to provide four parameters to the imported function *run_fusion*: *stations*, *path*, *method* and *noise*. The detail definition of requirements is presented in Table 1.

The example Python script can be write as follows:

```
from multidefusion.fusion import run_fusion

integration = run_fusion(stations = "ALL",
                         path = "/path/to/folder",
                         method = "forward",
                         noise = 0.03)
```

Table 1. Example table

Argument	Type	Description
stations	list or str	Provide a list of a particular station folder or type "*ALL"* to process all folders found in the specified path
path	str	Path to the directory containing station data
method	str	Fusion method (*forward* or *forward-backward*)
noise	float	Noise level of the integration system $[mm/day^2]$

The argument *method* provides the possibility of usage the Kalman filter algorithm in two modes: *forward* or *forward-backward*. The *forward* approach is suitable to continuous datasets without significant interruptions. This method works in a single recursive mode, therefore the detection and elimination of outliers, that potentially occurred in the past, is not possible. However, the gaps in the time series can occur due to a number of factors, such as a power supply failure of the GNSS receiver. In order to get better estimates of the integration results, the *forward-backward* mode should be applied. The Kalman backward filter operates in dual-recursive mode (first forward, then backward), therefore the backward approach cannot be run separately - the results from the forward filter are required in every back recursion.

In the fusion methodology the zero-mean acceleration model is used to represent the noise covariance matrix of the system. The *noise* parameter represents the variance of the acceleration, expressed in $\mathrm{mm/day^2}$ unit. The value of the *noise* should be assigned individually by user in empirical way, because the noise covariance matrix represents the unknown balance between inherent noise of the system and level of uncertainties of gained measurements.

3.4 Integration algorithm and output

After the proper preparation done by user, the integration procedure is run and made by four inherent classes created within the library: *fusion*, *datainterface*, *integration* and *results* (Figure 1).

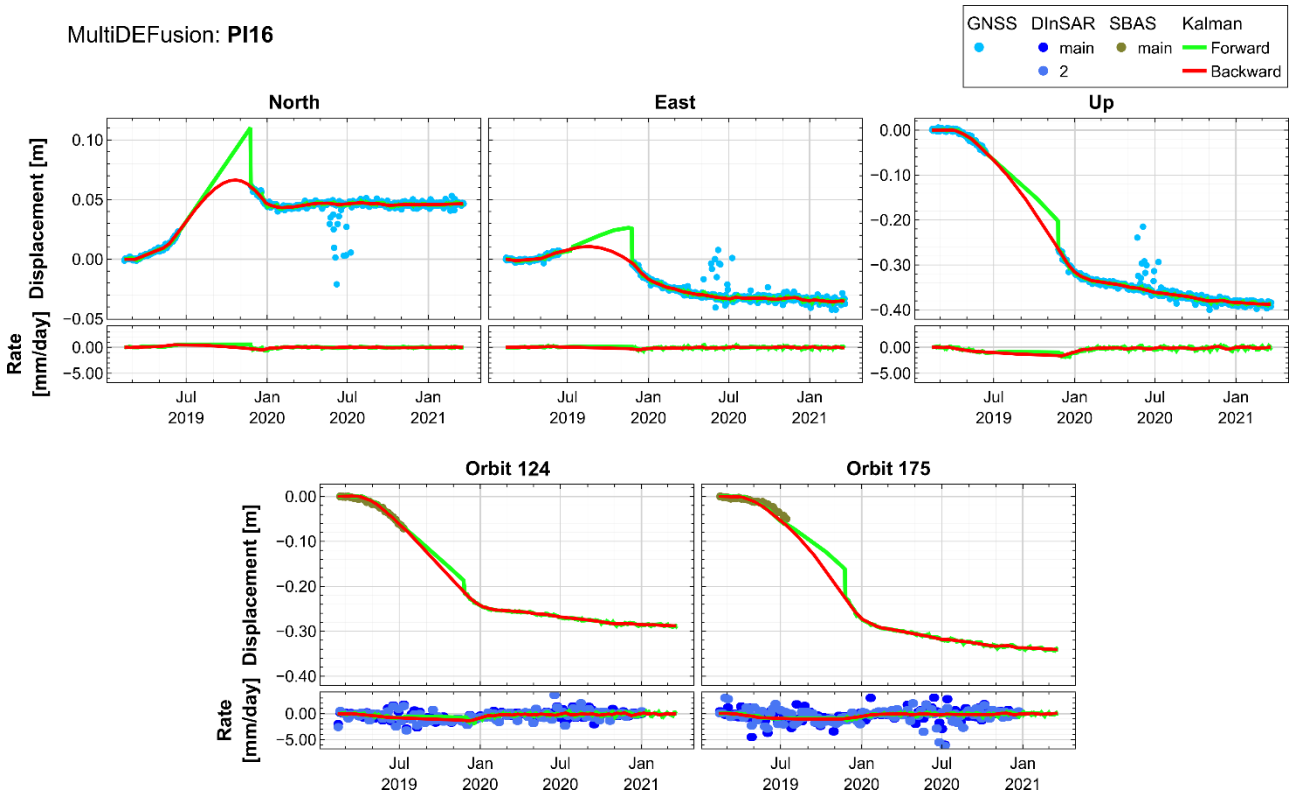

Figure 2. Displacement values (top rows) and rates (bottom rows) of the PI16 station. The charts present results of the integration performed on GNSS permanent data (light blue dots), DInSAR data (dark blue dots), SBAS data (brown dots), forward Kalman filter (green lines), and backward Kalman filter results (red lines).

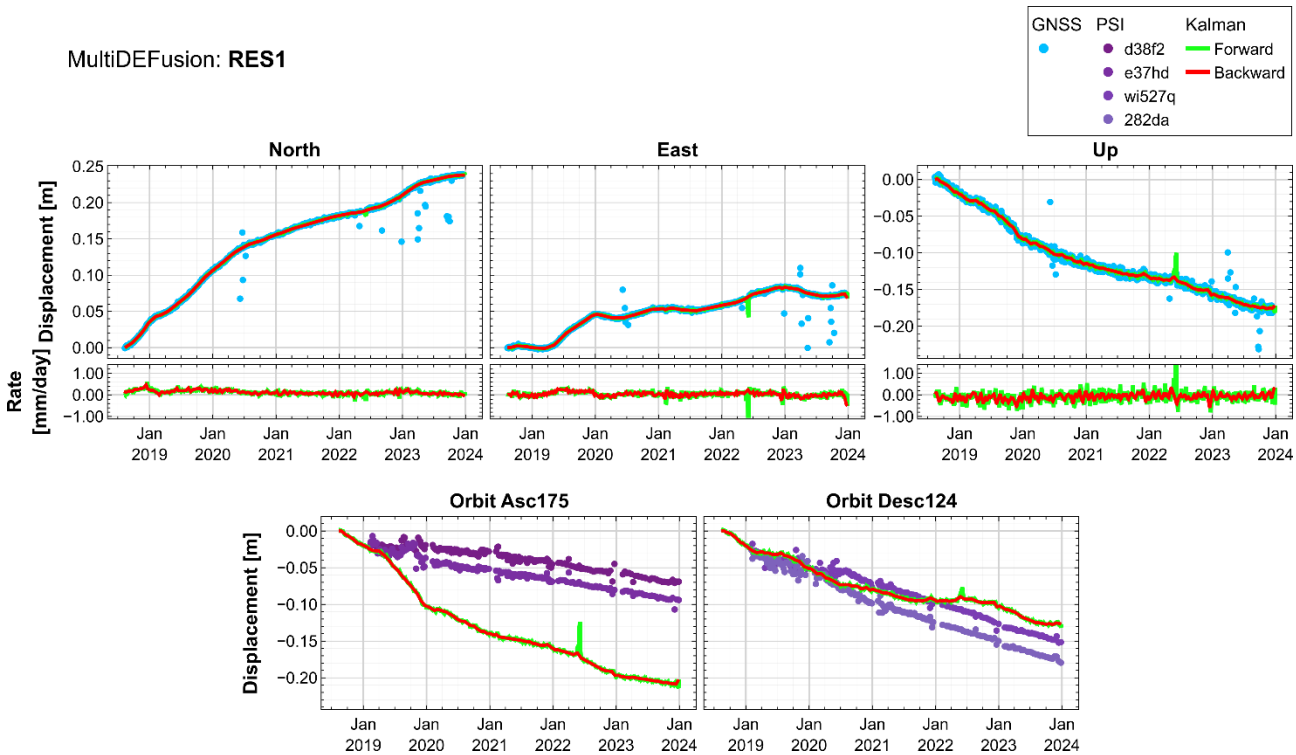

Figure 3. Displacement values (top rows) and rates (bottom rows) of the RES1 station. The charts present results of the integration performed on GNSS permanent data (light blue dots), PSI data from EGMS (purple dots), forward Kalman filter (green lines), and backward Kalman filter results (red lines).

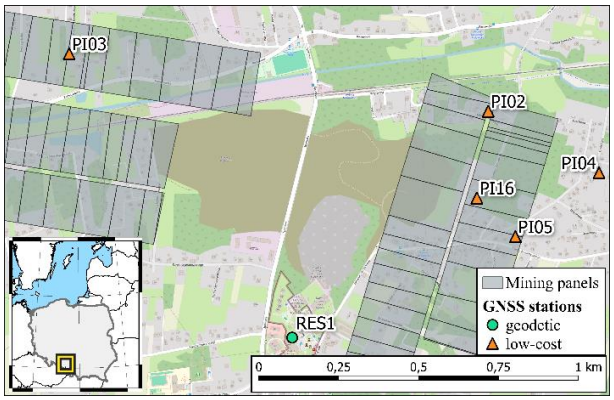

Figure 4. The study area including mining exploitation panels (grey rectangles) and locations of geodetic (green circle) and low-cost GNSS stations (orange triangles).

The *fusion* and *datainterface* modules are working in parallel. The *fusion* module loads all user-defined parameters from the Python script, while the *datainterface* module reads and harmonises the ASCII input files. The *integration* class performs the fusion procedure based on already prepared data frames of measurements. The *results* module provides the integration output from the library. The output contains two types of results: an interactive displacement plot, separate for each selected point and the Python dictionary which contains the time-series of the fusion results.

4 Case study

The library is designed to integrate observations of constant linear as well as rapid non-linear long-term ground motions. Nowadays, to improve spatial density, low-cost GNSS receivers are increasingly used. However, technical issues such as power loss may cause significant interruptions in observed time-series. Hence, to test the usability of the library, the case study containing rapid non-linear movements with the several months data-breaks is performed.

The area of interest is focused on the ground changes occurred due to coal extraction. The coal mine is located in Rydułtowy, in the southwestern part of the Upper Silesian Coal Basin (USCB), Poland. Rydułtowy is the oldest continuously operational mine in Upper Silesia (since 1792) with a planned lifetime of exploitation up to 2040 (https://pgg.pl/). The mining area covers 46 km^2 with operational resources of around 80.4 million tons and a production capacity of 9000-9500 tons per day. The longwall method of coal extraction is used in this mine with the depth of the seams between 800 and 1200 m. The work in the region of interest is ongoing on several walls at 4 seams, each with a thickness of about 18 m (Ćwiękała, 2019). The investigated area covers two walls, with a spatial extent of 2 km^2, and a moving, non-linear deformation bowl with a diameter of 200 m.

To assess the efficiency of the integration methodology, a comparative analysis was conducted on two stations located within the subsidence zone of the Rydułtowy coal mine: low-cost PI16 (Figure 2) and geodetic RES1 (Figure 3). These sites experience significant ground deformation due to longwall mining operations (Figure 4). For both points the observations from the same Sentinel-1 orbits are taken (Ascending - 175 and Descending - 124). The InSAR data for PI16 station are calculated using DInSAR (data from two pixels - “main” and “2”) and SBAS (data from one pixel “main”), whereas RES1 station corresponds with four PSI points (“d38f2”, “e37hd”, “wi527q” and “282da”). The PSI points represents the LOS Calibrated (Level 2B) ascending and descending time-series developed by European Ground Motion Service (EGMS).

The integration for the RES1 and PI16 sites was conducted by Kalman filter using both *forward* and *backward* approaches using the same *noise* level: 0.015 mm/day^2. The PSI data from 175 orbit at RES1 (Figure 3) show systematic offsets, suggesting potential biases in the EGMS-derived results. The GNSS data presents a data-break around May 2022. For this period, the Kalman forward results (green line) presents a significant jump for all components. The fluctuation relies on the mentioned disagreement between observed and expected EGMS time-series from 175 orbit. Nevertheless, the forward jump is reduced by the backward algorithm (red line). For both stations the GNSS time-series presents some outliers, however the these values are excluded by the fusion procedure.

The PI16 station (Figure 2) is using the DInSAR and SBAS datasets, where DInSAR is considered as a rate observation. In comparison to the Kalman filter results, the DInSAR shows the a high-level of data noise. The SBAS covers only 6 months of the whole PI16 observations, however for both InSAR orbits, the agreement with the fusion time-series is high.

5 Conclusions

The results indicate that the forward-backward Kalman filtering approach significantly enhances deformation estimates. The backward filter refines the initial forward estimation, reducing noise and enhancing continuity. The comparison between PI16 and RES1 stations highlights differences in

deformation behaviour, likely due to their specific positions relative to active mining panels.

Overall, while the Kalman filter approach effectively integrates multiple data sources, the analysis highlights the limitations of PSI and DInSAR data in capturing displacements towards the InSAR satellite and rapid motion events accurately. The discrepancies between the different datasets emphasise the need for careful interpretation of remotely sensed measurements, especially when combined with ground-based observations.

The presented results confirm that the developed library can effectively handle non-linear deformation processes and data gaps, ensuring reliable monitoring of long-term surface movements in mining-affected regions. Future work will focus on expanding the integration approach to incorporate additional data sources, such as levelling, total station or LiDAR.

Moreover, the library serves as a solid foundation for broader area-domain integration, where combining diverse geodetic and remote sensing techniques can improve deformation analysis. By leveraging the strengths of each data source, the approach enhances accuracy and resilience in geohazard assessment, infrastructure monitoring, and long-term environmental studies. Importantly, the developed library is publicly available, allowing researchers and professionals to freely access and utilise it for their own applications, fostering collaboration and further advancements in the field.

Acknowledgements

The GNSS, DInSAR and SBAS data have been developed within the frames of the EPOS-PL POIR.04.02.00-14-A003/16 and EPOS-PL+ POIR.04.02.00-00-C005/19-00 projects, that were funded by the Operational Program Smart Growth 2014–2020, Priority IV: Increasing research potential, Action 4.2: Development of modern research infrastructure in the science sector. The presented investigation was accomplished as part of a Polish-Chinese SHENG project at the Institute of Geodesy and Geoinformatics, Wroclaw University of Environmental and Life Sciences. The presented paper was performed thanks to the institutions which provided the access to their databases: Military University of Technology (raw GNSS observations from low-cost stations: PI02, PI03, PI04, PI05, PI16, and calculated results of estimation from GNSS campaign measurements), Główny Instytut Górnictwa (detailed information about coal extraction panels), International GNSS Service and the European Reference (EUREF) Permanent Network (raw GNSS observations from reference stations).

References

Ćwiękała, M. Korelacja wyników monitoringu geodezyjnego z sejsmicznością indukowaną w czasie eksploatacji ściany VIb-E1 w pokładzie 703/1 w KWK ROW Ruch Rydułtowy. 2019

Liu, L.; Jiang, L.; Jiang, H.; Wang, H.; Ma, N.; Xu, H. Accelerated Glacier Mass Loss (2011–2016) over the Puruogangri Ice Field in the Inner Tibetan Plateau Revealed by Bistatic InSAR Measurements. Remote Sens. Environ. 2019, 231, 111241.

Kalman, R.E. A new approach to linear filtering and prediction problems. J Basic Eng 1960, 82(1):35–45

Parker, A.L.; Biggs, J.; Lu, Z. Time-Scale and Mechanism of Subsidence at Lassen Volcanic Center, CA, from InSAR. J. Volcanol. Geotherm. Res. 2016, 320, 117–127.

Steinberg, A.; Sudhaus, H.; Heimann, S.; Krüger, F. Sensitivity of InSAR and Teleseismic Observations to Earthquake Rupture Segmentation. Geophys. J. Int. 2020, 223, 875–907.

Strozzi, T.; Caduff, R.; Wegmüller, U.; Raetzo, H.; Hauser, M. Widespread Surface Subsidence Measured with Satellite SAR Interferometry in the Swiss Alpine Range Associated with the Construction of the Gotthard Base Tunnel. Remote Sens. Environ. 2017, 190, 1–12.

Tondaś, D., Ilieva, M., van Leijen, F. et al. Kalman filter-based integration of GNSS and InSAR observations for local nonlinear strong deformations. J Geod. 2023, 97, 109.

Yang, Z.; Li, Z.; Zhu, J.; Wang, Y.; Wu, L. Use of SAR/InSAR in Mining Deformation Monitoring, Parameter Inversion, and Forward Predictions: A Review. IEEE Geosci. Remote Sens. Mag. 2020, 8, 71–90.

Wasowski, J.; Pisano, L. Long-Term InSAR, Borehole Inclinometer, and Rainfall Records Provide Insight into the Mechanism and Activity Patterns of an Extremely Slow Urbanized Landslide. Landslides 2020, 17, 445–457.