17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 522

An Operational Strategy to Enhance Real-Time Spacecraft Track Monitoring using Automated Historical Context Generation

Rishi Verma[a], Kyongsik Yun[a], Edward Barraza[a], John Mason[a,b], Jay Wyatt[a]
[a] *Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, 91108 Pasadena, CA USA* [b] *Peraton/Metis Technologies, 1400 S. Shamrock Monrovia, CA. 91016*

Abstract

The Deep Space Network (DSN), managed by NASA's Jet Propulsion Laboratory in Pasadena California, is a multinational, cooperative communications infrastructure for deep space communications and science applications. Use cases include tracking, telemetry, and commanding of NASA and partner-agency deep-space spacecraft. In addition to this, select radio astronomy, radio science, and radar activities also utilize the DSN. The network has operated with a high degree of reliability for over fifty years. New challenges are emerging however, including supporting more numerous (and smaller) spacecraft, as well as handling a greater complexity of operations due to increased requirements on network operational staff – specifically link-control operators (LCOs). Additionally, the DSN's recent adherence to a "follow-the-sun" (FtS) model, where a given Deep Space Communications Complex (DSCC) operates the entire global network during its respective daylight hours, presents potential challenges including in terms of numerosity and diversity of equipment that LCOs need to monitor as more spacecraft and antenna are added to the DSN in the future. These challenges motivate the need for greater automation and operational intelligence tools to streamline and simplify the information the DSN presents to LCOs. In light of these challenges, we present an automation strategy for identifying and highlighting the historical performance context for spacecraft communication link operations. We propose a software called Track Augmentation and Performance Analysis System (TAPAS) to automatically generate analytic comparisons between historical spacecraft communication tracking scenarios (hereafter referred to as "tracks") compared to real-time tracking scenarios as well as present a user interface strategy to communicate these complex calculations to LCOs in an intuitive way. Through TAPAS, it is our goal to offer a streamlined way for LCOs to simplify and reduce the time it takes to answer at least two key questions about ongoing, live spacecraft tracks: (1) is a given live (reference) track off-nominal compared to a historical baseline of similarly configured tracks, and (2) which segments of the reference track showcase a deviation, if at all, from historical tracks and why.

Keywords: spacecraft monitoring, machine learning, historical context, operations, statistical analysis

Acronyms/Abbreviation

- **NASA: National Aeronautics and Space Administration**
- **DSN: Deep Space Network**
- **LCO: Link Control Operator**
- **DSCC: Deep Space Communication Complex**
- **TAPAS: Track Augmentation and Performance Analysis System**
- **DCEP: DSN Complex Event Processing**
- **TV: Track Visualizer**

1. Background

The use case scenario motivating this work was gleaned from a comprehensive, electronic survey of LCOs' operational needs conducted in fiscal year 2021. The survey highlighted an important need for DSN link control operations: the ability to detect deviations between a reference track and similarly-configured historical tracks, as well as to pinpoint the nature of these key deviations. To address this use case, we propose a tool to automatically compare the time series performance of real-time track equipment monitoring information (hereafter referred to as "monitor data") between historical tracks and ongoing real-time tracks. The comparison operation involves modeling the time series behavior of monitor data for a series of tracks and using those models to subsequently calculate numerical scores that summarize the amount of deviation occurred. This score compactly aggregates the key similarities and differences between monitor data performance between tracks and is designed to be representative of track configuration difference subtleties. We generate this score using a combination of statistical techniques including Pearson correlation, dynamic time warping, Euclidean distance, and other techniques (see Section 2.1). Further, we model monitor data performance as piecewise functions in order to compress the underlying track data for more efficient use as well as to identify significant change points that may be of value to LCOs. These track comparison scores are presented to the LCO in a user interface that quickly brings attention to, and highlights key differences between given tracks. Multiple combinations are calculated in parallel between a given reference track and a series of historical tracks to provide an aggregate, visual summary of average deviation from historical norms. Thus we provide the LCO a novel method to assess the normality of track behavior that compactly scales against an increasing number of spacecraft tracks needing support. Although currently we generate these scores between individual monitor data items only, our overall strategy is to eventually extend this technique to generate aggregate comparison scores of whole track performances involving potentially hundreds of individual monitor data streams between a series of tracks.

Our tool is enabled by the ingestion, filtering, and storage of key DSN operationally relevant data into a real-time stream processing system called DSN Complex Event Processing (DCEP) [1]. DCEP stores 30 days' (or more in the future) worth of historical monitor data (as well as other data products). DCEP also provides an LCO with a user-application called "Track Visualizer" (TV), which subscribes to both real-time and historical data from DCEP and visualizes it in a time-series graphical user interface. Using TV, LCOs are able to view live DSN tracking scenarios, like ranging, uplinking, and downlinking, etc., and correlate them visually with other time series data products like logs or spacecraft sequences of events. Further, TV supports the loading of historical DSN track data to allow an LCO to visually evaluate the relationship between a live, reference DSN track performance and its historical counterpart tracks. This task is currently done manually through visual inspection and is limited by the screen real-estate available to the LCO in terms of combinations of tracks that can be evaluated. We build a track comparison user interface strategy on top of TV to automate this historical context generation process via numerical approaches and to scale this task against a variable number of tracks.

2. Technical Approach

At its core, TAPAS is an analytic engine that compares track performance using time series data and generates a correlation estimate score. Our technical approach is to offer a composite calculation, dubbed the "similarity score", to more quickly estimate the overall time-series behavior between DSN tracks' monitor data. Moreover, we seek to leverage existing DSN network monitor and control software infrastructure and integrate with existing user interfaces LCOs have familiarity with. In this section, we describe the similarity score computation in detail as well as describe the overall architecture of TAPAS in-depth.

2.1 Similarity Score

In developing the similarity score, we compared the effectiveness of three different time series comparison methods. The first method evaluated was Euclidean distance, also known as L2 distance, which is a widely used and simple method for measuring the similarity between two time series. However, this method had a limitation in that it could not accurately represent the distance between two time series of different lengths. To address this limitation, we also evaluated Dynamic Time Warping (DTW) as a measure of elastic dissimilarity. DTW is a technique that can align two time series, even if they are temporally unsynchronized, by warping the time axis non-uniformly. This allows for a more accurate representation of the similarity between two time series. Lastly, we evaluated the correlation coefficient as a measure of similarity. The correlation coefficient is a measure of the linear relationship between two variables and can range from -1 to 1, with values closer to 1 indicating a stronger correlation. To further improve the accuracy of our results, we devised an ensemble method by combining the three methods (see Equation 1). The ensemble method was designed to take advantage of the strengths of each individual method while minimizing their weaknesses. For more details on the ensemble method, please refer to our previous technical paper [2].

$$\boldsymbol{SimilarityScore} = \boldsymbol{PearsonCorrelation} - (\boldsymbol{EucleadianDistance} + \boldsymbol{DynamicTimeWarping})/\boldsymbol{k}$$

Eq. 1

2.2 Architecture

The software infrastructure we integrate with, DCEP, is designed to store time series data in real-time, with data being first stored in an open source data messaging platform called Apache Kafka (Kafka for short) and then subsequently in a document database called Elasticsearch. This allows for both ongoing tracks to be accessed from Kafka and historical tracks to be accessed from Elasticsearch. However, we found that extracting historical tracks from ElasticSearch resulted in significant data retrieval time delays, with delays ranging from 10 to 100 seconds. This is primarily due to the complexity of the queries required to extract target tracks under certain conditions and the large data volumes that need to be searched. Additionally, Elasticsearch by default is limited to the extraction of 10,000 time series data points per query, which typically forces our tool to invoke 20 to 100 iterations of queries to Elasticsearch in order to extract relevant tracks with our target conditions. This creates a bottleneck that prevents real-time processing of track comparisons. On the other hand, an ongoing track can be extracted from Kafka in real-time, with a delay of only 1 ms for receiving streaming data. This highlights the potential for using Kafka for real-time time series data processing.

When obtaining historical track information, the time lag of tens to hundreds of seconds due to Elasticsearch data retrieval is not feasible to support our on-demand needs. To overcome this limitation, we propose to access Elasticsearch track data offline, compress the time series data by at least 1000x, and then compute similarity metrics on demand (see Figure 1). Specifically, we use piecewise polynomial approximation to compress time series data in order to access Elasticsearch track data offline and compute upon it in real time. Piecewise polynomial approximation is a popular data compression method that approximates a raw data stream with multiple polynomials. The polynomial coefficients corresponding to the best-fit curve can be calculated by the least squares method, which minimizes the sum of the squared residuals between the observed and fitted values. In our study, we tested a variable number of hinge points from 5 to 30 and found that the approximation achieved a storage savings of 1000x while accurately modeling the original time series at more than 20 hinge points. This provides us with a balance of highly availability, accuracy, and reduced latencies to support real-time operations.

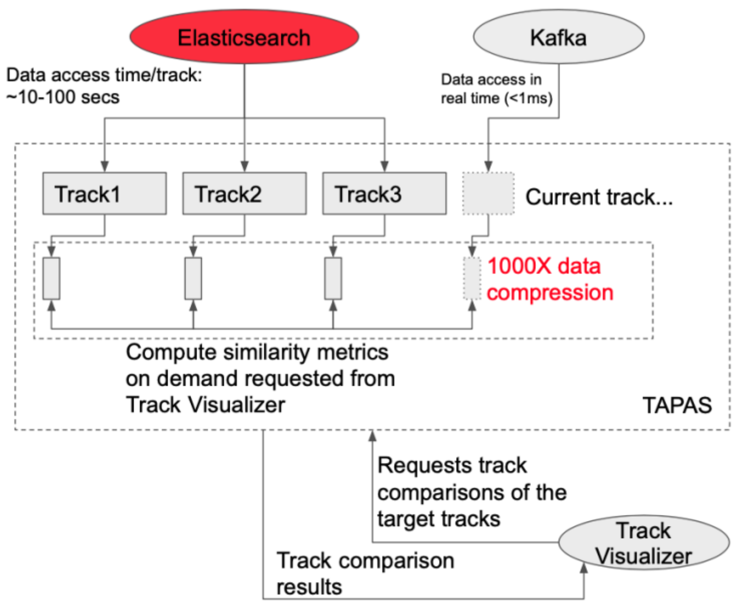

Fig. 1. Data flow in the DSN track comparison tool. Elasticsearch's data retrieval latency is more than 10 seconds, which requires retrieving data offline and compressing it by 1000X using piecewise polynomial approximation. Track Visualizer requests a track comparison to the Track Comparison Tool, and the system calculates a similarity metric on demand.

3. Operations

In this section we describe our operational scenarios. Specifically, we describe how TAPAS is expected to run within an operational setting and to generate the information expected by our clients in a time-optimal way. Specifically, we describe data processing scenarios where TAPAS is expected to provide comparison computations for the full set of historical tracks available within DCEP, as well as keep up with new real-time tracks as they are initiated by the DSN. Additionally, we describe the user experience of TAPAS through a discussion of a systematic user interface strategy. Through this prescribed approach to operations, we seek to provide an LCO with tools to better support their historical contextual estimation needs during live operations.

3.1 Data Processing

Given the real-time nature of DSN operations, our success is dependent on the ability of an LCO to generate historical context information, through our similarity score estimates, with as little latency as possible. One of the key complications in pursuing this goal is the scale of data volume we need to process to account for all real-time and historical tracks. Each DSN track can contain hundreds or even thousands of monitor data items. Each one of those monitor data items can involve thousands of individual time-stamped data points. Thus the total data volume, for example, 30 days of DSN tracks, can easily reach terabytes of data volume. Moreover, as new real-time tracks are initiated within the DSN, there exists a need to recompute similarity score estimates for these new tracks and their historical counterparts as the tracks progress over time. Recomputing this large volume of data in an ongoing fashion can be an expensive computational operation. For this reason, we compress track data (see Section 2.2) obtained from our data sources (Elasticsearch) as well as pre-generate our computations to the greatest extent possible for historical tracks (further described in Section 3.1.2). There are two processing modes we support to ensure our similarity score estimates are precomputed in a way that supports operational use: historical processing and forward keep-up processing. In both modes, we aim to pre-download the data necessary for our computations, generate the similarity scores in the background prior to user interaction with our tool, and return the results of these similarity score computations through the use of a cache. We describe these operational modes in the following sections.

3.1.1 Historical Processing

Historical processing involves the generation of track similarity scores for all combinations of available monitor data between all historical tracks available. As a result, our historical processing mode involves computing similarity score estimates for only non-real-time, fully completed DSN tracks. In this type of processing scenario, we expect to generate similarity scores a priori to support the low-latency track comparison scores that LCOs expect from TAPAS. For instance, consider a historical processing scenario such as pre-calculating the similarity score for all antennas' azimuth angles between all similarly configured tracks available in the past month. The keyword 'configured' here denotes the need to evaluate tracks based on initial preconditions, such as identical spacecraft, antenna, and other configuration factors such as downlink mode, equipment, etc. Table 1 provides a list of our current preconditions supported. Once these preconditions are held constant, the scope of available tracks that are presented to an LCO naturally reduces. For instance, in our example of computing similarity scores for azimuth angles, and holding preconditions such as the spacecraft and antenna type constant, we may identify only 5-10 relevant tracks within 30 days of track data as of contemporary DSN spacecraft tracking conditions. Further, the use of preconditions ensures we present only tracks that *should* be compared - according to an LCO defined criteria. In other words, although TAPAS in theory supports the generation of similarity score estimates for any combination of tracks, the addition of preconditions simplifies the user experience for the LCO to include only relevant comparisons they find fitting. We include a required set of preconditions as well as an optional set the LCO has the option to modify or remove entirely. Required preconditions such as monitor data item, spacecraft, and antenna are utilized pre-compute with respect to the similarity score computation, whereas all other preconditions are utilized post-compute (i.e. after the similarity scores have been generated) to down-filter the list of tracks the LCO receives when requesting a set of comparison scores (against other tracks) for a given reference track.

Name	**Required?**	**Description**
Monitor data item	Yes	A real-time, streaming data value providing monitoring information for DSN equipment
Spacecraft	Yes	The identifier for a given spacecraft being supported
Antenna	Yes	The unique identifier for a given antenna being leveraged for a track
Keywords	No	Miscellaneous, spacecraft-specific event information for a given track. For example, uplink or downlink labelled information or special advisories
Parameters	No	Values for miscellaneous keywords used to specifically configure tracks. For example, downlink modes, bands utilized,

		etc.
Equipment	No	The set of equipment involved within a particular track configuration

Table 1. A list of track comparison preconditions referenced prior to computing similarity scores. Preconditions help both ensure only similarly configured tracks are compared and reduce the comparison computational complexity. There are both required as well as optional preconditions. The latter is subject to grow.

Table 2 provides a summary of the expected processing scenarios, number of track combinations possible, and times taken to generate similarity scores for operational modes supported. These metrics assume 30 days of average DSN track data being available at a data volume reflective of spacecraft being tracked at the time this paper was published. As described in Table 2, our benchmarking indicates that precomputing 30 days of historical data for purely historical tracks takes over half a day of processing hours on a 4-core Intel Xeon Platinum 8260 CPU @ 2.40GHz. This is a significant amount of time; however, this is a one-time expense because once our computations have been made, they do not change for historical-to-historical track comparisons. Historical-to-historical track comparison scenarios represent the first step of our operational strategy. In other words, we expect to always initiate historical processing for the last 30 days of tracks during the initial run of our tool, as well as immediately after downtimes (discussed in Section 3.1.3), but after this initial run, we expect to execute the tool in forward keep-up processing mode only (discussed in Section 3.1.2). The exact ordering of processing modes utilized is thus as follows:

1. Historical processing mode: to precompute similarity scores between all historical, non-real-time tracks
2. Forward processing mode: ensuring live, real-time track comparison scores are generated as soon as the historical processing mode campaign completes
3. Catch-up processing mode: to eliminate any gaps in similarity score generation between the invocation of forward processing and historical processing

Track Comparison Scenario	**Processing Mode Required**	**No. of Track Combinations for Select Monitor Data**	**Pre-compute Similarity Score Computation Time**	**On-demand Similarity Score Computation Time**
Real-time track & Historical tracks	Forward processing	~10	~1 s	~0.2 s
Historical track & Historical tracks	Historical processing	~78,000	28 h	~0.2 s
Real-time track & Real-time / other tracks	Forward processing	~1	~0.2 s	~0.2 s

Table 2. A list of track comparison combination types and associated processing modes and count of possible combinations given an estimate of 30 days worth of tracks for current DSN operations. The count is subject to increase and vary in the coming years for the DSN.

3.1.2 Forward, Keep-Up Processing

This processing mode is meant to support the computation of similarity scores between monitor data within real-time DSN tracks and their historical or real-time counterparts. This is distinct from historical processing in that the similarity scores need to be regenerated throughout the duration of a track in order to keep up with real-time changes. Whenever a user of TAPAS needs to compare monitor data item performance between a real-time track and a historical track, or even another real-time track, they will expect the latest changes within the real-time track to be reflected in the computation of requested similarity scores quickly.

TAPAS receives real-time data from the DCEP's real-time data messaging engine, specifically Kafka[3]. Kafka enables the low-latency subscription of multiple monitor data items concurrently. TAPAS obtains this data through a few steps:

1. Obtaining the DSN schedule to identify all ongoing, real-time DSN tracks
2. Setting the required preconditions (i.e. spacecraft, antenna, and monitor data item)
3. For each real-time track, initiating a subscription to Kafka for monitor data matching the aforementioned required preconditions
4. Streaming monitor data to the TAPAS service as long as the track is ongoing
5. As new monitor data values from a track are received, computing similarity score estimates between the real-time track and all existing historical or real-time tracks matching the required preconditions at a predetermined update rate. Currently, TAPAS is targeting a 10 minute update frequency, which can be adjusted based on computational resources available.
6. Once the real-time track completes, close the Kafka subscription. Next, the previously real-time track is now treated as a historical track by TAPAS.

Using the above sequence of steps, forward processing ensures that all real-time tracks have corresponding similarity scores available on-demand by TAPAS clients. Moreover, forward processing also ensures that similarity scores are pre-computed and available after a given track commences and becomes a historical track. Thus, forward processing takes the place of historical processing in a keep-up fashion.

3.1.3 Catch-Up Processing

During rare occasions of TAPAS downtime, either due to unforeseen events or upgrade scenarios, forward processing is paused for a duration of time. It is thus necessary to ensure that a catch-up processing capability exists to ensure TAPAS has historical similarity score information available for missed tracks after downtime issues are resolved. Our approach to catch-up processing is to reuse historical processing capabilities within a set time range. In other words, we reuse TAPAS's historical processing capability and specify a start and end date time range that TAPAS is expected to compute historical processing similarity scores for. Prior to initiating this historical processing invocation, we must initiate forward processing to ensure we do not miss new, ongoing real-time tracks as well. Thus the sequence of events for catch-up scenarios is as follows:

1. Downtime event occurs bringing TAPAS offline.
2. Downtime event is resolved and TAPAS is brought online.
3. TAPAS historical-processing mode is invoked between the downtime start and end times only.
4. TAPAS historical-processing mode is invoked between downtime end and the current time.
5. TAPAS forward-processing mode is turned on again.

4. User Interaction

Thus far we have discussed the TAPAS service, or backend, in-depth. TAPAS provides a RESTful application programming interface (API) that can be invoked from a range of third-party tools. One of our main clients is DCEP. We make available the TAPAS similarity scores and related data products to the DCEP Track Visualizer (TV) user interface for supporting an intuitive experience for DSN LCOs. Our approach is to augment the TV feature-set to support the presentation of historical track context information through providing similarity scores and other information. To aid this endeavor, we've identified several usage scenarios and developed a user interface specification that is informed by quantitative and qualitative feedback from global DSN LCO user testing campaigns.

4.1 Usage Modes

There are several usage modes for the TV TAPAS user interface (UI) that are envisioned. The key starting point for our users is to have the users first select a single track to render in TV (called the "reference track"), and then to invoke TAPAS' similarity score estimation service for generating historical context information for tracks matching a set of preconditions with the reference track. These other tracks can be both historical as well as potentially real-time. Table 3 overviews the set of possible usage scenarios TAPAS currently supports, as well as expected functional responses. Note, that in addition to the computation of similarity scores, TAPAS also provides miscellaneous information useful to LCOs in the understanding and use of our similarity score. This includes histogram information related to how similarity scores are distributed among the historical tracks evaluated. This information in particular helps users assess the "normality" of an ongoing, real-time reference track in relation to a set of historical tracks. Additionally, we provide information such as a ranking of similarity scores by ascension or descension as well as the identification of outliers. The latter is an evolving set of augmentation overlays that is currently under active development. Finally, two analysis views are offered: summary mode and analysis mode. The former presents a summarization of comparison track data as it relates to a reference track given a similarity score - presented in a table format. The latter is an optional mode that offers a visual time series graph comparison between a reference track and comparison track(s) - providing more detailed information.

Usage mode	**Response generated**
1 reference track selected for a single monitor data item and summary mode requested	Table of all precondition matching tracks and their associated similarity scores, histogram information binning the frequency of identical similarity scores against tracks evaluated, and a default sorting of similarity scores by highest-to-lowest
1 reference track selected and 1 comparison track selected for a single monitor data item and analysis mode requested.	Similarity score values, time-series performance overlay of the two tracks, and optionally viewable spikes in the two tracks overlayed on top of their respective time series graphs
1 reference track selected and a variable set of comparison tracks selected for a single monitor data item and analysis mode requested.	Similarity score values, time-series performance overlay of the multiple tracks' monitor data, and optionally viewable spikes in the multiple tracks overlayed on top of their respective time series graphs

Table 3. Various usage scenarios envisioned with the DCEP TV integration with TAPAS UI. A single reference track can be compared with one, two, or many comparison tracks.

4.2 User Interface

TAPAS was designed to be a plugin within the Track Visualizer (TV), and, as a result, inherited its visual design language. Still, TAPAS underwent various cycles of design and iteration to ensure its workflow and design conventions were consistent with user expectations (See Section 4.3). TAPAS requires the LCO to first select a track in the TV tool (referred to as the Reference Track). TAPAS can then be launched by comparing all tracks available through DCEP against the Reference Track. Figure 2 shows the Monitor Data Item Selection window - the first of a two-step process by which users can visualize a comparison. Here, a user must select a monitor data item to be the basis of a TAPAS comparison.

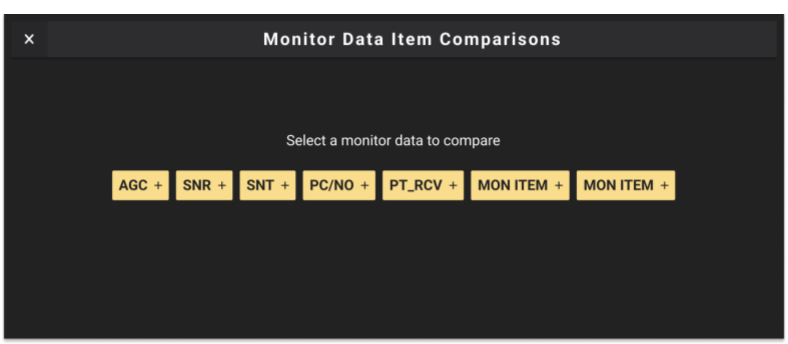

Fig. 2. The Monitor Data Item Selection window. Users are required to select one monitor data item among many.

Once a monitor data item is selected, the user progresses to the second step of the process, Comparison Track Selection. In this subsequent step, users are presented with preconditions (e.g., antenna, spacecraft, etc.) and a list of candidate comparison tracks. This list can be sorted by similarity scores (see Section 4.3). Figure 3 shows the Comparison Track Selection window. Once a user determines an appropriate track, the blue "Plot" button generates a visualization of both Reference and Comparison Tracks in a new view.

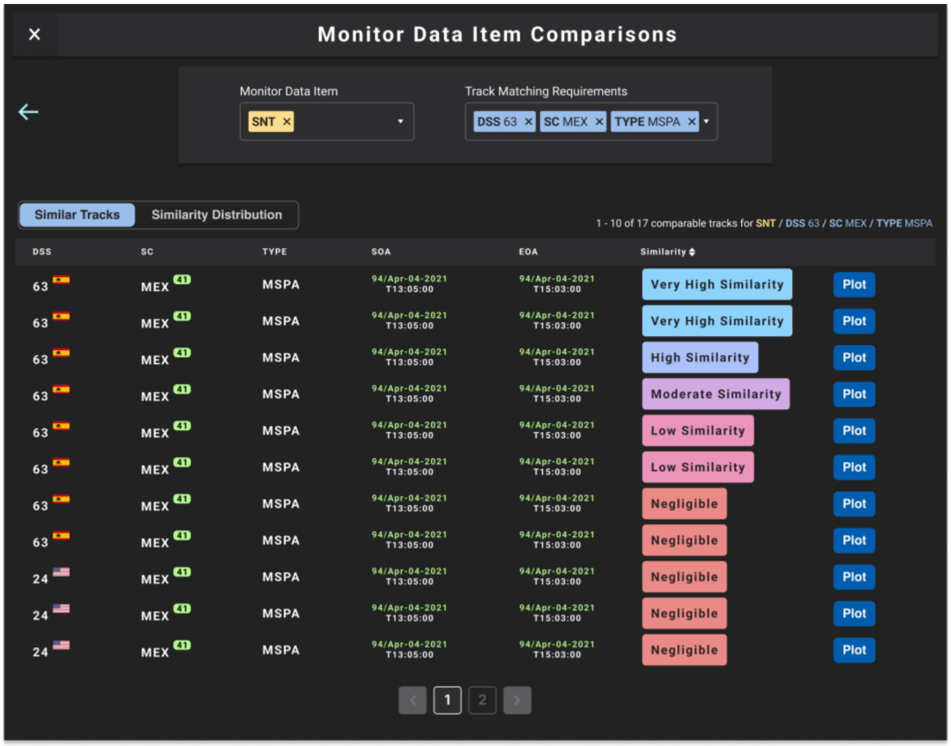

Fig. 3. The Comparison Track Selection window. Here users can determine an appropriate comparison track based on a variety of factors.

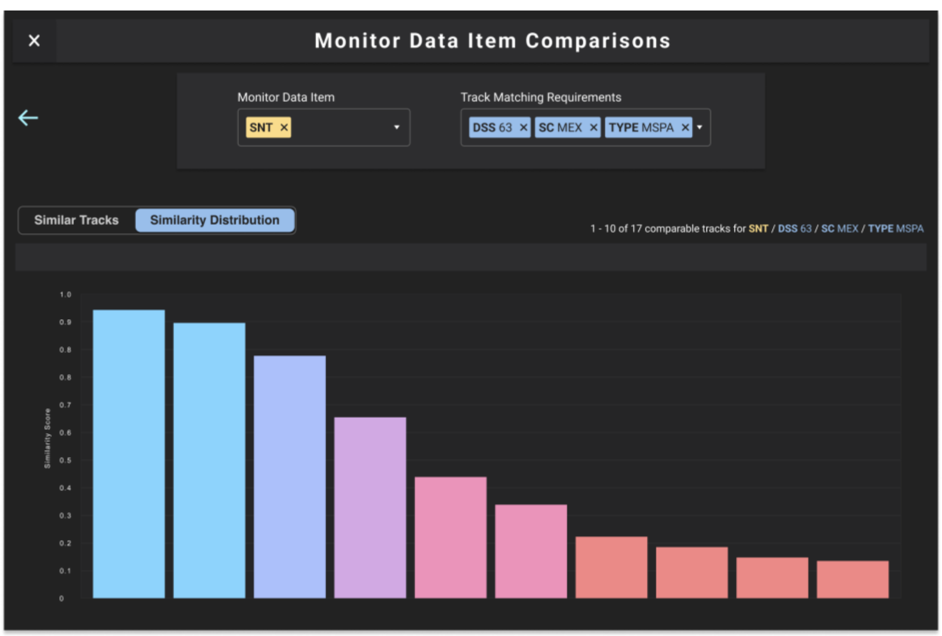

Fig. 4. The Comparison Track Selection window with Similarity Score distribution.

The TAPAS user interface uses a visual strategy to aid similarity score interpretations, including distributions (Figure 4). Rather than show the score as a numerical value only, the score is categorized. We employ the interpretations of correlation coefficients described in Hinkle et. al. [4] and prescribe these ranges to our Similarity Scale (a range of -1 to 1). Figure 5 shows the Similarity Scale and its nine interpretations across the range of -1 and 1. To supplement interpretation, we also applied a color to each label to visually emphasize a track similarity score and its interpretation.

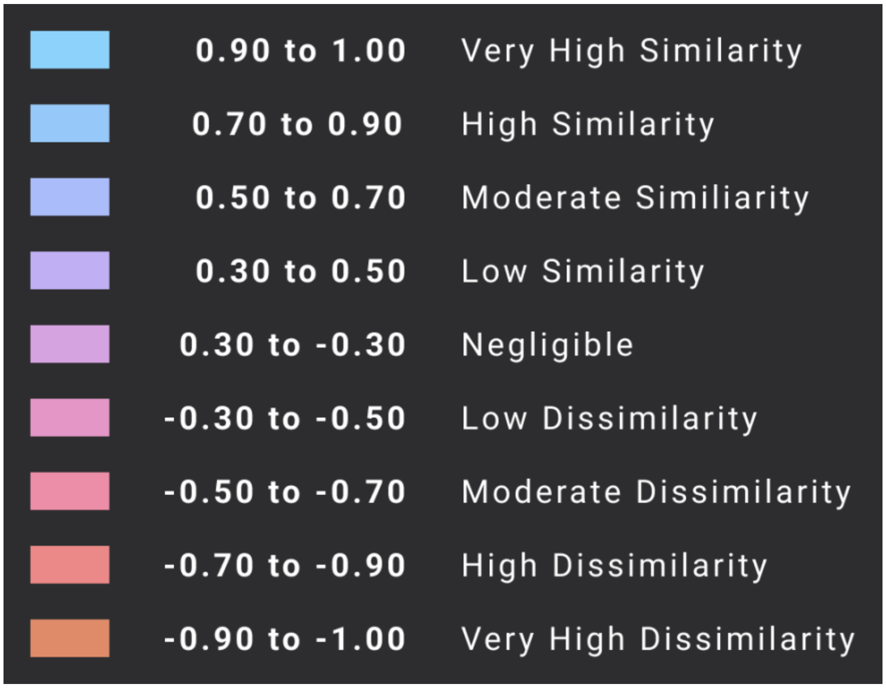

Fig. 5. The Similarity Scale. Nine interpretations correspond to scores between 1 and -1 (Hinkle *et al.*, 2003)

4.3 User Experience

In this section, we describe our user experience approach to validate TAPAS with its user community. Our methods range from surveying, interviewing, to usability testing. Each method was selected to address specific research questions for which it was best suited. The user engagements were conducted in-person across the DSN at global complexes. Where necessary, methods were adjusted for online arrangements. All users were presented with an informed consent form prior to their participation.

An electronic survey was disseminated across the DSN to solicit feedback on needs and desired capabilities. Users were presented with four candidate analysis tools for evaluation and for which provided ratings on their overall performance and impact. A total of 21 LCOs (users) provided their feedback, generally highlighting a need for 1) the ability to identify off-nominal behavior that may be present in an ongoing track, and 2) which aspects of track behavior deviated most and why.

Semi-structured interviews were scheduled with users to investigate the emergent topics from the survey, and continued to be carried out throughout various stages of later design. A total of seven users were available for interviewing and provided insight to determine factors that are helpful for track comparisons, informative groupings of monitor data, and features of track similarities. We were also able to explore notional interface designs for TAPAS.

Several rounds of usability testing were carried out to refine the TAPAS workflow with a total of nine users. Users were presented with clickable prototypes and were instructed to carry out relevant tasks. Information gathered during usability testing sessions allowed us to further iterate on the design of our approach. Most notably, users were asked to complete exercises to test the viability of our similarity scores when compared against LCO-derived similarity judgements. Users were presented with pairs of time series plots visualizing monitor data item performance throughout the entirety of the tracks (each track combination had been pre-screened and had TAPAS similarity scores generated for it). Users were asked to determine track similarities using the Similarity Scale discussed in the previous section (see Figure 5). An analysis of user responses and TAPAS similarity scores was performed post hoc. Qualitative information regarding their judgements were captured and used for subsequent iteration of user interface designs.

5. Discussion

Generation of relevant historical context and similarity scores is a task intuitively performed by LCOs today within the DSN. Through interviews with LCOs, we have confirmed that human memory of recent track performance that is relevant to a current, ongoing track is often leveraged and depended upon. Techniques such as paper archives or more recently electronic means of logging previous support information have been utilized, but LCOs still primarily rely on first-hand knowledge to compare tracks. Our novel similarity score estimations are an attempt at automating as well as capturing the key characteristics of recent historical tracks that are relevant in comparing them to ongoing real-time tracks. This is a complex problem and our approach has been tested with LCOs through user studies described in this paper. Further verification and assessment of our methods is needed and will be captured with the release of our tool for field testing within global DSN operations. Improvements and iterations are expected.

In our user testing, participants generally agreed with the guidance of generated TAPAS similarity scores. However, there were also discrepancies between user-derived judgements and TAPAS generated scores. A few factors explain this. First, our user testing was constrained on the number of tracks that were able to be shown to each participant. Each session lasted approximately 30 - 45 minutes, enough for two or three sets per participant (each set contained two tracks). As a result, some participants felt their calibration of the scale could not be established with so few examples. Secondly, some participants were reluctant to provide a similarity judgement due to lack of information or personal preference. Lastly, some participants were much more conservative in providing responses due to their own internal criteria, making it difficult to establish a baseline for this exercise. Qualitative feedback from our user testing shows that users were able to derive additional context for the tracks that they were observing (e.g., antennas and spacecraft inferred a particular frequency band for the monitor data item ranges). While LCOs were able to build on these contextual clues, it may serve our users best if we are explicit about these supplemental track characteristics. Providing more information may lend itself to additional context building. We also learned that users had a difficult time distinguishing the difference between similarity categories across the 9-point scale (i.e., difference between a Very High Similarity rating and High Similarity rating). This suggests that less categories may allow for more parity between LCO and TAPAS judgements. Lastly, the utility of negatively correlated tracks is still subject to debate. For scenarios where users were presented with negatively correlated tracks, none were able to accurately interpret the negative correlation and judge it accordingly.

6. Conclusions

The similarity score is a numerical approach to capturing the performance relationship between historical tracks and ongoing, real-time reference tracks. We've demonstrated the feasibility of generating this score in a scalable fashion for up to 30 days of DSN tracks, and a path to support further growth. This approach has been the subject of user testing to validate the approach and assess an appropriate presentation method via a graphical user interface. Further field testing will provide valuable real-world results on the utility of our method. Further, future work centering on generating similarity scores for groups of monitor data collectively, or entire tracks in aggregate, set the potential for further value for the LCO.

Acknowledgements

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). We thank John Mason for his input on the DSN operational use case. We also thank Umaa Rebbapragada for her contributions to spike detection algorithms.

References

[1] Choi, Josh, Rishi Verma, and Shan Malhotra. "Achieving fast operational intelligence in NASA's Deep Space Network through complex event processing." 14th International Conference on Space Operations. 2016.

[2] Yun, Kyongsik, Rishi Verma, and Umaa Rebbapragada. "Time Series Comparisons in Deep Space Network." American Institute of Aeronautics and Astronautics, Accelerating Space Commerce, Exploration, and New Discovery (2021).

[3] Verma, Rishi. "Mission-Critical, Real-Time Fault-Detection for NASA's Deep Space Network Using Apache Kafka." Kafka Summit San Francisco 2019. Kafka Summit San Francisco 2019, San Francisco.

[4] Hinkle DE, Wiersma W, Jurs SG. Applied Statistics for the Behavioral Sciences. 5th ed. Boston: Houghton Mifflin; 200